Avancées en Vision Neuromorphique : Représentation Événementielle, Réseaux de Neurones Impulsionnels Supervisés et Pré-entraînement Auto-supervisé

Sami Barchid

Université de Lille

Thèse

pour obtenir le grade de :

Docteur de l'Université de Lille

dans la spécialité

"Informatique"

par

Sami Barchid

Avancées en Vision Neuromorphique : Représentation Événementielle, Réseaux de Neurones Impulsionnels Supervisés et Pré-entraînement Auto-supervisé

Thèse soutenue le 05/12/2023 devant le jury composé de :

M.	Jean MARTINET	Professeur, i3S, Université Côte d'Azur	(Rapporteur)
M.	Laurent PERRINET	Directeur de Recherche, INT (CNRS), Université Aix-Marseille	(Rapporteur)
Mme	Clarisse DHAENENS	Professeur, CRIStAL, Université de Lille	(Examinatrice)
M.	Timothée MASQUELIER	Directeur de Recherche, CerCo (CNRS), Université de Toulouse 3	(Examinateur)
M.	José MENNESSON	Maître de Conférences, CERI Numérique, IMT Nord-Europe	(Co-encadrant)
M.	Chaabane DJÉRABA	Professeur, CRIStAL, Université de Lille	(Directeur de Thèse)

Centre de Recherche en Informatique, Signal et Automatique (CRIStAL)
Bâtiment ESPRIT Avenue Henri Poincaré 59655 Villeneuve d'Ascq
France

Résumé

Avec l'avènement de l'apprentissage profond, les réseaux de neurones artificiels (ANNs) sont devenus l'approche prédominante pour résoudre les tâches de vision par ordinateur, atteignant des performances remarquables lorsqu'ils sont correctement entraînés. Cependant, au fil du temps, les ANNs ont gagné en complexité et en taille, exigeant de plus en plus de ressources informatiques et entraînant une consommation d'énergie significative.

Pour résoudre le problème de la consommation d'énergie, des technologies neuromorphiques telles que les réseaux de neurones impulsionnels (SNNs) et les caméras événementielles ont émergé comme des solutions prometteuses. Les SNNs sont des réseaux de neurones inspirés de la biologie qui traitent l'information sous forme d'impulsions binaires asynchrones. Les caméras événementielles sont des capteurs visuels économes en énergie composés de pixels indépendants réagissant de manière asynchrone aux changements de luminosité, produisant une sortie binaire et asynchrone appelée "événements". Malgré leurs avantages, notamment en termes d'efficacité énergétique, ces approches neuromorphiques restent moins développées que les solutions de vision conventionnelles comprenant des images et des ANNs.

La principale motivation de cette thèse est d'approfondir notre compréhension de ces technologies neuromorphiques. Pour ce faire, nous explorons d'abord de nouveaux problèmes de vision en utilisant ces technologies, puis nous utilisons ces nouvelles tâches comme contextes expérimentaux pour analyser les aspects fondamentaux de la vision neuromorphique. Dans nos contributions, nous explorons trois principales orientations de recherche.

Tout d'abord, nous développons une nouvelle technique de représentation d'événements en images événementielles, en mettant l'accent sur l'intégration efficace de l'information temporelle. Nos expériences démontrent les avantages compétitifs de notre nouvelle approche, tant en termes de performances que de robustesse face aux corruptions des caméras événementielles.

Deuxièmement, nous examinons l'utilisation de SNNs profonds supervisés pour des solutions de vision artificielle économes en énergie. Nous abordons de nouveaux

défis de vision par ordinateur pour les SNNs, notamment la localisation d'objet (à partir d'images ou d'événements) et la reconnaissance d'expressions faciales (FER) basée sur les événements. De plus, nous exploitons la conception de SNNs profonds pour des tâches de vision par ordinateur afin d'analyser l'impact de plusieurs aspects fondamentaux des SNNs sur leurs performances. Cela inclut une étude sur les codages neuronaux pour convertir les images en trains d'impulsions, la robustesse des SNNs face aux corruptions des capteurs, l'influence de la latence temporelle, les avantages de l'augmentation de données pour l'entraînement des SNNs, et l'efficacité énergétique des SNNs par rapport à des ANNs de même complexité. Nos études fournissent des informations cruciales, révélant les comportements spécifiques des SNNs lorsqu'ils sont entraînés de manière supervisée, par rapport à d'autres règles d'apprentissage traditionnelles pour les neurones impulsionnels.

Enfin, nous posons les bases d'une nouvelle approche visant à réduire le besoin de données événementielles annotées utilisées pour former des réseaux de neurones (ANN ou SNN). Nous concevons une méthode simple mais très efficace d'apprentissage de représentations auto-supervisé (SSRL) pour pré-entraîner un encodeur convolutif sans supervision. Comme cette méthode est nouvelle, nous définissons des protocoles d'évaluation standardisés pour comparer les performances de notre approche de SSRL événementiel avec les futurs travaux de recherche. À travers nos études expérimentales, nous démontrons l'impact significatif du SSRL événementiel pour réduire la nécessité de données annotées et analysons les distinctions entre différents types de réseaux de neurones dans l'extraction de caractéristiques non supervisées.

Mots Clés : Intelligence Artificielle, Vision par Ordinateur, Caméra Événementielle, Réseau de Neurones Impulsionnels, Calcul Neuromorphique

Abstract

With the emergence of deep learning, Artificial Neural Networks (ANNs) have become the predominant approach for solving computer vision tasks, achieving remarkable performance when properly trained. However, over time, ANNs have grown in complexity and size, demanding increasingly more computational resources and resulting in significant energy consumption.

To address this energy consumption issue, neuromorphic technologies inspired by the biological brain, particularly Spiking Neural Networks (SNNs) and event cameras, have emerged as promising solutions. SNNs, on one hand, are neural networks inspired by biology that efficiently process information as asynchronous sequences of binary spikes. Event cameras, on the other hand, are energy-efficient visual sensors comprising independent pixels that react asynchronously to changes in brightness, producing a sparse, asynchronous, and binary output known as "events." Due to their numerous advantages, notably in terms of energy efficiency, these neuromorphic approaches have attracted considerable attention in recent years. However, despite their increasing adoption for addressing computer vision challenges, this emerging field of "neuromorphic vision" still lags behind more conventional vision solutions such as images and ANNs.

The primary motivation of this thesis is to advance our understanding of these neuromorphic technologies. To do so, we first explore new vision problems using these technologies and subsequently utilize these novel tasks as experimental contexts to analyze fundamental aspects of neuromorphic vision. In our research contributions, we investigate three key research directions.

First, we develop a novel technique for representing events in event frames, with a focus on the efficient integration of temporal information. Our experiments demonstrate the competitive advantages of our new approach, both in terms of performance and robustness against event camera corruptions.

Secondly, we delve into the use of deep SNNs trained in a supervised manner for energy-efficient computer vision solutions. We address new computer vision challenges for SNNs, including single object localization (using frames or events) and event-based

Facial Expression Recognition (FER). Additionally, we leverage the effective design of deep SNNs for complex computer vision tasks to analyze the impact of several fundamental design aspects of supervised SNNs on their performance. This includes an investigation into neural coding schemes for converting images into spike trains, the robustness of SNNs against sensor corruptions, the influence of temporal latency, the benefits of event-based data augmentation for SNN training, and the energy efficiency of the designed SNNs compared to ANNs with similar architecture. Our studies yield critical insights, revealing specific behaviors of SNNs when trained in a supervised manner, compared to other traditional learning rules for spiking neurons.

Finally, we establish the groundwork for a novel framework aimed at reducing the need for annotated event-based data used to train neural networks (ANNs or SNNs). We design a straightforward yet highly effective method for Self-Supervised Representation Learning (SSRL) to pretrain a convolutional encoder without supervision. As this method represents a novelty, we define standardized evaluation protocols to benchmark the performance of our event-based SSRL approach in comparison to future research works. Through our experimental investigations, we demonstrate the substantial impact of event-based SSRL in reducing the requirement for labeled data and analyze the distinctions between various types of neural networks in extracting unsupervised features.

Keywords: Artificial Intelligence, Computer Vision, Event Camera, Spiking Neural Network, Neuromorphic Computing

Remerciements

Je tiens à exprimer mes plus sincères remerciements aux personnes suivantes qui ont joué un rôle important dans la réalisation de cette thèse.

Tout d'abord, je remercie chaleureusement ma partenaire, Vérane Martin, pour m'avoir apporté un soutien indéfectible pendant ces trois années d'épreuves. Sa présence, son écoute et ses encouragements ont été essentiels pour me permettre de poursuivre mes recherches malgré les difficultés.

Je suis également très reconnaissant envers ma famille pour leur soutien inconditionnel durant cette longue période. Leurs encouragements et leur confiance en moi ont été des sources d'inspiration constantes.

Je tiens à remercier mon directeur de thèse, Chaabane Djéraba, pour son travail d'encadrement.

Mes plus grands remerciements vont à mon co-encadrant, José Mennesson, pour son travail d'encadrement respectueux, la relation de confiance et de transparence qu'il a su installer, pour ses conseils et explications avisés sur chaque étape de la thèse, pour sa réactivité et pour sa patience. Sa présence a été un soutien précieux tout au long de cette aventure.

Je souhaite également remercier les collègues doctorants et amis que j'ai pu rencontrer, Mireille, Nicolas, Mathieu, Gaspard, Victor, Vivien et Théo, pour leur aide, leur bonne humeur et leurs conseils.

Je tiens également à exprimer ma gratitude envers les communautés open-source en générale, qui permettent d'ouvrir les savoirs à tous, sans lesquelles je n'aurais sans doute pas pu me former aussi librement. Dans le même registre, je remercie les nombreux utilisateurs de StackOverflow pour avoir répondu à tant de mes questions en termes de programmation.

Je remercie le personnel de l'Institut Paul Lambin, haute-école bruxelloise, pour avoir éveillé en moi une passion pour les sciences informatiques, grâce à son cursus de Bachelor en Informatique de Gestion de grande qualité.

Enfin, je ne saurais oublier Olly, mon jeune chiot, pour ne (presque) pas avoir détruit le mobilier de la maison pendant la rédaction de ce manuscrit. Je lui promets de rajouter secrètement quelques croquettes à chaque repas pour saluer son calme et son contrôle de soi.

Encore une fois, je remercie sincèrement toutes les personnes qui ont contribué de près ou de loin à cette thèse.

Table des Matières

Glossary

2D-CNN – 2D Convolutional Neural Network (Réseau de neurones convolutif 2D).

3D-CNN – 3D Convolutional Neural Network (Réseau de neurones convolutif 3D).

AER – Adress-Event Representation (Représentation adresse-événement).

ANN – Artificial Neural Network (Réseau de neurones artificiels).

BPTT – Backpropagation Through Time (Rétropropagation à travers le temps).

CNN – Convolutional Neural Network (Réseau de neurones convolutif).

CPU – Central Processing Unit (Unité centrale de calcul, communément appelée "Processeur").

CSNN – Convolutional Spiking Neural Network (Réseau de neurones impulsionnels convolutif).

EDA – Event Data Augmentation (Augmentation de données événementielle).

FER – Facial Expression Recognition (Reconnaissance d'expressions faciales).

FLOPS – FLoating-point Operation Per Second (Opérations en virgule flottante par seconde).

GPU – Graphics Processing Unit (Processeur graphique).

HPC – High Performance Computing (Calcul haute performance).

IF – Integrate-and-Fire.

IMU – Inertial Measurement Unit (Unité de mesure inertielle).

IoU – Intersection over Union (Intersection sur l'union).

LIF – Leaky Integrate-and-Fire.

MAC – Multiply-ACcumulate operation (Opération de multiplication et accumulation).

MLP – Multi-Layer Perceptron (Perceptron multicouche).

NLP – Natural Language Processing (Traitement du langage naturel).

RNN – Recurrent Neural Network (Réseau de neurones récurrents).

SNN – Spiking Neural Network (Réseau de neurones impulsionnels).

SSL – Self-Supervised Learning (Apprentissage auto-supervisé).

SSRL – Self-Supervised Representation Learning (Apprentissage de représentations auto-supervisé).

STDP – Spike-Timing Dependent Plasticity (Plasticité en fonction du timing des impulsions).

ViT – Vision Transformer (Transformeur de Vision).

1

Introduction

Sommaire

1.1 Motivation

La vision artificielle (ou vision par ordinateur) est un domaine important de nos jours en raison de ses applications variées, telles que la conduite des véhicules autonomes, l'inspection des lignes de production en usine, la surveillance de situations inhabituelles et le diagnostic des maladies à partir de l'imagerie médicale. Toutes ces tâches sont gérées avec aisance par les humains, grâce aux capacités formidables de notre cerveau, un des systèmes de traitement de l'information les plus avancés. Composé de 80 milliards de neurones reliés par 150 trillions de synapses, le cerveau humain est un système massivement parallèle qui nous octroie nos facultés pour raisonner, apprendre et réagir aux stimuli de notre environnement. En plus d'être l'un des systèmes de traitement de l'information les plus perfectionnés, le cerveau est également très efficace en termes de consommation énergétique, ne nécessitant que 20W pour fonctionner. De par son incroyable complexité, le fonctionnement du cerveau n'est toujours pas percé à jour, et les échanges d'informations ayant lieu entre les neurones ne sont pas encore pleinement compris, au point que la compréhension de l'origine du raisonnement et de l'intelligence reste toujours un mystère.

Le domaine de l'intelligence artificielle a souvent cherché à s'inspirer du cerveau pour développer de nouvelles approches. Dans cette tendance, l'avancée actuelle la plus remarquable concerne les réseaux de neurones artificiels ("*Artificial Neural Networks*" ou ANNs, en anglais). Grâce à l'émergence de l'apprentissage profond, les architectures ANNs profondes ont démontré des capacités impressionnantes pour résoudre des problèmes complexes en vision artificielle, en traitement du langage, et bien d'autres domaines.

Malgré les performances impressionnantes et toujours en amélioration de ces ANNs, leur conception exige la mise en place de plusieurs millions, voire de milliards de neurones artificiels, tous simulés sur une architecture matérielle de type Von Neumann (CPU ou GPU). Cette simulation entraîne une consommation énergétique considérable. Avec l'augmentation constante de la taille de ces ANNs, cette consommation d'énergie devient un problème majeur, tant du point de vue environnemental que pour la création de systèmes de vision embarqués.

Une solution potentielle pour résoudre le problème de la consommation énergétique des ANNs pourrait impliquer l'utilisation de réseaux de neurones impulsionnels ("*Spiking Neural Networks*" ou SNNs, en anglais). Les SNNs représentent une forme de réseau de neurones plus réaliste du point de vue biologique. Dans les SNNs, les

neurones communiquent en utilisant des signaux électriques binaires asynchrones appelés "impulsions". Ce mode de fonctionnement particulier offre la perspective d'une réduction de la consommation d'énergie, surtout lorsqu'il est mis en œuvre sur des architectures spécialisées telles que le matériel neuromorphique. Ces architectures neuromorphiques présentent des similitudes avec la structure de notre cerveau, comprenant des unités de mémoire et de calcul parallélisées et distribuées. De plus, elles nécessitent moins d'énergie car elles traitent des valeurs binaires éparses, éliminant ainsi le besoin d'opérations de multiplication gourmandes en énergie.

Cependant, un défi majeur entravant l'application des SNNs pour résoudre les tâches de vision par ordinateur réside dans la représentation non conventionnelle de l'information à l'aide d'impulsions, qui diffère de la représentation conventionnelle en images, c'est-à-dire des grilles denses de valeurs réelles. En regardant du côté de la biologie, notre système visuel utilise déjà une représentation basée sur les impulsions émises par notre rétine. En s'inspirant de ce dispositif biologique, un nouveau type de capteur visuel émergent a vu le jour : les caméras événementielles.

Ces caméras événementielles capturent des données visuelles en utilisant des pixels photorécepteurs qui détectent de manière autonome les changements de luminosité, imitant ainsi le comportement de nos yeux. La sortie de ces caméras se présente sous forme d'événements, où chaque changement de luminosité est caractérisé par sa position, son horodatage précis (à la microseconde près) et une valeur binaire indiquant si le changement de luminosité est positif ou négatif. En termes pratiques, ces changements de luminosité correspondent aux mouvements d'objets, ce qui signifie que la caméra ne génère des événements que lorsqu'il y a un mouvement à des emplacements et à des moments spécifiques. En comparaison avec une caméra classique, une caméra événementielle présente de multiples avantages, notamment une résolution temporelle plus élevée et une faible consommation énergétique grâce à la parcimonie des événements par rapport aux images.

Les deux technologies bio-inspirées que nous avons mentionnées, à savoir les SNNs et les caméras événementielles, peuvent être désignées sous le terme d'"approches" ou de "technologies neuromorphiques". Leur mode de fonctionnement spécifique basé sur des impulsions asynchrones offre d'excellentes opportunités pour réduire la consommation énergétique des modèles profonds en vision artificielle. Cependant, ces technologies émergentes sont moins étudiées et moins matures que les méthodes conventionnelles basées sur les ANNs et les images. Par conséquent, ces approches

neuromorphiques nécessitent des développements et des analyses approfondies dans le domaine de la vision artificielle, dans le but de mieux comprendre les spécificités de ces technologies et de concevoir des systèmes de vision novateurs à haute efficacité énergétique. La motivation des travaux présentés au cours de cette thèse est fondée sur cette constatation, car nous aspirons à développer et à analyser l'utilisation de ces technologies neuromorphiques dans le contexte de la vision artificielle, que l'on peut ainsi désigner comme "vision neuromorphique".

1.2 Objectifs

Par le biais de cette thèse, notre objectif est de faire progresser l'application des technologies neuromorphiques dans les tâches de vision artificielle. Dans cette perspective, nous visons à développer différents modèles d'apprentissage profond qui intègrent au moins l'une des technologies neuromorphiques étudiées, à savoir les SNNs ou les caméras événementielles. En plus de leur conception, nous utilisons les méthodes proposées comme des cadres d'étude expérimentale visant à approfondir notre compréhension du fonctionnement de ces technologies neuromorphiques sous divers aspects fondamentaux. Plus précisément, nous nous concentrons sur trois contextes particuliers, qui sont décrits comme suit :

La Représentation d'Événements en Images Événementielles. Afin de traiter des flux d'événements asynchrones avec des algorithmes de vision conventionnelle, tels que les ANNs, il est impératif d'appliquer une étape de pré-traitement spécifique qui transforme ces événements en images événementielles. Dans le cadre de cette thèse, nous voulons concevoir et examiner une nouvelle méthode pour cette représentation sous forme d'images événementielles, dans le but d'améliorer le traitement des flux d'événements par des algorithmes de vision conventionnelle.

La Conception et l'Étude de SNNs Profonds. Récemment, des avancées significatives ont permis d'entraîner des architectures SNNs profondes de manière supervisée, obtenant ainsi des performances compétitives par rapport aux ANNs. Toutefois, l'application des SNNs dans de nombreuses tâches de vision reste largement inexplorée. Un des objectifs de cette thèse est donc double : premièrement, concevoir des modèles profonds basés sur des SNNs pour explorer leur utilisation dans un nouveau contexte de vision, et deuxièmement, analyser les effets de divers aspects fondamentaux de conception des SNNs sur les performances du modèle.

La Réduction du Besoin en Données Événementielles Annotées. Dans le domaine de l'apprentissage profond, la recherche de méthodes visant à réduire la dépendance aux données annotées pour entraîner des modèles suscite un intérêt considérable, notamment en vision artificielle. Cependant, en ce qui concerne la vision événementielle, de telles approches sont encore peu développées, ce qui entrave la création de nouveaux modèles performants. Cette thèse a pour objectif de développer une méthode nouvelle, à la fois simple et efficace, permettant de réduire le besoin en

données annotées lors de la phase d'apprentissage d'un réseau de neurones profond (qu'il s'agisse d'un ANN ou d'un SNN) pour le traitement des flux d'événements.

1.3 Organisation du Manuscrit et Contributions

Conformément aux objectifs établis dans le cadre de cette thèse, nous avons réalisé des contributions scientifiques tout au long de ce manuscrit. Cette section a pour but de présenter le contenu de ces contributions et leur disposition au sein de ce manuscrit. En relation avec les trois objectifs exposés dans la Section 1.2, notre travail est structuré en trois chapitres distincts, précédés par un chapitre de revue des travaux pertinents de l'état de l'art, et suivis par un chapitre présentant la conclusion générale de notre manuscrit.

Deuxième Chapitre. Avant de présenter nos contributions, le Chapitre 2 offre une vue d'ensemble des domaines de recherche liés à notre thèse, à savoir les ANNs, SNNs et la vision événementielle. Ce chapitre se penche plus particulièrement sur la définition des concepts clés nécessaires à la compréhension de notre travail. De plus, il explore les avancées notables dans chaque domaine étudié, tout en mettant en évidence les défis scientifiques qui subsistent, lesquels ont motivé notre thèse.

Troisième Chapitre. Dans le Chapitre 3, nous abordons l'objectif lié à la représentation des événements en vue de leur traitement par des algorithmes de vision conventionnelle, c'est-à-dire basés sur des images statiques. Nous développons une nouvelle technique de représentation appelée "Bina-Rep" qui permet d'intégrer de manière efficace l'information temporelle d'un flux d'événements dans une image événementielle. Pour ce faire, nous concevons une architecture de référence basée sur un ANN et nous comparons différentes représentations d'événements de l'état de l'art à Bina-Rep sur des tâches de classification couramment utilisées en vision événementielle. En outre, nous profitons de ce cadre expérimental pour évaluer la robustesse de chaque représentation d'événements étudiée face à des corruptions courantes des capteurs événementiels. Ce manuscrit décrit la "corruption d'un capteur visuel" comme toute altération du signal capté par ce capteur. La robustesse d'une méthode de vision face à une corruption donnée est définie comme sa capacité à maintenir ses performances malgré cette corruption.

Quatrième Chapitre. Le Chapitre 4 est dédié au développement et à l'analyse des SNNs profonds pour des applications en vision par ordinateur. Nous explorons diverses tâches de vision, notamment la localisation d'objet dans des images statiques, la localisation d'objet dans des flux d'événements, ainsi que la reconnaissance des

expressions faciales ("*Facial Expression Recognition*" ou FER, en anglais) à l'aide d'une caméra événementielle. Dans un premier temps, nous établissons une preuve de concept pour évaluer l'applicabilité des architectures SNNs profondes supervisées dans le contexte de tâches de vision complexes. Nous développons un modèle SNN destiné à la localisation d'objet dans des images statiques encodées sous forme d'impulsions. Forts des enseignements tirés de cette preuve de concept, nous concevons ensuite un modèle SNN convolutif simple et polyvalent pour étudier sa capacité à extraire des informations pertinentes pour diverses tâches de vision. Ce modèle générique est évalué dans le contexte de la localisation d'objet, que ce soit avec des images statiques ou des flux d'événements, et comparé à un modèle ANN d'architecture similaire. Nos études expérimentales explorent différents aspects fondamentaux de la conception des SNNs, notamment les schémas de codages neuronaux pour le traitement d'images statiques, l'importance de la latence temporelle pour les performances, la robustesse des modèles face à des corruptions courantes des caméras, ainsi que l'estimation de l'efficacité énergétique lors du déploiement. Enfin, nous élargissons le champ d'application des SNNs en abordant la FER événementielle à l'aide du modèle générique basé sur un SNN. Nous créons des bases de données événementielles synthétiques pour évaluer notre approche et réalisons une étude exploratoire de cette nouvelle tâche de vision événementielle. Cette étude inclut l'analyse de l'impact des techniques d'augmentation de données sur l'apprentissage de nos modèles et l'évaluation de leur efficacité énergétique.

Cinquième Chapitre. Dans le Chapitre 5, notre attention se tourne vers l'objectif de réduire la dépendance aux données annotées pour les modèles d'apprentissage en vision événementielle. Nous introduisons une nouvelle pratique dans ce domaine, baptisée l'apprentissage de représentations auto-supervisées ("*Self-Supervised Representation Learning*" ou SSRL, en anglais). L'objectif de cette approche est de pré-entraîner un réseau de neurones (qu'il s'agisse d'un ANN ou d'un SNN) sur des flux d'événements sans l'aide d'annotations. Pour ce faire, nous développons une approche de référence pour le SSRL événementiel, basée sur un encodeur convolutif et une distribution d'augmentations de données. De plus, nous définissons plusieurs protocoles d'évaluation essentiels pour permettre des comparaisons avec les travaux futurs de l'état de l'art. En utilisant ce nouveau cadre d'étude, nous évaluons l'efficacité de plusieurs techniques d'augmentation de données événementielles de l'état de l'art pour déterminer celles qui apportent une contribution significative au SSRL événementiel. Nous analysons également les différences entre les types de réseaux de

neurones étudiés en ce qui concerne leur capacité à extraire des caractéristiques de manière non supervisée.

Sixième Chapitre. Le Chapitre 6 présente les conclusions finales de nos travaux, en récapitulant les résultats de nos contributions et en introduisant les différentes pistes de recherche à suivre pour les travaux postérieurs à cette thèse.

2

État de l'Art

Sommaire

Dans ce chapitre, nous effectuons une revue de la littérature afin de situer nos travaux par rapport aux avancées antérieures dans le domaine de la vision artificielle et événementielle, en mettant l'accent sur l'utilisation des réseaux de neurones artificiels traditionnels ainsi que des réseaux de neurones impulsionnels. Notre revue de l'état de l'art se divise en trois principales sections.

La première section est consacrée aux réseaux de neurones artificiels traditionnels. Nous présentons une vue d'ensemble de ces réseaux, en expliquant brièvement leur fonctionnement et certaines de leurs grandes avancées passées. Nous mettons notamment en évidence les limitations de ces méthodes motivant nos travaux.

La deuxième section se penche sur les réseaux de neurones impulsionnels, en mettant l'accent sur les concepts et les principes qui sous-tendent ces réseaux. Nous expliquons en détail le fonctionnement des neurones impulsionnels, les stratégies existantes pour leur apprentissage, les plateformes matérielles adaptées à leur déploiement, et les verrous posés sur le domaine. Dans le même temps, nous introduisons les formulations à la base de nos travaux.

La troisième section se penche sur la vision événementielle et l'application des réseaux de neurones dans ce domaine spécifique. Nous débutons en exposant la manière dont les caméras événementielles génèrent des événements. Ensuite, nous discutons des différentes méthodes de représentation de ces événements, visant à faciliter leur traitement par les algorithmes sous-jacents. Nous dressons également une liste des problématiques de vision dans lesquelles les caméras événementielles ont été évaluées dans la littérature. De plus, nous effectuons une brève revue des méthodes de l'état de l'art basées sur des réseaux de neurones profonds pour résoudre ces défis visuels. Enfin, nous concluons en évoquant les défis ouverts en vision événementielle que nous abordons dans nos travaux.

2.1 Réseaux de Neurones Artificiels pour la Vision par Ordinateur

Les Réseaux de Neurones Artificiels ("*Artificial Neural Networks*" ou ANNs, en anglais) sont des systèmes de calcul inspirés par les réseaux de neurones biologiques. Les ANNs se composent d'unités interconnectées, ou neurones artificiels, qui modélisent de manière approximative les neurones biologiques. Ces neurones calculent leur sortie grâce à une fonction d'activation non linéaire appliquée à la somme de leurs entrées, telle que la "sigmoid" ou la "Rectified Linear Unit" (ReLU) :

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

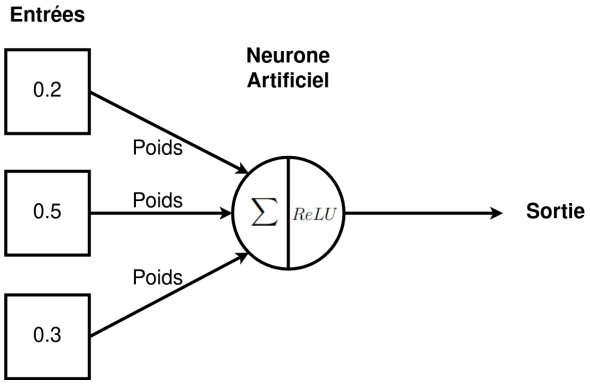
$$ReLU(x) = max(0, x) \tag{2.2}$$

Les connexions entre les neurones, analogues aux synapses, possèdent des *poids* qui modulent la force du signal d'entrée. Ces poids sont optimisés lors de la *phase d'entraînement*, où des exemples d'entrée sont utilisés pour calculer les erreurs entre les prédictions du réseau et les résultats attendus (la "*vérité terrain*"). Le réseau ajuste ensuite ses *paramètres* (poids) à l'aide d'une règle d'apprentissage pour minimiser ces erreurs. Après un nombre suffisant d'itérations, le réseau est considéré comme entraîné et prêt à effectuer des prédictions sur de nouvelles données, ce que l'on appelle l'*inférence*, dans le cadre de l'apprentissage supervisé. Les ANNs sont généralement organisés en couches. La Figure 2.1 montre un aperçu du fonctionnement d'un neurone artificiel (Figure 2.1a) ainsi qu'un exemple d'un réseau organisé en couches (Figure 2.1b).

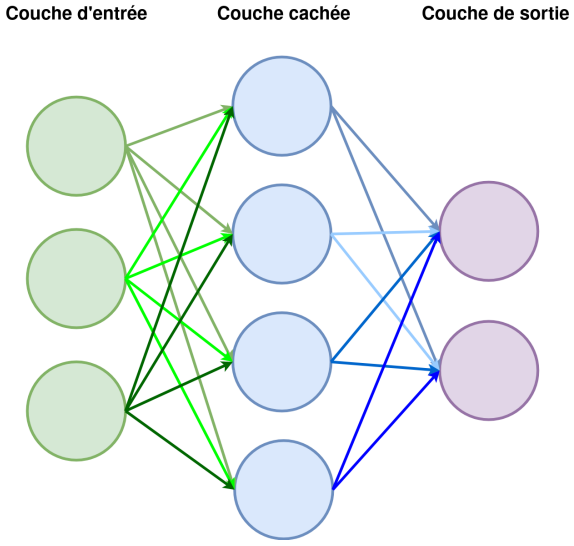
2.1.1 Historique des Réseaux de Neurones Artificiels

Initialement, [MP43] a introduit le modèle computationnel des réseaux de neurones, tandis que [Ros58] a présenté le "perceptron", un réseau de neurones artificiels d'une seule couche. Cependant, ces perceptrons à une seule couche ont montré une capacité limitée, capable uniquement d'apprendre des motifs linéairement séparables, ce qui a entraîné un déclin de l'intérêt pour la recherche sur les ANNs pendant plusieurs décennies.

Dans [HSW89], il a été démontré qu'un perceptron multicouche ("*Multi-Layer Percep-*



(a) Illustration d'un neurone artificiel. Les entrées sont pondérées par les *poids*, des paramètres optimisés par une règle d'apprentissage, puis accumulées. Ensuite, le neurone artificiel calcule sa sortie via une fonction non linéaire d'activation (ici, une fonction ReLU).



(b) Illustration d'un MLP, une architecture de réseau de neurones typique.

Figure 2.1: Aperçu du fonctionnement des réseaux de neurones artificiels.

tion" ou MLP, en anglais) à deux couches pouvait approximer avec précision n'importe quelle fonction non linéairement séparable. Néanmoins, l'entraînement de réseaux de neurones multicouches est resté un défi jusqu'à ce que [RHW86] et [RMP86] introduisent l'algorithme de rétropropagation ("*backpropagation*", en anglais). Cet algorithme calcule efficacement le gradient d'une fonction de coût entre la réponse de l'ANN et la vérité terrain par rapport aux poids du réseau à optimiser.

La forme moderne de l'algorithme d'apprentissage par rétropropagation pour les ANNs a été présentée dans [LF87]. La rétropropagation calcule les gradients des poids à l'aide de la règle de la chaîne du gradient ("*gradient chain rule*", en anglais), en calculant les gradients couche par couche à partir de la dernière pour éviter les calculs redondants de termes intermédiaires dans la règle de la chaîne. Cette méthode est combinée avec des approches d'entraînement basées sur les gradients pour les réseaux multicouches, la plus courante étant la descente de gradient et son approximation stochastique, la descente de gradient stochastique (SGD).

Les modèles basés sur le MLP nécessitent de nombreux neurones et connexions pour des tâches de classification complexes, en particulier avec des données d'entrée volumineuses telles que des images. Inspirés par le cortex visuel des chats [HW59], [Fuk80] a introduit le néocognitron, un ANN utilisant des couches de convolution au lieu des couches entièrement connectées que l'on trouve dans les MLP.

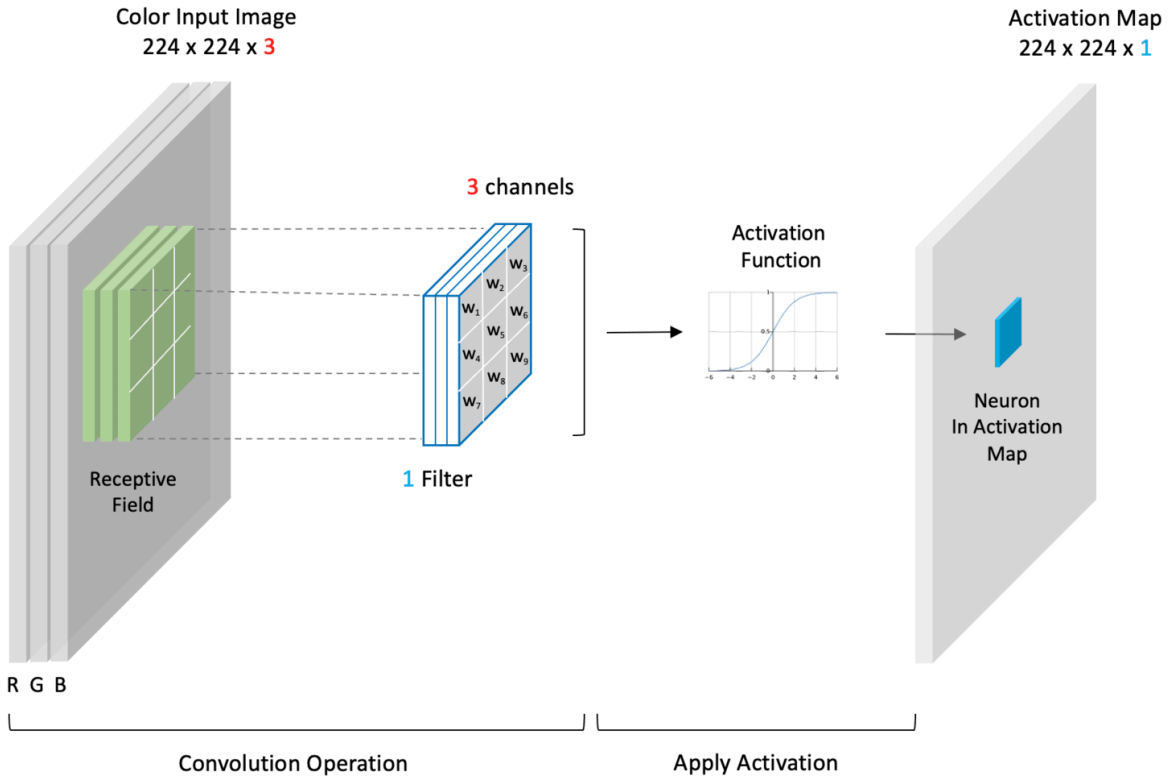


Figure 2.2: Illustration du fonctionnement d'une couche de convolution dans un ANN. Source : https://learnopencv.com/understanding-convolutional-neural-networks-cnn/.

Une couche de convolution (voir la Figure 2.2) utilise des opérations de convolution mathématique. Elle est composée d'unités dont les champs récepteurs couvrent des patchs de la couche précédente. Les poids de l'opération de convolution sont partagés entre plusieurs neurones et sont optimisés par la rétropropagation. Cette approche réduit le nombre de paramètres entraînables et introduit implicitement des invariances en translation [Kau17], ce qui se traduit par de meilleures performances dans les tâches de vision artificielle.

La première démonstration des Réseaux de Neurones Convolutifs ("*Convolutional Neural Networks*" ou CNNs, en anglais) remonte à [LeC+98] avec l'introduction de LeNet-5, un réseau de convolution de 7 couches entraîné par rétropropagation. Il classifiait avec succès des chiffres manuscrits à partir d'images de 32x32 pixels. Néanmoins, la manipulation d'images "naturelles" complexes de plus haute résolution nécessite des CNNs plus profonds et avec plus de paramètres. Le développement de tels CNNs a été fortement empêché à l'époque de LeNet-5 (1987) à cause de la puissance de calcul des ressources informatiques disponibles, trop limitée par rapport

aux besoins des réseaux profonds.

Dans les années 2010, l'abondance de données et la disponibilité de la puissance de calcul (notamment grâce aux processeurs graphiques) ont considérablement amélioré la puissance et l'efficacité des ANNs, y compris les CNNs. Cette tendance a notamment été initiée par [KSH12] en proposant un CNN profond qui a dépassé de loin les performances des autres travaux de l'époque pour la classification d'images sur ImageNet [Den+09] (une base de données de grande envergure). Cette avancée, désormais appelée apprentissage profond ("*Deep Learning*", en anglais), a consisté à entraîner des réseaux avec de plus en plus de couches. Il a constamment surpassé les algorithmes traditionnels dans divers domaines, notamment la vision par ordinateur, la reconnaissance vocale et le traitement du langage naturel.

Les CNNs mentionnés précédemment effectuent des opérations de convolution en deux dimensions, ce qui les désigne comme des 2D-CNNs. Cependant, cette approche n'est pas adaptée au traitement de données spatio-temporelles telles que les vidéos. Pour cette raison, les 3D-CNNs ont été introduits, où les opérations de convolution s'appliquent en trois dimensions (hauteur, largeur et temps) [Tra+15]. L'avènement de ces 3D-CNNs, ainsi que d'autres architectures tels que les réseaux de neurones récurrents [LBE15], a considérablement contribué aux avancées de l'apprentissage profond, en particulier dans le domaine du traitement vidéo.

La normalisation par lots ("*batch normalization*", en anglais) [IS15] et les connexions résiduelles [He+16] ont été introduites pour atténuer le problème de la disparition du gradient ("*vanishing gradient problem*", en anglais). Ce problème a entravé l'entraînement de réseaux profonds avec de nombreuses couches à l'aide de la rétropropagation, car le gradient diminue significativement dans les couches initiales, ce qui affaiblit les mises à jour des poids. En conséquence, les CNNs avec des centaines de couches [Sze+17b] ont obtenu des améliorations remarquables des performances dans diverses tâches de vision par ordinateur, notamment en classification d'images [He20], en détection d'objets [WSH20] et en segmentation [Min+21].

L'une des avancées récentes majeures est l'introduction des transformeurs ("*transformers*", en anglais) [Vas+17], une nouvelle classe de réseaux de neurones initialement conçue pour le traitement du langage naturel ("*natural language processing*" ou NLP, en anglais). Basés notamment sur la décomposition des données en patchs et l'utilisation de mécanismes d'attention, ces transformeurs ont révolutionné le domaine du NLP, réalisant des progrès significatifs [Dev+18; Rad+18; Tou+23]. L'adaptation de cette

architecture au domaine de la vision par ordinateur est appelée les "transformeurs de vision" ("*Vision Transformers*" ou ViTs, en anglais) [Dos+20]. Tout comme en NLP, les ViTs ont montré d'importants progrès dans de nombreuses tâches de vision [Liu+23], émergeant comme un nouveau paradigme dominant pour la conception de réseaux neuronaux profonds pour des tâches de vision à haute performance.

Pour surmonter le besoin en données massives pour entraîner les ViTs, de nouvelles techniques d'apprentissage non supervisé, appelées "apprentissage auto-supervisé" ("*Self-Supervised Learning*" ou SSL, en anglais), ont permis la création de modèles ViTs de grande envergure pré-entraînés sur des volumes massifs de données non annotées [Rad+21; Car+21]. Ces modèles se sont avérés hautement polyvalents pour extraire des informations utiles à partir d'images variées et pour résoudre divers problèmes de vision. Ces modèles de grande envergure, pré-entraînés sur de vastes ensembles de données non annotées, sont désignés sous le terme de "modèles de fondation" ("*Foundation Models*", en anglais).

2.1.2 Formulations

Dans cette section, nous proposons une formulation étendue de l'inférence réalisée par les réseaux de neurones, en particulier le processus d'extraction de caractéristiques. Nous exprimons le fonctionnement d'un réseau de neurones de la manière suivante : un réseau de neurones, quelle que soit son architecture, peut être représenté comme une fonction $f_\alpha(\cdot)$, où l'ensemble α englobe les paramètres ajustables de ce réseau (les poids). Cette fonction est définie comme suit :

$$f_\alpha(Input) = Output \tag{2.3}$$

$Input$ désigne les données en entrée du réseau, et $Output$ représente les caractéristiques produites en sortie. L'adaptabilité des ANNs leur permet de manipuler le format des données $Output$ et $Input$ en fonction de la tâche spécifique qu'ils sont chargés de résoudre.

Par exemple, dans le cas d'un problème de classification [KSH12] avec $\mathcal{C}$ classes, un ANN $f_\alpha(\cdot)$ sous la forme d'un CNN prend en entrée une image RGB $\mathbf{I} \in \mathbb{R}^{C\times H\times W}$ de résolution $(H \times W)$ avec $C = 3$ et génère une prédiction de classe $c \in [1, \mathcal{C}]$. Ainsi, dans notre formulation, nous pouvons identifier $Input = \mathbf{I}$ et $Output = c$

de la manière suivante :

$$f_\alpha(\mathbf{I}) = c \tag{2.4}$$

2.1.2.1 Extraction de Caractéristiques par un Encodeur Convolutif

Les réseaux de neurones profonds présentent un avantage majeur : leur capacité à extraire automatiquement des caractéristiques pertinentes à partir d'une image pour la tâche de vision visée. Cette approche diffère des algorithmes traditionnels [SJE14], qui nécessitent une définition manuelle des caractéristiques à calculer, sans garantie d'optimalité. Les caractéristiques apprises par un CNN s'ajustent de manière adaptative en fonction de la fonction de coût et des données utilisées lors de la phase d'apprentissage.

Il est crucial de souligner que l'efficacité des caractéristiques extraites par un réseau de neurones dépend de la qualité de l'ensemble d'apprentissage. En d'autres termes, si les données de cet ensemble sont de mauvaise qualité, les informations apprises par le réseau seront également de mauvaise qualité. Ainsi, la qualité des données est tout aussi cruciale que le modèle lui-même .

Ce concept d'extraction de caractéristiques joue un rôle central dans nos travaux. Ainsi, nous adoptons la formulation précédemment énoncée dans l'Équation 2.3 pour décrire l'extraction de caractéristiques par un CNN profond, que nous appelons "encodeur convolutif". Qu'il s'agisse d'un 2D-CNN ou un 3D-CNN, un encodeur convolutif peut être interprété comme une fonction, notée $f_\alpha(\cdot)$, qui prend en entrée des données visuelles, telles qu'une image $\mathbf{I}$:

$$f_\alpha(\mathbf{I}) = \mathcal{F} \tag{2.5}$$

où $Output = \mathcal{F} \in \mathbb{R}^K$ représente les informations extraites de l'image sous la forme d'un vecteur de K nombres réels. La Figure 2.3a illustre un exemple d'architecture 2D-CNN qui agit comme un encodeur convolutif. Ce vecteur résultant, souvent appelé "**vecteur de caractéristiques**" (ou "*feature vector*" en anglais), est un concept largement utilisé dans de nombreuses approches d'apprentissage profond [KZS+15; SSZ17; BPL22]. Ensuite, ce vecteur de caractéristiques peut être acheminé vers d'autres modules pour effectuer des prédictions, tel qu'un classifieur linéaire pour la classification d'images. La Figure 2.3b donne un aperçu de ce processus d'extraction

de vecteur de caractéristiques.

Outre l'utilisation de vecteurs, de nombreuses recherches axées sur l'extraction d'informations spatiales à partir d'images [Min+21] privilégient l'extraction de caractéristiques sous une forme à deux dimensions ou plus. Dans ce cas, on parle de **"cartes de caractéristiques"** ("*feature maps*" en anglais) de résolution $(H_{\mathcal{M}} \times W_{\mathcal{M}})$:

$$f_\alpha(\mathbf{I}) = \mathcal{M} \quad \text{où} \quad \mathcal{M} \in \mathbb{R}^{K \times \cdots \times H_{\mathcal{M}} \times W_{\mathcal{M}}} \tag{2.6}$$

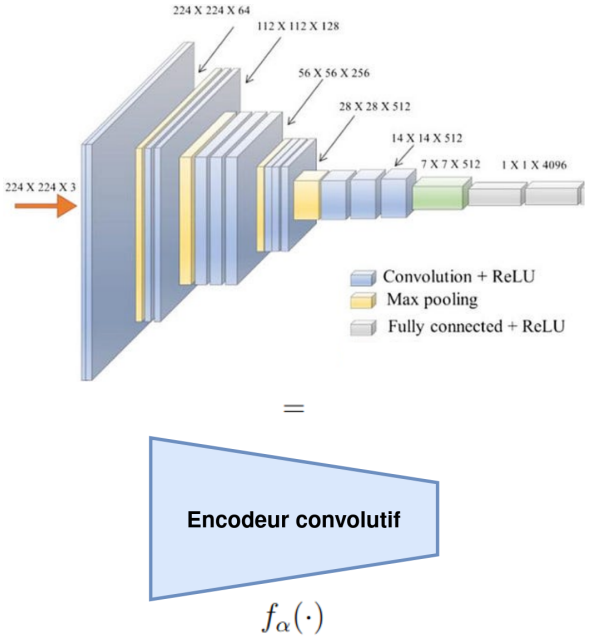
Un exemple du processus d'extraction de cartes de caractéristiques est donné en Figure 2.3c.

2.1.3 Verrous Scentifiques

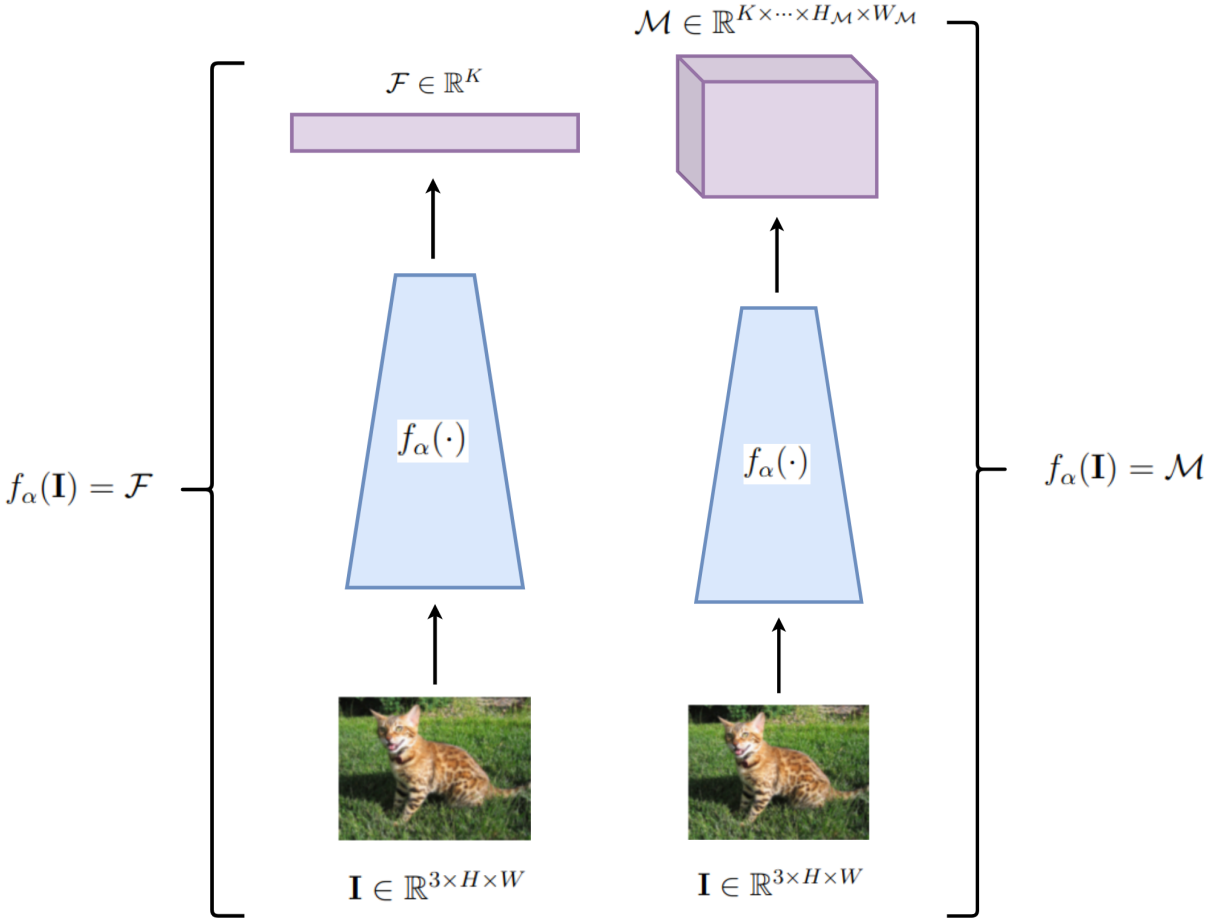
Tout au long de cette Section 2.1, nous avons parcouru les principales avancées dans le domaine des ANNs depuis leur création jusqu'à nos jours, tout en décrivant le processus d'extraction de caractéristiques. À travers cette revue, nous avons mis en évidence la puissance de l'apprentissage profond, qui a permis de développer des ANNs de plus en plus profonds et sophistiqués, capables de traiter des problèmes de vision de plus en plus complexes avec une précision croissante.

Néanmoins, cette augmentation de la complexité des réseaux de neurones s'accompagne d'une demande croissante en termes de puissance de calcul, comme le montre l'évolution illustrée dans la Figure 2.4a. En conséquence, cette demande croissante en ressources informatiques entraîne une consommation d'énergie de plus en plus élevée, comme indiqué dans la Figure 2.4b, ce qui soulève des préoccupations environnementales majeures [Kaa+22].

Une part croissante de la communauté scientifique s'efforce de résoudre ce problème en explorant diverses solutions potentielles, telles que les accélérations matérielles [Lee+18], l'initiative "Green AI" [Sch+20], et d'autres. Dans nos travaux, nous nous concentrons sur l'une de ces solutions, qui s'inspire du fonctionnement des neurones biologiques en utilisant des neurones impulsionnels [Abb99; GK02b].



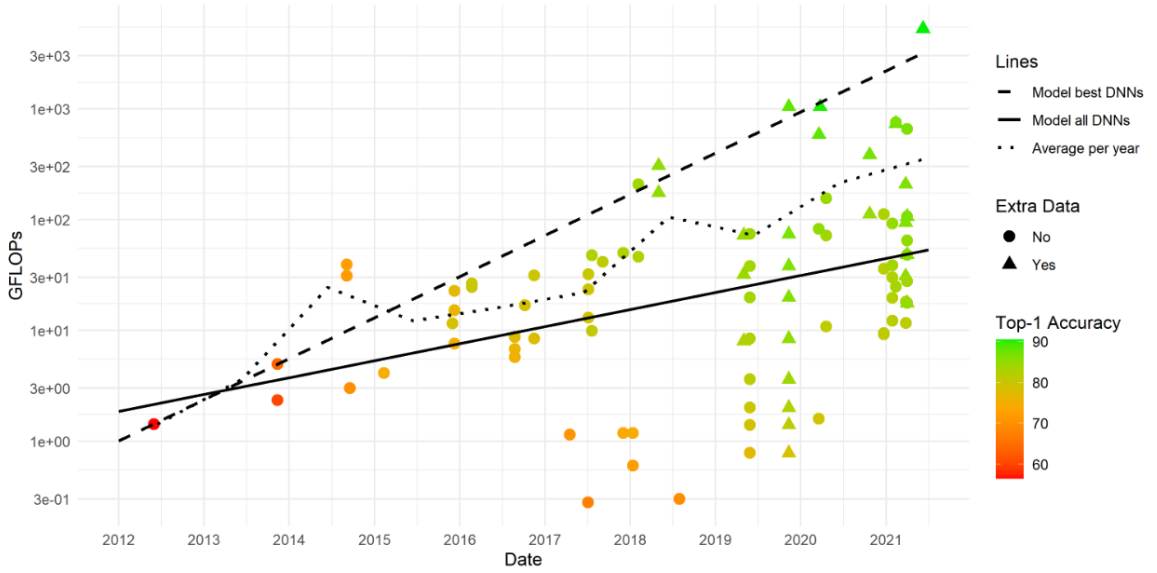
(a) Exemple d'une architecture CNN désignée comme un encodeur convolutif. Le CNN illustré provient de [SZ14].



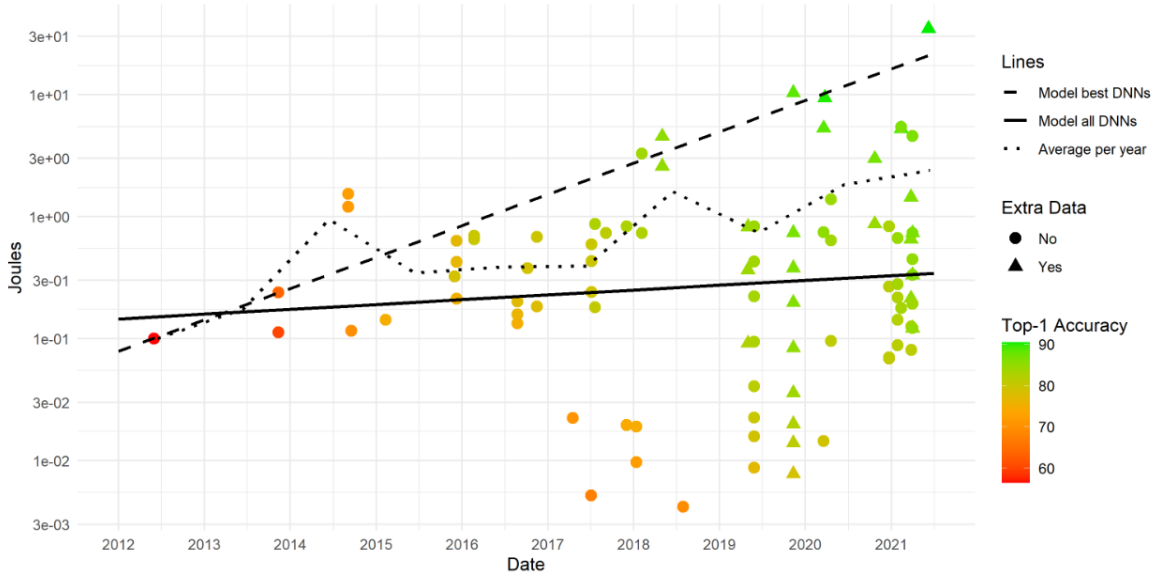
(b) Extraction d'un vecteur de caractéristiques par un encodeur convolutif.

(c) Extraction d'une carte de caractéristiques par un encodeur convolutif.

Figure 2.3: Exemples d'extractions de caractéristiques par un encodeur convolutif.



(a) Puissance de calcul (en GFLOPS) **pour une inférence selon les ANNs et leur année de conception.** On observe une croissance exponentielle du besoin en puissance de calcul au fur et à mesure des années (l'échelle est logarithmique).



(b) Énergie (en Joules) **estimée pour une inférence selon les ANNs et leur année de conception.** La ligne montrant la tendance des meilleurs modèles de chaque année indique clairement un accroissement exponentiel de l'énergie consommée (l'échelle est logarithmique).

Figure 2.4: Évolutions par année des demandes en puissance de calcul et énergie des modèles ANNs de l'état de l'art. Les couleurs (de rouge à vert) représentent le score de précision obtenu par les modèles étudiés sur ImageNet [Den+09]. L'acronyme "*DNN*" dans les graphiques désigne le terme "réseau de neurones profonds". Source : [Des21].

2.2 Réseaux de Neurones Impulsionnels

Les **réseaux de neurones impulsionnels** ("*Spiking Neural Networks*" ou **SNN**s, en anglais) sont un type de réseau de neurones artificiels qui se démarquent des ANNs conventionnels décrits dans la Section 2.1, principalement en raison de leur fonctionnement basé sur des neurones impulsionnels ("*spiking neurons*", en anglais). Les SNNs reproduisent plus fidèlement le mode de communication entre les neurones, similaire à celui des neurones biologiques, par rapport aux neurones artificiels classiques.

Outre l'emploi de neurones impulsionnels, les SNNs suivent les mêmes principes que les ANNs en ce qui concerne la connexion des neurones via des poids synaptiques pour pondérer les informations échangées. Ils peuvent ainsi adopter des architectures en couches similaires aux ANNs, telles que les MLPs [Fal+19], les ViTs [Zho+22], ou les CNNs [Fan+21a]. Lorsqu'un SNN est configuré comme une architecture CNN, on le désigne comme un **SNN convolutif** ("*Convolutional SNN*" ou **CSNN**, en anglais).

2.2.1 Du Neurone Artificiel au Neurone Impulsionnel

Dans cette section, nous décrivons le fonctionnement général d'un neurone impulsionnel en mettant en évidence la différence fondamentale par rapport aux neurones artificiels : la méthode d'échange d'informations au sein du réseau. Parallèlement, nous discutons des avantages en termes d'efficacité de calcul, ce qui se traduit également par une meilleure efficacité énergétique.

Cette distinction fondamentale peut être expliquée à l'aide de trois termes complémentaires [Esh+21] : les **impulsions**, le **traitement basé sur les événements**, et la **sparsité**.

Impulsions. Contrairement aux neurones artificiels, qui échangent de l'information en utilisant des valeurs réelles issues de leur fonction d'activation, les neurones impulsionnels communiquent par le biais de flux asynchrones d'**impulsions** ("*spikes*", en anglais) binaires dans le temps. Ce manuscrit utilise le terme "asynchrone" pour décrire les échanges d'informations entre neurones, où les impulsions sont enregistrées dans des événements distribués sans suivre un rythme régulier défini par une horloge globale, contrairement aux ordinateurs classiques basés sur l'architecture Von Neumann. Par conséquent, ces neurones effectuent intrinsèquement un traitement spatio-temporel de l'information. Les activations des neurones impulsionnels sont ainsi limitées à un seul bit, ce qui réduit la complexité des calculs dans le réseau

par rapport aux neurones artificiels, qui nécessitent des multiplications de nombres à virgule flottante entre les valeurs continues des activations et les poids synaptiques.

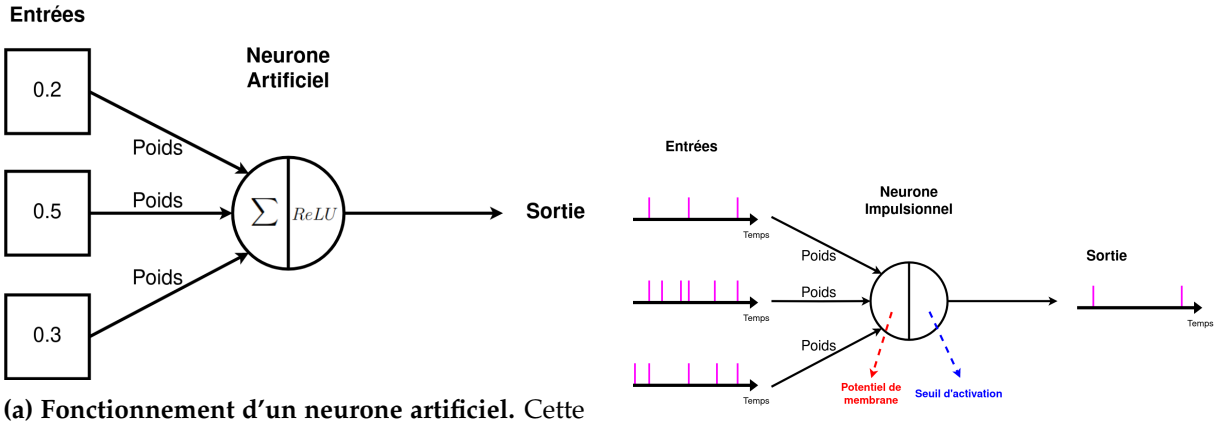
Traitement Basé sur les Événements. Comme les neurones artificiels, un neurone impulsionnel accumule les entrées (provenant par exemple des neurones de la couche précédente) pondérées par des poids synaptiques. Cependant, au lieu d'appliquer une fonction non linéaire à la somme des entrées comme le ferait un neurone artificiel, les neurones impulsionnels disposent d'un état interne appelé "**potentiel de membrane**" ("*membrane potential*", en anglais) qui accumule les impulsions pondérées jusqu'à atteindre un **seuil** d'activation ("*threshold*", en anglais). Une fois ce seuil atteint, le neurone génère une impulsion et son potentiel de membrane revient à son état de repos. En conséquence, un neurone impulsionnel ne réagit qu'aux événements résultant d'un dépassement du potentiel de membrane par les neurones presynaptiques. Cette caractéristique des neurones impulsionnels, qui traitent uniquement les changements d'état pour l'information, permet de réduire considérablement la charge de calcul nécessaire à l'inférence, par opposition aux ANNs, qui requièrent des calculs denses pour l'ensemble du réseau.

Sparsité. À l'instar des neurones biologiques, les neurones impulsionnels sont actifs uniquement lorsqu'ils émettent une impulsion et restent au repos le reste du temps. Ainsi, les calculs effectués dans un SNN lors de l'inférence sont épars, ce qui implique le stockage et le transfert de structures de données binaires et éparses. Ces caractéristiques minimisent les exigences en termes de calcul et de mémoire pour la plateforme matérielle sur laquelle un SNN est déployé, car l'espace mémoire requis dépend uniquement du nombre d'éléments non nuls. En comparaison, un ANN nécessite le stockage de données denses et à haute précision, entraînant des besoins plus importants en termes de mémoire et de puissance de calcul.

La Figure 2.5 présente les fonctionnements des neurones artificiels et impulsionnels, mettant en lumière les différences mentionnées.

2.2.2 Modèles de Neurone Impulsionnel

Avant d'être considérés comme un paradigme de modèle d'apprentissage, les neurones impulsionnels ont été conçus pour imiter le fonctionnement des neurones biologiques à des fins d'analyse dans le domaine de la neuroscience computationnelle. C'est pourquoi il existe de nombreux modèles de neurones impulsionnels créés dans le



(a) Fonctionnement d'un neurone artificiel. Cette Figure est reprise de la Figure 2.1a.

(b) Fonctionnement d'un neurone impulsionnel.

Figure 2.5: Comparaison des fonctionnements des neurones artificiels et impulsionnels.

but d'émuler les neurones biologiques. Dans cette section, nous passons en revue les modèles de neurones impulsionnels existants, et établissons une formulation des modèles retenus pour nos travaux.

2.2.2.1 Revue des Modèles Existants

Au fil des années, plusieurs modèles de neurones ont été proposés pour reproduire les neurones biologiques. Le modèle de neurone pionnier, nommé le modèle Hodgkin-Huxley (HH) [HH52], est un système complexe d'équations différentielles non linéaires qui imite de près le comportement électrique des neurones. Cependant, le coût computationnel de la résolution numérique de ces équations est prohibitif, ce qui a conduit à l'adoption de versions simplifiées pour la modélisation des SNNs.

Dans [Izh04], plusieurs modèles de neurones sont évalués en fonction de leur plausibilité biologique, caractérisée par 20 caractéristiques neurocomputationnelles, et de leur coût computationnel mesuré en opérations en virgule flottante par seconde (FLOPS) nécessaires pour simuler le modèle pendant 1 ms. La Figure 2.6 présente les résultats de cette évaluation. Les modèles ayant une plausibilité biologique plus élevée ont tendance à entraîner des coûts d'implémentation plus élevés. Par conséquent, pour les simulations à grande échelle des SNNs, en particulier dans les applications d'apprentissage automatiques et de vision artificielle avec des millions de neurones, les modèles de neurones plus simples sont préférés.

Au vu des résultats en Figure 2.6, on remarque que le modèle de neurone d'Izhikevich [Izh03] allie une forte plausibilité biologique à une bonne efficacité, ce qui encouragerait son adoption pour des SNNs profonds. Cependant, ce modèle est difficile à utiliser

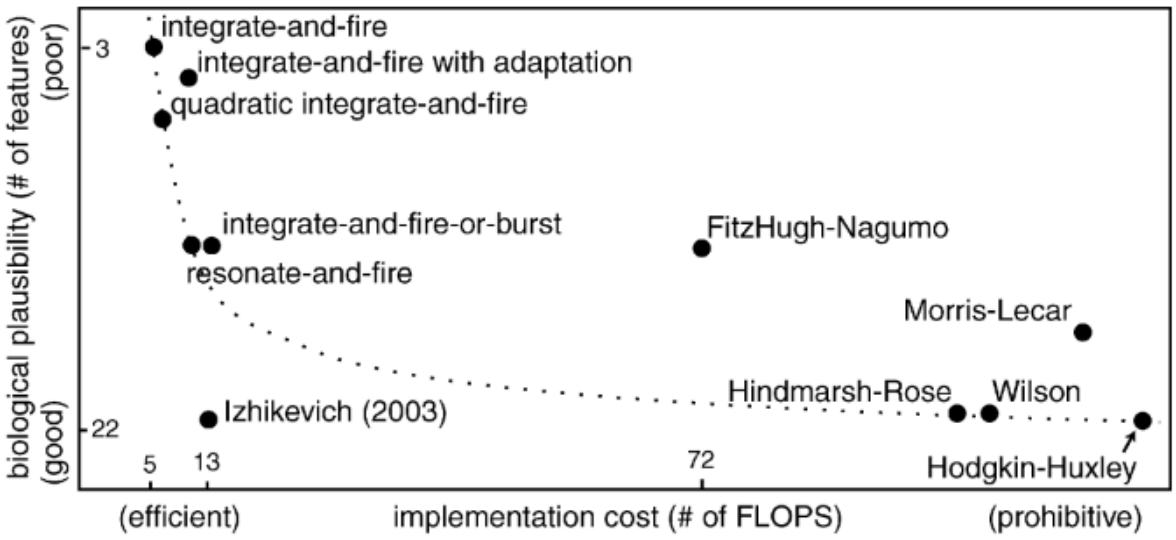


Figure 2.6: Comparaison des modèles de neurones existants en fonction de leur bio-plausibilité et de leur coût d'implémentation. Source : [Izh04]

pour les tâches d'apprentissage profond en raison de sa fonction quadratique de mise à jour du potentiel de membrane [Cor22]. C'est pourquoi ce modèle est rarement employé dans des modèles d'apprentissage.

Dans nos travaux portés sur les SNNs, nous employons des modèles de neurones largement adoptés pour les problématiques de vision artificielle [Fan+21a; Fal+19; Esh+21], à savoir les modèles de neurones "**Integrate-and-Fire**" (IF) et "**Leaky Integrate-and-Fire**" (LIF) introduits par les travaux pionniers de Lapicque en 1907 [Lap07].

2.2.2.2 Formulation du Modèle "Leaky Integrate-and-Fire"

Dans ses travaux pionniers de 1907 [Lap07], bien avant la compréhension des mécanismes de génération des impulsions [HH52], Louis Lapicque a remarqué que le comportement d'un neurone impulsionnel peut être analogiquement comparé à un circuit passe-bas constitué d'une résistance et d'une capacité. En d'autres termes, la dynamique du potentiel de membrane d'un neurone, soumis à un courant d'intensité donné, peut être modélisée comme un "circuit RC" illustré dans la Figure 2.7. Ce modèle suggère que, en plus de l'accumulation des impulsions en entrée et de la génération d'impulsions en cas de dépassement du seuil d'activation (comme expliqué dans la Section 2.2.1), le potentiel de membrane du neurone a tendance à retourner à son état de repos en l'absence de stimulation. Ce phénomène est communément appelé "fuite" ("*leak*", en anglais).

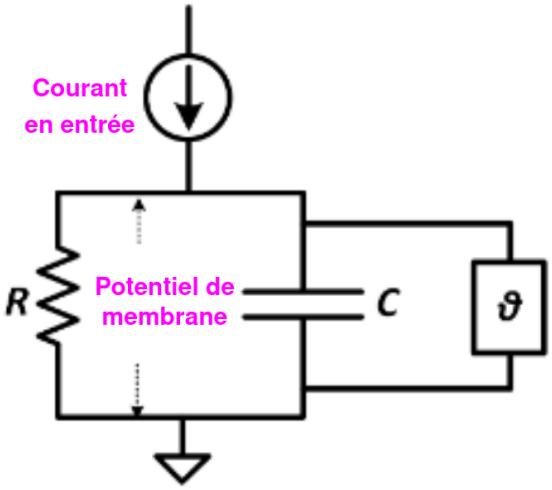


Figure 2.7: Illustration du circuit RC représentant la dynamique du potentiel de membrane d'un neurone LIF. Adapté de [Esh+21].

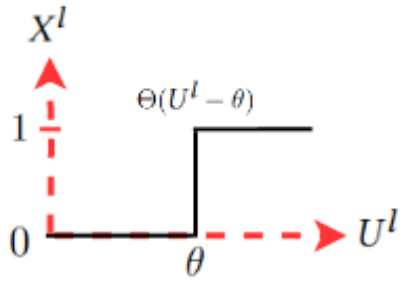


Figure 2.8: Fonction de Heaviside qui représente la relation entre le potentiel de membrane U^l et la génération d'une impulsion X^l.

Dans nos travaux, nous utilisons une forme discrétisée de la dynamique du potentiel de membrane sur un nombre de T étapes temporelles. Comme notre attention se porte sur la conception de SNNs profonds, nous formulons nos concepts en termes de couches de neurones, en évitant la dérivation complète de la dynamique du circuit RC d'origine pour des raisons de simplicité. Une résolution discrète de la dynamique du potentiel de membrane pour une couche de neurones LIF peut être trouvée dans [Esh+21].

La dynamique discrétisée des potentiels de membrane U_t^l d'une couche l de neurones LIF dans un SNN à une certaine étape temporelle $1 \leq t \leq T$ est représentée

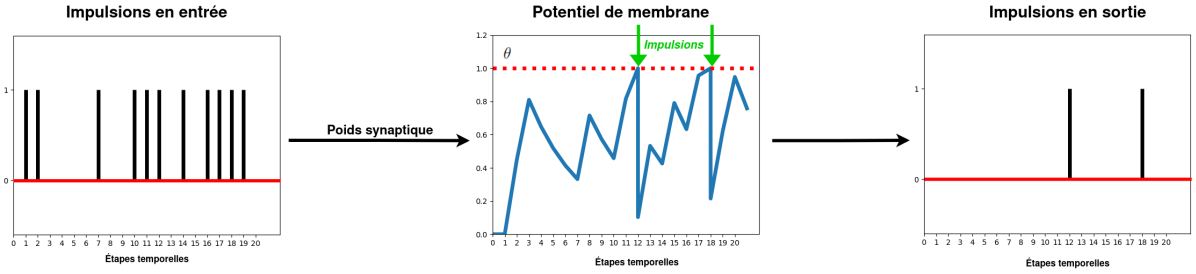


Figure 2.9: Fonctionnement d'un neurone LIF.

de la manière suivante :

$$U_t^l = \beta U_{t-1}^l + \mathcal{W}^l X_t^{l-1} - \theta X_t^l \tag{2.7}$$

$$X_t^l = \Theta(U_{t-1}^l - \theta) \tag{2.8}$$

où $\mathcal{W}^l$ représente l'ensemble des poids synaptiques de la couche l, X_t^l désigne les impulsions émises par la couche l à l'instant t, θ est le seuil d'activation des neurones, et β est le taux de fuite, généralement exprimé comme une valeur réelle comprise entre 0 et 1, influençant le retour du potentiel de membrane vers sa valeur de repos (qui est fixée à 0 dans nos travaux). Le tenseur d'impulsions X_t^l est composé de valeurs égales à 1 lorsque le potentiel de membrane correspondant dans U_t^l dépasse la valeur du seuil θ, et de 0 dans le cas contraire. Dans nos travaux, nous fixons la valeur du seuil à $\theta = 1$. Ce mécanisme de génération d'impulsions est formulé dans l'Équation 2.8 et repose sur la fonction de Heaviside $\Theta(\cdot)$, illustrée en Figure 2.8. Enfin, le retour du potentiel de membrane à sa valeur de repos après la génération d'une impulsion est modélisé par le terme le plus à droite dans l'Équation 2.7. La Figure 2.9 illustre le fonctionnement discrétisé du neurone LIF.

2.2.2.3 Formulation du Modèle "Integrate-and-Fire"

Plus simple que le neurone LIF, le neurone IF possède la même dynamique que ce dernier, à la différence que le neurone IF ne tient pas compte du mécanisme de fuite. Cela se traduit par la fixation d'une valeur $\beta = 1$ dans l'Équation 2.7, tel que :

$$U_t^l = U_{t-1}^l + \mathcal{W}^l X_t^{l-1} - \theta X_t^l \tag{2.9}$$

$$X_t^l = \Theta(U_{t-1}^l - \theta) \tag{2.10}$$

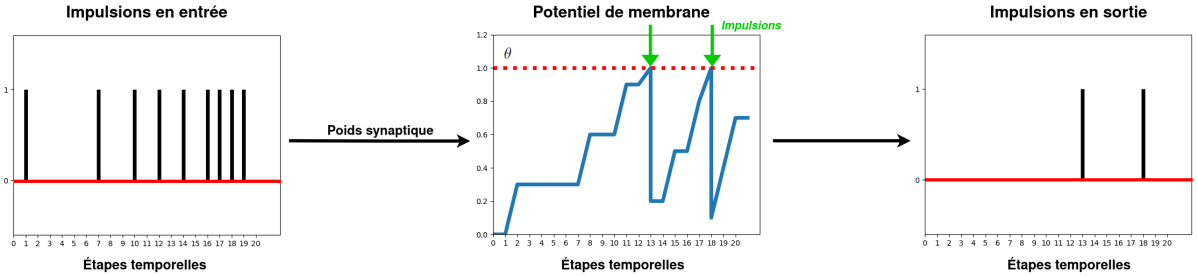


Figure 2.10: Fonctionnement d'un neurone IF.

La Figure 2.10 illustre le fonctionnement discrétisé du neurone IF.

2.2.2.4 Comparaison des Modèles IF et LIF pour l'Apprentissage Profond

En ce qui concerne la conception de SNNs pour l'apprentissage profond et la vision artificielle, le choix entre le modèle LIF et IF dépend principalement du besoin en termes de dynamique de fuite. D'un côté, le neurone IF est légèrement plus optimisé que le neurone LIF, car il élimine le calcul du produit βU_t^l nécessaire pour le mécanisme de fuite dans l'Équation 2.7. D'un autre côté, en supprimant la fuite, les neurones IF sont susceptibles de générer des impulsions même lorsque les entrées ne sont pas corrélées du point de vue temporel.

En ce qui concerne les implications pratiques de l'utilisation du modèle LIF par rapport au modèle IF dans la conception de SNNs profonds, peu de travaux ont exploré cette question. D'une part, l'étude menée par [Bou+22] compare les deux modèles de neurones impulsionnels sur des tâches de reconnaissance telles que [Orc+15] et [Cra+20]. Cette recherche démontre notamment l'intérêt de la dynamique de fuite uniquement lorsque les données d'entrée présentent une structure temporelle riche et que le SNN suit une architecture récurrente. D'autre part, l'étude réalisée par [CLR21] mène des expérimentations similaires et met en évidence que les neurones LIF rendent le SNN plus robuste face à des entrées bruitées tout en améliorant sa capacité de généralisation.

En conclusion, les études existantes [CLR21; Bou+22] indiquent qu'il n'existe pas de réponse claire quant au choix préférable entre les deux modèles de neurones considérés. Ceci explique en partie pourquoi certaines approches de l'état de l'art pour les SNNs profonds tendent à préférer l'usage de neurones IF [Fan+21a; KCP21] tandis que d'autres méthodes emploient des neurones LIF [Fan+21b; KMN20].

2.2.3 Codage Neuronal pour Traiter les Images Statiques

Le terme "image statique" $\mathbf{I} \in \mathbb{R}^{C \times H \times W}$ représente une image capturée par une caméra conventionnelle de résolution $(H \times W)$, où chaque pixel exprime une ou plusieurs valeur(s) réelle(s) (généralement, comprise entre 0 et 1 ou entre 0 et 255). Par exemple, les pixels d'une image couleur RGB expriment des suites de trois valeurs représentant le rouge, le vert et le bleu ($C = 3$) ; tandis que les images en niveaux de gris expriment uniquement une seule valeur ($C = 1$). Comme expliqué en Section 2.2.1, les neurones impulsionnels traitent l'information sous forme de trains d'impulsions (généralement binaires) asynchrones. C'est pourquoi des tenseurs synchrones de valeurs réelles comme les images statiques ne sont pas adaptés au fonctionnement des neurones impulsionnels, empêchant le traitement de ce type de données.

Pour résoudre ce problème d'incompatibilité, la solution réside dans l'utilisation d'une fonction $u(\cdot)$, appelée "**schéma de codage neuronal**" (ou aussi "**codage neuronal**"), prenant en entrée l'image statique $\mathbf{I}$, et produisant des trains d'impulsions visant à encoder les informations visuelles présentes dans l'image. Dans nos travaux, on discrétise ces trains d'impulsions en sortie sur un nombre de T étapes temporelles. Par conséquent, le résultat de cette fonction $u(\cdot)$ est un tenseur représentant ces trains d'impulsions, nommé "**tenseur impulsionnel**" ou "**tenseur d'impulsions**" tel que :

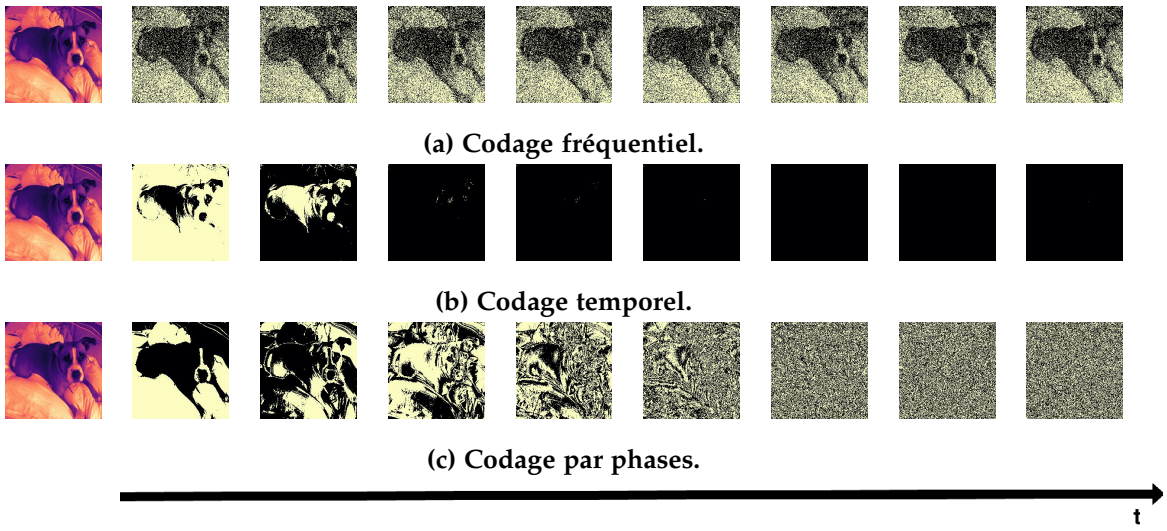
$$\mathbf{X}_T = u(\mathbf{I}) \tag{2.11}$$

où $\mathbf{X}_T \in \mathbb{B}^{T \times C \times H \times W} = \{X_t\}_{t=1}^{T}$ est le tenseur impulsionnel créé.

Dans l'état de l'art, il existe plusieurs codages neuronaux différents [Guo+21], où chacun possède sa propre manière d'exprimer les valeurs des pixels (qu'elle soit plausible biologiquement ou non) avec certains avantages et inconvénients. Dans le reste de cette Section, nous expliquons les schémas de codages neuronaux qui sont considérés dans nos travaux. La Figure 2.12 illustre l'application de chacun des codages neuronaux présentés sur des valeurs de pixel différentes, et la Figure 2.11 montre des exemples des résultats de ces codages neuronaux sur une image statique.

2.2.3.1 Codage Fréquentiel (Rate Coding)

Le codage fréquentiel [AZ26] est une méthode de représentation de l'information sous forme de fréquence d'impulsions. Ce schéma de codage est l'un des plus populaires en raison de la relation observée entre la fréquence des impulsions de certains neurones

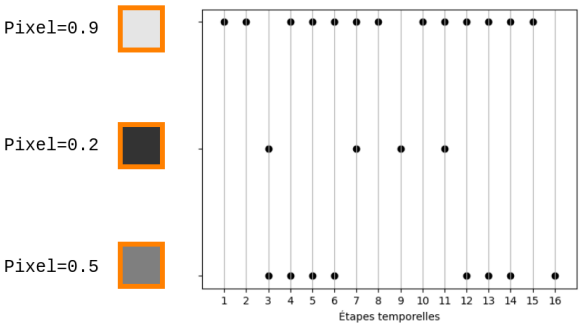


(a) Codage fréquentiel.

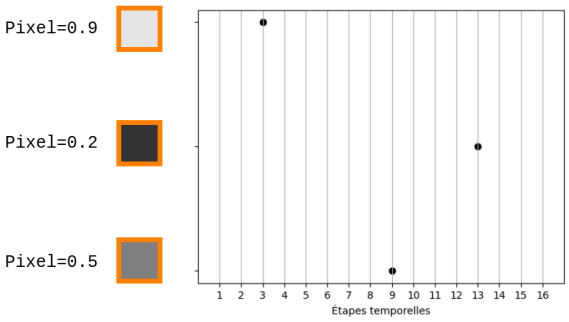
(b) Codage temporel.

(c) Codage par phases.

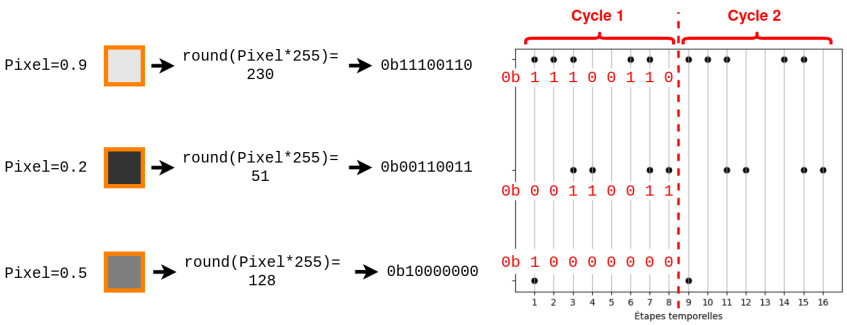
Figure 2.11: Illustration des schémas de codages neuronaux étudiés sur un exemple de Oxford-IIIT-Pet [Par+12].



(a) Codage fréquentiel.



(b) Codage temporel.



(c) Codage par phases.

Figure 2.12: Illustrations des codages neuronaux fréquentiels, temporels et par phases.

biologiques et l'intensité du stimulus, comme cela est observé dans les muscles [AZ26] et le cortex visuel [HW62]. Lorsqu'il s'agit de coder une image statique, la pratique courante consiste à convertir chaque valeur de pixel en une fréquence d'impulsions [Fal19]. Ainsi, un pixel d'intensité élevée se traduit par une fréquence plus élevée que celle d'un pixel de valeur plus faible.

Dans nos travaux, nous utilisons l'implémentation du codage fréquentiel provenant de la bibliothèque SnnTorch [Esh+21]. Chaque valeur de pixel I_{ijk} d'une image statique $\mathbf{I} \in \mathbb{R}^{C \times H \times W}$, normalisée entre 0 et 1, représente une probabilité d'occurrence d'une impulsion à n'importe quelle étape de temps $1 \leq t \leq T$. Cette probabilité suit une distribution de Bernoulli $B(n, p)$, avec $n = 1$ essai et $p = I_{ijk}$ en tant que probabilité de succès (c'est-à-dire, la création d'une impulsion). Par exemple, un pixel blanc ($I_{ijk} = 1$) correspond à une probabilité de 100% de produire une impulsion à chaque étape temporelle, tandis que les pixels noirs ($I_{ijk} = 0$) indiquent qu'aucune impulsion ne se produit. La Figure 2.12a présente le processus de création de ces fréquences pour des valeurs de pixels prises en exemple, et la Figure 2.11a illustre un exemple d'image statique codée en fréquences d'impulsions.

Cependant, ce codage neuronal présente certaines limitations : le nombre d'étapes temporelles T fixé détermine directement l'étendue des fréquences qu'il est possible de produire. De ce fait, le codage d'une large gamme de valeurs de pixels différentes nécessite de produire une grande variété de fréquences d'impulsions, ce qui implique de définir une valeur T plus grande. En d'autres termes, un nombre de T valeurs de pixels différentes nécessite au moins T étapes temporelles. Cette situation introduit une latence dans le réseau, car chaque neurone nécessite du temps pour intégrer plusieurs impulsions avant d'en générer de nouvelles. Certaines études en neuroscience suggèrent que cette latence est incompatible avec les mesures biologiques, comme on peut l'observer dans le cortex visuel [TDV01], où le temps de réponse des neurones est bien plus court que le temps nécessaire pour intégrer des fréquences d'impulsions.

2.2.3.2 Codage Temporel (Temporal Coding)

Le codage temporel repose sur l'idée que l'information est principalement transmise par le timing précis d'une impulsion plutôt que par sa fréquence [TDV01]. Ainsi, une seule impulsion peut représenter une valeur réelle en fonction du moment où elle est émise. Cette approche de codage neuronal présente un avantage significatif par rapport au codage fréquentiel : la représentation de données, comme des images

statiques, ne nécessite que la génération d'un petit nombre d'impulsions (au maximum une par valeur d'entrée à convertir), ce qui garantit une représentation plus légère en termes d'impulsions transmises. En neurosciences, plusieurs études [RR00; TDV01] suggèrent que cet avantage pourrait expliquer la rapidité des neurones biologiques à réagir rapidement aux stimuli [TFM96].

Les deux principales techniques utilisées pour le codage temporel sont le "*Time-To-First-Spike*" (TTFS) et le "*rank order coding*". Dans le TTFS [JB04], l'impulsion qui représente une valeur élevée en entrée est émise plus tôt, tandis qu'une impulsion correspondant à une valeur faible est générée plus tard. En revanche, le rank order coding [TG98] se concentre sur l'ordre d'arrivée des impulsions sans tenir compte du timing précis.

Dans nos travaux, nous utilisons la technique du TTFS comme méthode de codage temporel, en utilisant l'implémentation fournie par la bibliothèque SnnTorch [Esh+21]. L'objectif est d'obtenir le timing précis d'une impulsion, représentant le moment où l'unique impulsion est générée, à partir d'une valeur normalisée de pixel de l'image $\mathbf{I}$. Pour ce faire, nous utilisons une dépendance logarithmique entre la valeur d'intensité d'entrée I_{ijk} et le timing de l'impulsion associée, dérivée à l'aide d'un modèle de circuit RC. Dans ce modèle, la valeur normalisée du pixel d'entrée I_{ijk} est représentée comme une injection de courant constant (les détails sont disponibles dans l'Annexe B.2 de [Esh+21]). En résumé, le timing de l'impulsion en sortie $t(I_{ijk})$ d'un pixel d'entrée est donné par :

$$t(I_{ijk}) = \begin{cases} \tau\left[\ln\left(\frac{I_{ijk}}{I_{ijk}-0.01}\right)\right], & \text{if } I_{ijk} > 0.01 \\ T, & \text{otherwise} \end{cases} \tag{2.12}$$

où $\tau = 1$ est la constante temporelle du circuit RC.

La Figure 2.12b illustre l'application de ce schéma de codage temporel à partir d'exemples de valeurs de pixel, tandis que la Figure 2.11b présente le résultat obtenu en appliquant ce codage temporel à une image statique.

2.2.3.3 Codage par Phases (Phase Coding)

Le codage par phases [Kim+18] vise à diviser la durée totale d'un train d'impulsions en plusieurs phases synchronisées. Ces phases multiples représentent des rythmes d'oscillation observés expérimentalement dans des régions telles que l'hippocampe

et le système olfactif [OR93].

Dans notre étude, nous utilisons une mise en œuvre simple et efficace du codage par phases qui se base sur une représentation en 8 bits des pixels d'entrée non normalisés (c'est-à-dire des valeurs de 0 à 255) [Kim+18]. La séquence de 8 bits ainsi obtenue est utilisée pour créer un cycle d'impulsions sur 8 étapes temporelles : chaque bit ayant une valeur de 1 génère une impulsion à l'étape temporelle correspondante. Par conséquent, un cycle complet de 8 phases est formé (soit 8 étapes temporelles). Ensuite, ces 8 phases sont répétées successivement jusqu'à ce que le nombre d'étapes temporelles T soit atteint. La Figure 2.12c illustre le processus de création de ces phases sur trois valeurs de pixel d'exemple, et la Figure 2.11c montre un exemple d'image convertie par le codage par phases.

Afin de refléter l'importance de chaque bit dans la représentation en 8 bits d'origine, les impulsions créées à une étape temporelle t sont pondérées par des un poids fixe $w_s(t) = 2^{-1+mod(t-1,8)}$. En pratique, cette pondération peut être envisagée comme un ensemble de poids synaptiques appliqués à $\mathbf{X}_T$.

Bien que cette approche simplifiée du codage par phases ne soit pas conforme aux mécanismes biologiques, elle est considérée comme une stratégie pratique pour l'implémentation sur du matériel neuromorphique [Guo+21].

2.2.4 Techniques d'Apprentissages

Comme détaillé dans la Section 2.1, les ANNs doivent une grande partie de leur succès à l'algorithme de rétropropagation [RHW86], qui est actuellement l'approche prédominante pour la conception des algorithmes d'apprentissage profond [LBH15]. En revanche, les SNNs aspirent à être des solutions bio-inspirées et économes en énergie, mais ils font face à des défis quant à la conception d'une règle d'apprentissage aussi performante que la rétropropagation. Alors que les neurones biologiques communiquent via des impulsions, les mécanismes précis d'apprentissage dans des réseaux de neurones biologiques restent une question ouverte, et plusieurs techniques d'apprentissage, plus ou moins plausibles biologiquement, ont été conçues.

Dans cette section, nous passons en revue certains des types principaux de ces règles d'apprentissage pour les SNNs, en mettant en lumière leurs spécificités et en détaillant leurs applications dans les tâches de vision artificielle.

2.2.4.1 Conversion ANN-vers-SNN

En termes généraux, les SNNs peuvent être classés en deux groupes principaux : les ANNs convertis en SNNs et les SNNs directement entraînés. Dans le premier cas, les ANNs traditionnels sont soumis à un apprentissage complet à l'aide de la rétropropagation puis sont convertis en un modèle équivalent composé de neurones impulsionnels. Cette approche implique souvent la transformation des sorties continues des ANNs conventionnels en trains d'impulsions à l'aide d'un codage fréquentiel [Die+15]. D'autre part, les SNNs directement entraînés sont formés à l'aide de principes d'apprentissage biologiquement plausibles ou d'approximations permettant la rétropropagation, tout en exploitant pleinement les capacités des neurones impulsionnels.

Les ANNs convertis en SNNs atteignent généralement des niveaux de performance comparables à ceux des ANNs de l'état de l'art, bien que ces performances restent en deçà de l'originale [Sen+19a; Tav+19; Din+21]. Cette différence de performance peut être attribuée à l'hypothèse selon laquelle une fréquence d'impulsions des SNNs est équivalente à l'activation des ANNs, ce qui n'est pas toujours exact et peut introduire des erreurs. De plus, cette approche d'apprentissage présente plusieurs inconvénients, notamment le manque de bio-plausibilité et les limitations dans la mise en œuvre d'opérateurs ANN essentiels pour améliorer les performances du réseau, tels que le max-pooling, la normalisation par lots ou la fonction d'activation softmax [Rue+17]. Par conséquent, la conversion des ANNs en SNNs implique de nombreuses approximations qui réduisent la généralisation des méthodes de conversion [Rue+17]. Pour remédier à cela, des efforts ont été déployés pour parvenir à une conversion quasi sans perte de performances [RL18; HSR20; Wan+22b].

Un autre inconvénient est lié au codage fréquentiel des activations, où les coûts de calcul augmentent linéairement avec les fréquences d'impulsions. En conséquence, le SNN converti nécessite d'intégrer un grand nombre d'impulsions, ce qui se traduit par une grande latence temporelle. Son efficacité peut être compromise, en particulier dans les architectures profondes ou dans les scénarios impliquant de nombreux neurones très actifs. À cause de cette haute activité des impulsions, certains travaux montrent que le SNN converti en devient moins efficace en termes de consommation énergétique que les ANNs [Dav+21], ce qui est le contraire du but recherché.

Un inconvénient majeur de l'apprentissage basé sur la conversion des activations d'un ANN en fréquences d'impulsions d'un SNN réside dans la limitation des données d'entrée du SNN à celles pouvant être encodées en fréquences. Bien que cela ne

pose généralement pas de problème pour le traitement d'images statiques grâce au codage fréquentiel (comme décrit dans la Section 2.2.3.1), cela rend l'application de la conversion ANN-vers-SNN inutile pour traiter d'autres types de flux d'impulsions, tels que ceux générés par les caméras événementielles.

En ce qui concerne l'application de la conversion ANN-vers-SNN dans le domaine de la vision artificielle, la classification d'images est l'un des contextes les plus populaires dans lesquels ce paradigme d'apprentissage est utilisé. Par conséquent, de nombreuses méthodes de conversion se concentrent sur le développement de CSNNs profonds pour résoudre des problèmes de classification d'images statiques [CCK15; HE15; HE16; Zha+22a]. En outre, cette approche s'est avérée efficace pour traiter des tâches plus avancées, telles que la détection d'objets [Kim+20], la détection d'objets en 3D [ZW18], ou la segmentation sémantique [KCP21], en reproduisant des SNNs de même architecture profonde que les ANNs d'origine.

Malgré ses avantages dans la conception de CSNNs profonds et sa capacité à traiter des problèmes de vision plus avancés que la simple classification, la conversion ANN-vers-SNN présente des inconvénients qui limitent son utilisation dans le cadre de nos travaux axés sur l'efficacité énergétique et le traitement d'entrées qui ne sont pas basées sur des fréquences (codages neuronaux alternatifs ou flux d'événements de caméras événementielles).

2.2.4.2 Règles d'Apprentissage Bio-plausibles

De nombreuses recherches [GK02a] s'efforcent de comprendre et de reconstituer les mécanismes d'apprentissage des neurones biologiques en vue d'exploiter cette bio-plausibilité pour former des SNNs à résoudre des tâches d'apprentissage. Les techniques qui suivent cette intuition sont souvent désignées sous le terme de "règles d'apprentissage hebbien" ("*hebbian learning rules*", en anglais) [Heb05], car elles s'inspirent de l'intuition formulée par Donald Olding Hebb : "**Les neurones qui s'excitent ensemble se connectent entre eux**" ("*cells that fire together, wire together*", en anglais). La famille de règles d'apprentissage hebbien la plus étudiée découle de la règle appelée "Plasticité en fonction du timing des impulsions" ("*Spike-Timing Dependent Plasticity*" ou STDP, en anglais) [SMA00].

La règle STDP [SMA00; Toy+04] est une règle d'apprentissage non supervisée biologiquement plausible qui ajuste les poids synaptiques en fonction du délai entre l'émission d'une impulsion par les neurones présynaptiques et postsynaptiques. Il

convient de souligner que la STDP est locale, ce qui signifie que les mises à jour des poids dépendent uniquement des informations provenant des neurones voisins, ce qui la rend appropriée pour une implémentation sur du matériel neuromorphique [Dav+18]. En substance, la règle STDP repose sur le principe que si une impulsion de sortie suit de près une impulsion d'entrée, la force de la connexion synaptique est renforcée, tandis que si une impulsion d'entrée ne parvient pas à déclencher une impulsion de sortie, la force de la connexion synaptique est affaiblie.

Cependant, à ce jour, les SNNs optimisés avec la règle STDP ne parviennent pas à atteindre les mêmes performances que les approches d'apprentissage supervisé [Sri+20]. En particulier, une cause peut être identifiée : la STDP permet l'entraînement efficace de SNNs peu profonds, mais elle ne converge pas lorsque le réseau entraîné dépasse quelques couches. Cette limitation de profondeur se traduit par une réduction de l'expressivité du réseau de neurones, ce qui limite son applicabilité pour résoudre des problèmes complexes du monde réel [BSF07; Zha+14; SPR18]. De plus, la plupart des SNNs optimisés via STDP pour la vision artificielle se composent généralement de deux couches de convolution [Moz+18], voire d'une seule [FTB20], et se limitent principalement à des tâches de classification d'images simples sur de petites bases de données, telles que la reconnaissance de chiffres manuscrits sur la base de données "MNIST" [LeC+98].

Pour répondre à ces limitations, de nombreux travaux tentent d'améliorer l'apprentissage par STDP, notamment en adaptant plusieurs autres mécanismes observés en biologie, tels que l'inhibition des synapses [Que+13], l'homéostasie [Car+13], la plasticité intrinsèque [ZL19], etc. D'autres travaux cherchent à créer des variantes de la STDP afin d'améliorer l'optimisation des SNNs, comme la STDP modulée par une récompense inspirée de l'apprentissage par renforcement ("Reward-modulated STDP") [Moz+18].

Bien que les progrès récents dans le domaine de l'apprentissage bio-plausible [Khe+18; Fal+19] laissent entrevoir la possibilité d'entraîner efficacement des SNNs profonds, les solutions actuellement disponibles demeurent insuffisantes pour résoudre des problèmes de vision artificielle au-delà de la simple classification sur des données simplistes telles que MNIST [LeC+98] et CIFAR-10 [KH+09]. Par conséquent, ce paradigme d'apprentissage ne peut pas être appliqué dans le cadre de nos travaux qui visent à résoudre des tâches plus complexes.

2.2.4.3 Rétropropagation basée sur les Impulsions

En raison du succès de la rétropropagation du gradient dans le contexte des ANNs, l'adaptation de cet algorithme d'apprentissage pour les SNNs suscite un vif intérêt au sein de la communauté scientifique [Tav+19]. Cette adaptation offre la possibilité d'atteindre des performances élevées dans l'apprentissage supervisé de SNNs profonds, suivant la tendance des avancées réalisées pour les ANNs.

Cependant, comme exposé dans la Section 2.2.2, le mécanisme de génération d'une impulsion peut être décrit mathématiquement comme une fonction de Heaviside (comme illustré dans la Figure 2.8). Or, la fonction de Heaviside n'est pas une fonction différentiable. Cette non-différentiabilité des impulsions pose un défi significatif quant à l'application directe de la rétropropagation du gradient aux SNNs. En conséquence, plusieurs travaux de recherche [BKLoo; NMZ19; Tav+19] se concentrent sur l'élaboration de solutions à ce problème afin de rendre l'apprentissage supervisé des SNNs profonds réalisables.

La première approche [BKLoo] visant à adapter la rétropropagation pour les SNNs est connue sous le nom de "*SpikeProp*". Cette approche repose sur l'intuition que, bien que l'impulsion elle-même ne soit pas une fonction différentiable, son timing est un processus continu et donc potentiellement différentiable. Par conséquent, il devient possible de calculer la dérivée du timing de l'impulsion par rapport aux poids synaptiques du SNN, ce qui permet l'application de la rétropropagation avec des résultats satisfaisants. Cependant, il est important de noter que SpikeProp présente une limitation importante. Il nécessite que chaque neurone émette effectivement une impulsion pour calculer un gradient. Par conséquent, l'absence d'impulsions entraîne un gel de l'entraînement, ce qui limite son applicabilité dans certaines situations. Des développements ultérieurs de cette méthode basée sur le timing précis des impulsions, ont montré des progrès significatifs dans la conception de CSNNs profonds supervisés [Zha+20; Xu+13; Jin+23; WP21]. Ces approches, regroupées sous le terme de "rétropropagation événementielle" ("*event-based backpropagation*", en anglais), continuent de bénéficier de développements importants [WP21].

En parallèle des avancées dans le domaine de la rétropropagation événementielle, une autre approche a émergé pour adapter l'algorithme de rétropropagation aux SNNs. Cette approche a gagné considérablement en popularité, en particulier pour la conception de SNNs profonds, devenant ainsi la technique la plus répandue pour aborder des problèmes d'apprentissage, notamment en vision artificielle. Cette méthode, connue

sous le nom d'"apprentissage par substitut du gradient" ("*surrogate gradient learning*", en anglais) [NMZ19; SO18; ZG18], consiste à utiliser une approximation différentiable de la dérivée de la fonction de Heaviside lors de la phase de rétropropagation. Certaines études [Nun+22; Yam+22] ont démontré que cette approche d'apprentissage atteint des très hautes performances. En conséquence, il est désormais possible de concevoir des SNNs profonds qui rivalisent en complexité et en profondeur avec les modèles ANNs de l'état de l'art, voire qui les surpassent dans certaines situations [ICL21]. Ces avantages permettent la création de CSNNs profonds capables de s'attaquer aux mêmes tâches de vision avancées que les ANNs [Zou+23a; Zhu+22a]. Dans le cadre de nos propres travaux, l'apprentissage par substitut du gradient semble être une approche appropriée pour atteindre nos objectifs dans le développement de CSNNs profonds. Par conséquent, nous l'adoptons pour entraîner nos modèles CSNNs conçus dans nos analyses. Une explication plus détaillée de cette méthode d'apprentissage est décrite dans la Section 2.2.5.

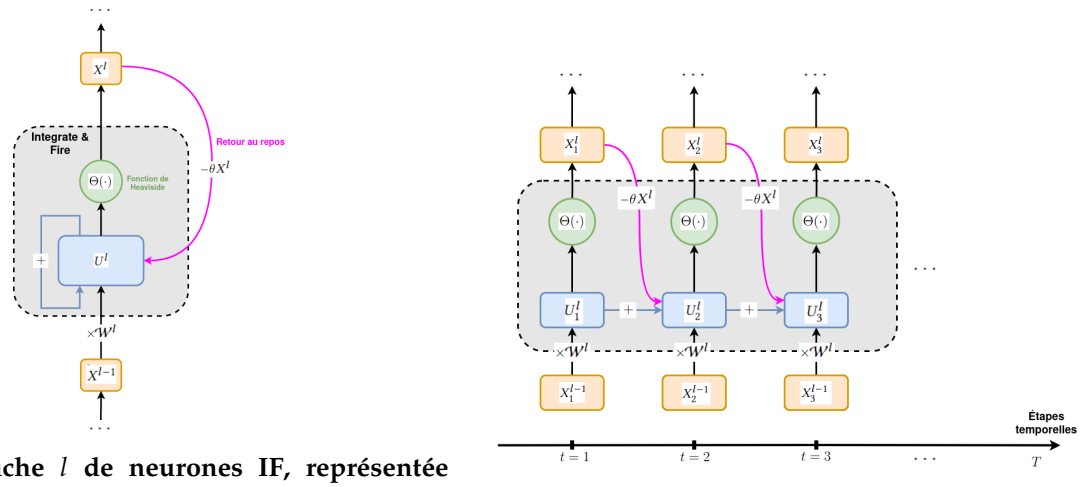
2.2.5 Apprentissage par Substitut du Gradient

Cette section est consacrée à une explication de la méthode d'apprentissage utilisée dans nos travaux portés sur les CSNNs profonds, à savoir l'apprentissage supervisé par substitut du gradient [NMZ19]. À travers nos explications, nous réalisons une formulation élargie de la rétropropagation appliquée aux SNNs, en présentant la problématique de non-différentiabilité des impulsions et sa solution sous la forme du substitut du gradient. Pour ce faire, nous nous basons sur la formulation des neurones IF établie en Section 2.2.2.3. Enfin, nous passons en revue les applications en vision artificielle de SNNs profonds entraînés par cette méthode, afin d'illustrer son efficacité.

2.2.5.1 Les Neurones Impulsionnels en tant que Réseau de Neurones Récurrents

Dans [NMZ19], il a été démontré qu'un SNN peut être exprimé comme un cas spécifique d'un RNN couvrant T étapes temporelles, dont l'état interne est représenté par les potentiels de membranes. De ce fait, il est possible d'employer les mêmes méthodes d'apprentissage pour un SNN que pour un RNN, et plus précisément d'utiliser la version de la rétropropagation adaptée aux RNNs : la **rétropropagation à travers le temps** ("*BackPropagation Through Time*" ou **BPTT**, en anglais) [Wer90].

Nous illustrons cette expression d'un SNN en RNN sur une couche l de neurones IF. Originellement, cette couche de neurones impulsionnels est considérée comme un



(a) La couche l de neurones IF, représentée comme un graphe dirigé à connexions récurrentes.

(b) Graphe de calcul déroulé de la couche l de neurones IF.

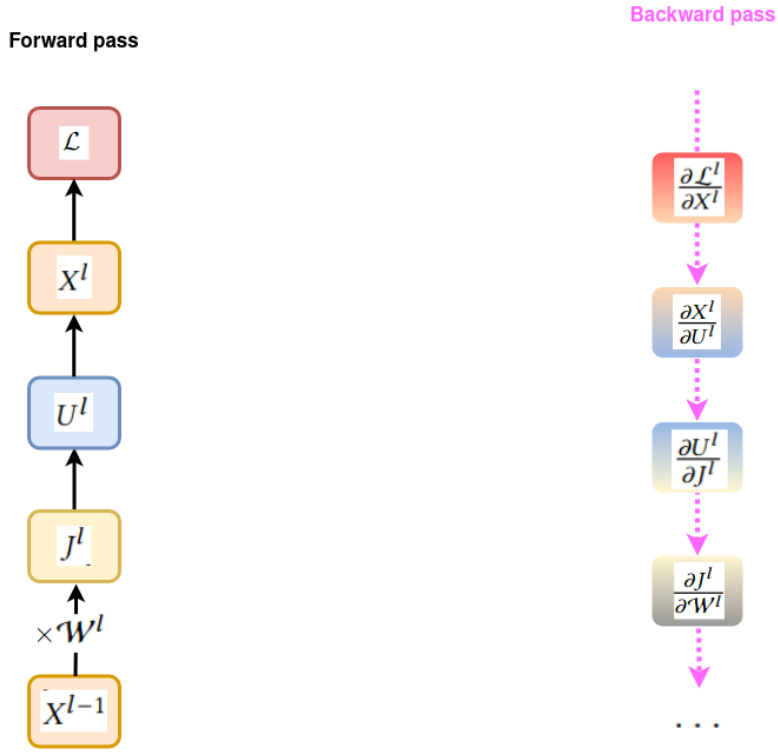
Figure 2.13: Différentes modélisations d'une couche l de neurones IF. En déroulant la vue simplifiée du graphe dirigé à connexions récurrentes (Fig. 2.13a), la dimension temporelle est restaurée, menant à une représentation des neurones IF en tant que RNN (Fig. 2.13b).

graphe dirigé à connexions récurrentes, comme présenté en Figure 2.13a où les variables des neurones IF sont exprimées en étant abstraites de leur dimension temporelle discrétisée (c'est-à-dire, $X_t^l \to X^l$, $U_t^l \to U^l, \dots$). Le déroulement de ce graphe de calcul selon une dimension temporelle (en T étapes temporelles) résulte en une interprétation de cette couche de neurones l en un RNN, comme illustré en Figure 2.13b.

C'est à partir de cette représentation discrétisée des couches de neurones impulsionnels qu'il est possible d'appliquer la BPTT de la même manière que cette dernière est utilisée pour entraîner des RNNs.

2.2.5.2 Surmonter la Non-différentiabilité des Impulsions

La représentation de neurones impulsionnels en tant que RNN s'avère être une approche prometteuse pour bénéficier des avancées dans l'entraînement des RNNs et des modèles basés sur des séquences. Cette représentation des SNNs facilite l'utilisation de la BPTT pour l'entraînement supervisé de modèles SNN profonds avec une fonction de coût $\mathcal{L}$ définie. De plus, l'utilisation répandue de l'algorithme BPTT dans les RNNs et sa mise en œuvre efficace dans les librairies de différenciation automatique (comme PyTorch [Pas+19]) soulignent son efficacité pour atteindre des performances de pointe [LBE15]. Pour des raisons de brièveté, une explication détaillée de l'algorithme BPTT est omise dans cette section, qui se concentre plutôt sur un défi fondamental : adapter l'algorithme BPTT aux neurones impulsionnels, en se penchant plus spécifiquement

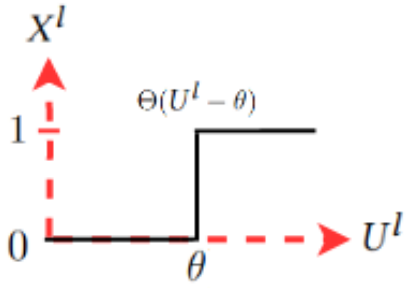


(a) Phase de propagation (forward pass). **(b) Phase de rétropropagation** (backward pass).

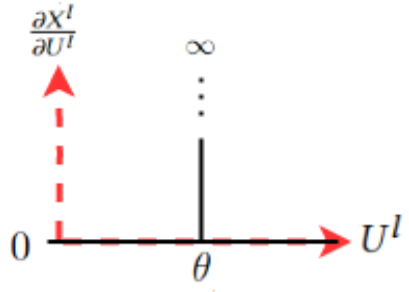
Figure 2.14: Vues simplifiées des phases de propagation et de rétropropagation pour une couche l de neurones IF au sein d'une même étape temporelle t. Pour des raisons de simplicité de présentation, nous définissons $J^l = X^{l-1}W^l$.

sur la non-différentiabilité de la fonction de génération des impulsions.

En résumé, lors de la phase de propagation, la BPTT traite les données d'entrée séquentiellement le long de l'axe temporel dans le SNN, accumulant les états internes sur T étapes temporelles. La Figure 2.14a illustre de manière simplifiée la phase de propagation pour une couche de neurones IF sur une étape temporelle donnée. Lors de la phase de rétropropagation subséquente, les gradients sont calculés pour les états accumulés, ce qui permet de mettre à jour les paramètres du réseau en fonction de la fonction de coût spécifiée $\mathcal{L}$. La Figure 2.14b représente de manière schématique la phase de rétropropagation guidée par la règle de la chaîne de gradients qui découle de la phase de propagation illustrée dans la Figure 2.14a. Cette vue simplifiée de la phase de rétropropagation établit le gradient de la fonction de coût par rapport aux poids synaptiques du réseau, en vue de l'optimisation visant à minimiser cette fonction de coût.



(a) La fonction de Heaviside qui caractérise la relation entre X^l et U^l (voir l'Équation 2.9 ou 2.7).



(b) Dérivée de la fonction de Heaviside, requise pour calculer $\frac{\partial X^l}{\partial U^l}$ dans la BPTT appliquée aux SNNs (voir l'Équation 2.13).

Figure 2.15: Illustration du problème de non-différentiabilité du mécanisme de génération des impulsions. Durant la phase de propagation (Fig. 2.15a), la fonction de Heaviside est utilisée pour modéliser la relation entre les potentiels de membrane U^l et la création des impulsions X^l. Durant la phase de rétropropagation de la BPTT (Fig. 2.15b), la dérivée de cette fonction de Heaviside est une fonction de Dirac dont la valeur est o en tout point sauf au niveau du seuil θ, brisant ainsi la chaîne des gradients.

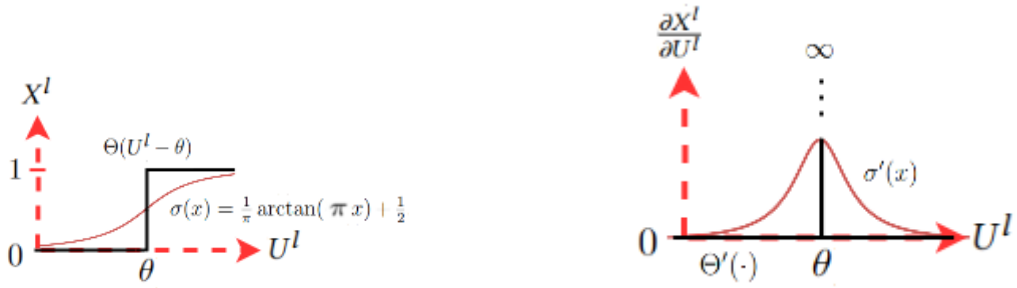
$$\frac{\partial \mathcal{L}}{\partial \mathcal{W}^l} = \frac{\partial \mathcal{L}}{\partial X^l}\frac{\partial X^l}{\partial U^l}\frac{\partial U^l}{\partial J^l}\frac{\partial J^l}{\partial \mathcal{W}^l} \tag{2.13}$$

où on définit $J^l = X^{l-1}\mathcal{W}^l$ pour simplifier la présentation.

Le principal obstacle, à savoir la non-différentiabilité des impulsions, se concentre sur le terme $\frac{\partial X^l}{\partial U^l}$, qui nécessite le calcul de la dérivée de la fonction de Heaviside de l'équation 2.7. Comme illustré dans la Figure 2.15, la dérivée de la fonction de Heaviside correspond à la fonction delta de Dirac, qui s'annule partout sauf au seuil, où elle tend vers l'infini. Par conséquent, les gradients sont principalement annulés, ce qui entrave l'apprentissage efficace par l'intermédiaire de la BPTT dans l'ensemble du SNN. Ce problème est couramment désigné sous le nom de "*problème du neurone mort*" ("*dead neuron problem*", en anglais) [Esh+21].

2.2.5.3 Résoudre la Non-différentiabilité des Impulsions

Afin de résoudre le problème de la non-différentiabilité de la fonction de Heaviside, une solution largement utilisée, décrite dans [NMZ19] et connue sous le nom de "substitut du gradient" ("*surrogate gradient*", en anglais), consiste à maintenir le comportement de la fonction de Heaviside lors de la phase de propagation, mais à remplacer le terme de dérivée $\Theta'(\cdot)$ par la dérivée d'une autre fonction différentiable $\sigma(\cdot)$ lors de la phase de rétropropagation. Malgré les inexactitudes introduites par ce substitut



(a) $\Theta(\cdot)$ (en noir) **vs** $\sigma(\cdot)$ (en rouge). **(b)** $\Theta'(\cdot)$ (en noir) **vs** $\sigma'(\cdot)$ (en rouge).

Figure 2.16: Comparaison entre la fonction de Heaviside $\Theta(\cdot)$ et la fonction $\sigma(\cdot)$ utilisée pour calculer le substitut du gradient durant la rétropropagation lors de la BPTT.

lors de la rétropropagation, il s'avère que, en pratique, les SNNs sont robustes à de telles approximations, car cela ne les empêche pas d'atteindre de bonnes performances dans des problèmes d'apprentissage, y compris en vision artificielle.

Diverses fonctions peuvent être utilisées comme $\sigma(\cdot)$, telles que les fonctions sigmoïde ou tangente hyperbolique (`tanh`). Dans cette étude, nous utilisons $\sigma(x) = \frac{1}{\pi}\arctan(\pi x) + \frac{1}{2}$. La Figure 2.16 présente une comparaison entre $\sigma(\cdot)$ et la fonction de Heaviside originale $\Theta(\cdot)$, ainsi que leurs dérivées, mettant en évidence leurs différences et leurs similitudes. En essence, la fonction $\sigma(\cdot)$ désignée peut être considérée comme une version lissée de $\Theta(\cdot)$, ce qui permet le calcul de $\frac{\partial X^l}{\partial U^l}$.

Par conséquent, cette approche de substitut du gradient permet de résoudre efficacement le problème du neurone mort, rendant possible l'entraînement d'un SNN via la BPTT.

2.2.5.4 Apprentissage par Substitut du Gradient en Vision Artificielle

En utilisant la BPTT en conjonction avec le substitut du gradient pour former des SNN, il devient possible de concevoir des architectures SNN profondes efficaces pour résoudre une vaste gamme de problèmes de vision artificielle [GHM23].

D'un point de vue conceptuel, de nombreuses recherches antérieures ont porté sur le développement d'architectures profondes inspirées par les ANNs de l'état de l'art, notamment des CSNNs [Fan+21a; CMT22], voire des ViTs à base de neurones impulsionnels [Zha+22b; Zho+22; Zho+23; Yao+23]. Le Tableau 2.1 répertorie de telles architectures SNN et établit des correspondances avec les modèles ANN d'origine. D'autre part, d'autres travaux exploitent la polyvalence de la BPTT pour introduire des modules complémentaires aux couches de neurones impulsionnels, tels que des mécan-

Modèle(s) CSNN(s)	ANN(s) correspondant(s)	# couches
[Sen+19b]	VGG [SZ14], ResNet [He+16]	VGG : 7, 9 ResNet : 7, 9, 11
tdBN ResNet [Zhe+21]	ResNet [He+16]	19, 34, 50
SEW-ResNet [Fan+21a]	ResNet [He+16]	18, 34, 50, 101, 152
NDA [Li+22b]	VGG [SZ14], ResNet [He+16]	VGG: 11 ResNet: 19
[CMT22]	VGG [SZ14], MobileNet [How+17], DenseNet [Hua+17]	VGG : 11, 13, 16 MobileNet : 16, 32, 64 DenseNet : 121, 169

Table 2.1: Liste d'architectures CSNNs entraînées par le substitut du gradient, adaptées de modèles ANN existants de l'état de l'art.

ismes d'attention [Yao+21], la distillation des connaissances [Xu+23], l'apprentissage actif [Zha+23], des paramètres optimisés par rétropropagation [Fan+21b; Tan+22c], ...

Grâce à ces diverses architectures de CSNNs ou ViTs impulsionnels, de nombreuses tâches de vision impliquant des images statiques ont été traitées avec succès, obtenant des performances compétitives à celles des ANNs. Au-delà des problèmes de classification sur de petites bases de données telles que MNIST [LeC+98], CIFAR-10 [KH+09], et autres, les CSNNs formés à l'aide du substitut du gradient [Fan+21a; Zhe+21] ont démontré leur capacité à obtenir de bonnes performances en matière de classification à grande échelle sur des ensembles de données d'images statiques naturelles tels qu'ImageNet [Den+09]. En plus de la classification, d'autres tâches de vision ont été explorées, telles que la segmentation sémantique [KCP21] ou la détection d'objets [Qu+23].

En ce qui concerne l'intégration de SNNs profonds avec des caméras événementielles, qui fournissent un type de données d'entrée adapté aux traitements par les neurones impulsionnels (les flux d'événements), l'utilisation de modèles SNN formés à l'aide du substitut du gradient suscite un grand intérêt pour combiner les avantages des deux technologies simultanément [Gal+20]. En conséquence, il existe un large éventail d'applications des SNNs en vision événementielle. La Table 2.2 passe en revue plusieurs modèles SNNs entraînés par le substitut du gradient pour la vision événementielle.

Modèle	Tâche(s) traitée(s)	Base(s) de Données
tdBN [Zhe+21] SEW-ResNet [Fan+21a] PLIF [Fan+21b] TA-SNN [Yao+21] Spikeformer [Zho+22] TCJA-SNN [Zhu+22b] NDA [Li+22b]	Reconnaissance d'objets, Reconnaissance d'actions	CIFAR10-DVS [Li+17], DVSGesture [Ami+17] + N-Caltech101 [Orc+15] + N-Cars [Sir+18]
[Niu+23] [Bul+23]	Reconnaissance labiale	DVS-Lip [Tan+22a]
[CMT22]	Détection d'objets	Gen1 [De +20]
[KCP21]	Segmentation sémantique	DDD17 [Bin+17]
[Zha+22b]	Tracking d'objet	FE240hz [Zha+21], EED [Mit+18], VisEvent [Wan+21]
[Cua+23]	Estimation de flux optique	DSEC [Geh+21b]
[Zou+23a]	Estimation/Tracking de pose humaine 3D	SynEventHPD [Zou+23a], DHP19 [Cal+19]
Stereospike [Ran+21] MSS-DepthNet [Wu+22]	Estimation de profondeur	MVSEC [Zhu+18b]
[Zhu+22a]	Reconstruction événements vers vidéo	MVSEC [Zhu+18b], IJRR [Mue+17], HQF [Sto+20]

Table 2.2: Revue des architectures SNN profondes entraînées par la BPTT et le substitut du gradient pour traiter des tâches de vision événementielle.

2.2.6 Exécution des Réseaux de Neurones Impulsionnels

Les neurones impulsionnels, en raison de leur mode de communication basé sur des impulsions asynchrones, ne sont pas naturellement adaptés aux architectures matérielles conventionnelles de type Von Neumann, telles que les processeurs (CPUs) et les processeurs graphiques (GPUs) habituellement utilisés pour les ANNs. Au lieu de cela, ces neurones s'épanouissent dans des environnements matériels spécifiques appelés "matériels neuromorphiques" [Bas+22]. Ces puces neuromorphiques sont particulièrement intéressantes en raison de leur vitesse d'exécution élevée et de leur faible consommation d'énergie. Cependant, en raison du stade de développement encore préliminaire de nombreuses puces neuromorphiques existantes, il est courant de simuler des SNNs sur du matériel conventionnel, en particulier pour l'entraînement de SNNs profonds utilisant des techniques d'apprentissage telles que le substitut du gradient [Cor22].

Dans cette section, nous abordons les méthodes courantes pour exécuter des SNNs et détaillerons les spécificités de chacune. Tout d'abord, nous examinerons les outils de simulation conçus pour les architectures conventionnelles de type Von Neumann (CPU et GPU), en évoquant leurs caractéristiques distinctes (règles d'apprentissage

disponibles, matériel cible, etc.). Ensuite, nous passons en revue les architectures neuromorphiques disponibles dans l'état de l'art afin de mieux comprendre les tendances du domaine.

2.2.6.1 Simulation sur Matériel Conventionnel

Les outils de simulation de neurones impulsionnels visent à faire fonctionner des SNNs sur du matériel conventionnel, mais ils diffèrent grandement selon leur but. En effet, les SNNs se situent à la croisée de beaucoup de domaines différents, notamment les neurosciences ou l'apprentissage automatique. Par conséquent, les fonctionnalités et caractéristiques d'un simulateur de SNNs varient avant tout en fonction de son but originel.

Simulateurs Cérébraux. Le terme "simulateur cérébral" ("*brain simulator*", en anglais) désigne les simulateurs de SNNs principalement utilisés pour les neurosciences. Ces simulateurs se caractérisent par leurs performances computationnelles robustes, leur utilisation simple pour définir les modèles de neurones et les synapses, ainsi que leur intégration transparente avec les plates-formes de calcul haute performance ("*high performance computing*" ou HPC, en anglais) parallèles. Les simulateurs les plus largement utilisés dans cette catégorie comprennent NEURON [CH06], NEST [GD07] et Brian [SBG19]. Ils fournissent une vaste gamme de modèles de neurones et de synapses, complétée par des outils efficaces pour configurer et interconnecter des réseaux à grande échelle. Au sein de ces réseaux, il est possible que différents modèles de neurones et de synapses coexistent, avec la flexibilité d'établir de multiples connexions, chacune ayant des propriétés distinctes, entre n'importe quelle paire de neurones. En général, ces simulateurs mettent en œuvre des mécanismes d'apprentissage grâce à des variations de la STDP et permettent souvent aux utilisateurs de mettre en place de nouvelles règles d'apprentissage. D'un point de vue technique, ces simulateurs sont principalement implémentés en langage C ou C++, profitant de la vitesse et de l'efficacité computationnelle offertes par ces langages. Cependant, Python est fréquemment utilisé comme langage fournissant une interface de programmation plus simple pour définir les réseaux de neurones. Il convient de noter que cette utilisation implique un interpréteur pour traiter l'architecture computationnelle, ce qui signifie que la structure du réseau reste statique pendant la simulation. Pour pallier cette limitation, [Dav+09] a introduit PyNN, une interface Python haut niveau indépendante du simulateur conçue pour construire des SNNs qui peuvent fonctionner sans modification sur n'importe quel simulateur pris en charge (y compris NEURON [CH06], NEST [GD07] et Brian [SBG19]). Cependant, l'utilisation de ces simulateurs cérébraux pour des

tâches complexes d'apprentissage automatique pose des défis. Des tâches telles que l'organisation des neurones en couches et la mise en œuvre d'opérations telles que les convolutions nécessitent des efforts supplémentaires d'implémentation. De plus, les simulateurs cérébraux manquent généralement d'une règle d'apprentissage supervisé car ils n'intègrent pas de moteur de différenciation automatique. Par conséquent, une catégorie distincte de simulateurs de SNNs, spécialisés dans les applications d'apprentissage automatique, a émergé.

Simulateurs pour l'Apprentissage Automatique. L'adoption généralisée des librairies pour la conception d'ANNs, tels que PyTorch [Pas+19] et TensorFlow [Mar+15], a considérablement simplifié la création des ANNs profonds avec une grande efficacité sur les CPUs et les GPUs. Ces librairies offrent une interface Python de haut niveau qui simplifie le développement de nouvelles couches et règles d'apprentissage novatrices, rapides et efficaces. Ils fournissent également un moyen simple d'incorporer de nouvelles opérations et de nouvelles couches, ce qui a conduit à l'émergence de plusieurs simulateurs spécialement conçus pour les SNNs dédiés à régler des problématiques d'apprentissage automatique et de vision artificielle. BindsNET [Haz+18] a marqué un effort pionnier en tant que premier simulateur de SNNs orientés pour l'apprentissage profond. BindsNET est basé sur PyTorch [Pas+19] pour construire des SNNs, intégrant de manière transparente des opérations typiques telles que les convolutions et le pooling, qui sont optimisées efficacement sur des dispositifs GPUs. L'apprentissage dans BindsNET repose principalement sur la STDP et ses variations. L'utilisation de l'accélération GPU dans BindsNet se traduit par des temps de simulation considérablement réduits pour les grands réseaux contenant des milliers de neurones, surpassant ainsi les simulateurs cérébraux populaires [CH06; SBG19]. Cela met en évidence le rôle crucial de l'exploitation des librairies d'apprentissage profond existantes (principalement PyTorch et Tensorflow) dans la construction de SNNs de grande échelle. SlayerTorch [SO18] est une autre implémentation basée sur PyTorch [Pas+19] qui implémente la règle d'apprentissage supervisée "*Spike LAYer Error Reassignment*" (SLAYER) pour l'entraînement de SNNs. De plus, SLAYER est intégré à l'outil LAVA [Int22], un système complet pour le calcul neuromorphique, axé en particulier sur la compatibilité avec la puce neuromorphique "Intel Loihi" [Dav+18]. D'autres librairies notables incluent Norse [PP21], basée sur PyTorch, qui offre l'entraînement de SNNs avec la règle d'apprentissage SuperSpike [ZG18], et Nengo-DL [Ras19], qui est basé sur TensorFlow/Keras [Mar+15]. Nengo-DL étend les capacités du simulateur Nengo [Bek+14] en introduisant des opérations d'apprentissage automatique et la conversion

ANN-vers-SNN de [HE15]. Nengo permet, à travers son interface, une compatibilité avec de plateformes matérielles (puces neuromorphiques, GPU, FPGA, ...). Enfin, SpikingJelly [Fan+20] est un autre outil d'entraînements pour SNNs basé sur PyTorch, et est grandement reconnu pour utiliser la règle d'apprentissage par substitut du gradient avec une importante efficacité sur GPU et une grande facilité d'utilisation. Grâce à ses qualités pour l'apprentissage profond, SpikingJelly est devenu le choix le plus populaire pour la conception de SNNs pour des problèmes d'apprentissage, notamment en vision artificielle. Par exemple, la majorité des travaux mentionnés en Table 2.2 sont implémentés avec SpikingJelly. Néanmoins, d'autres librairies adaptées pour l'apprentissage par substitut du gradient et basées sur PyTorch, telles que SnnTorch [Esh+21] et Sinabs [She+22], peuvent être mentionnées pour leurs fonctionnalités additionnelles (déploiement sur puce neuromorphique spécifique, interface de programmation alternative, ...).

Choix du Simulateur. En fonction de nos objectifs, nous pouvons sélectionner l'outil de simulation de SNNs le plus approprié pour notre cas d'utilisation. Étant donné que nos travaux sont axés sur le développement de CSNN profonds pour résoudre des problèmes de vision artificielle grâce à l'apprentissage par le substitut du gradient, nous recherchons un simulateur qui répond à plusieurs critères essentiels : **(1)** il doit être orienté vers l'apprentissage automatique ; **(2)** il doit avoir la capacité de fonctionner sur des GPUs de haute performance pour simuler efficacement des millions de neurones organisés en couches de convolution ; et **(3)** il doit intégrer la BPTT avec le substitut du gradient. À la lumière de ces exigences, nous avons opté pour l'utilisation de SpikingJelly [Fan+20] dans nos travaux. Cette librairie répond à tous nos besoins, offrant une interface de programmation polyvalente et complète pour notre approche en matière de SNNs en vision artificielle, tout en satisfaisant les critères mentionnés.

2.2.6.2 Matériel Neuromorphique

Le terme "matériel neuromorphique" désigne un nouveau type d'architecture matérielle inspirée des neurosciences et consistuée d'accélérateurs massivement parallèles pour les SNNs. Ces processeurs neuromorphiques sont composés d'unités de calcul physiques interconnectées capables de traiter des informations représentées par des impulsions, offrant ainsi un traitement plus efficace et localisé des données, ce qui se traduit par une consommation d'énergie réduite et une latence moindre. Étant un domaine émergent, se procurer un processeur neuromorphique reste difficile car ceux-ci sont soit indisponible à grande échelle (par exemple, ils requièrent un accord avec le

Processeur neuromorphique	Catégorie	# neurones	# synapses	Consommation énergétique
Braindrop [Nec+18]	Mixte analogique/numérique	4k	-	150μW
DYNAP-SEL [Mor+17]		1k	80k	200μW
SpiNNaker [Fur+14]	Numérique	1k	1M	1W
TrueNorth [Ako+15]		1M	256M	70mW
Loihi [Dav+18]		130k	130M	100mW
Loihi 2 [Orc+21]		1M	120M	100mW
Akida		1.2M	100B	20mW

Table 2.3: Liste comparative des processeurs neuromorphiques évoqués en Section 2.2.6.2. Source : [Cor22].

constructeur [Dav+18]), soit inadapés pour un cas d'utilisation précis. Néanmoins, des produits neuromorphiques commerciaux ont récemment fait leur apparition [Bra20].

Les architectures neuromorphiques sont généralement catégorisées en "implémentations numériques" ("*digital implementations*", en anglais) ou "implémentations mixtes analogiques/numériques" ("*mixed analog/digital implementations*", en anglais). Les systèmes analogiques exploitent les caractéristiques physiques des dispositifs électroniques dans leurs calculs, tandis que les systèmes numériques s'appuient sur des portes logiques booléennes. En conséquence, les systèmes analogiques fonctionnent de manière asynchrone sur des valeurs continues, tandis que les systèmes numériques travaillent de manière synchrone avec des valeurs discrètes régies par un signal d'horloge.

En comparant les deux catégories mentionnées, il s'avère que les implémentations numériques sont généralement plus matures que les implémentations mixtes analogiques/numériques, dans le sens où ces premières permettent de faire fonctionner un plus grand nombre de neurones connectés avec un plus grand nombre de synapses (et donc de faire fonctionner un SNN plus complexe). Néanmoins, ces avantages viennent de pair avec une efficacité énergétique moins intéressante que les systèmes mixtes analogiques/numériques.

D'un point de vue conceptuel, les architectures analogiques sont plus attrayantes car naturellement asynchrones et bien plus efficaces en termes de consommation énergétique [Bas+22]. Cependant, ils sont aussi moins matures que les implémentations numériques, plus conventionnelles, et sont plus sensibles aux bruits. Des exemples de matériels neuromorphiques mixtes analogiques/numériques comprennent DYNAP-SEL [Mor+17] et Braindrop [Nec+18]. Braindrop est une puce de recherche, avec 4 000 neurones utilisant des calculs et des communications analogiques, ce qui se traduit par une consommation d'énergie aussi faible que 150 μW. DYNAP-SEL comprend 1 000 neurones analogiques et jusqu'à 80 000 connexions synaptiques configurables,

mais sa consommation d'énergie dépend de la fréquence des impulsions émises par les neurones, allant de 200 μW (pour une fréquence d'impulsion de 0 Hz) à 1 mW (pour une fréquence de 100 Hz). Bien que ces systèmes neuromorphiques analogiques aient un nombre limité de neurones, ils restent un domaine de recherche prometteur pour l'avenir. Dans l'état actuel, les systèmes analogiques présentent des limitations pour faire fonctionner des SNNs profonds de grande taille.

SpiNNaker [Fur+14] a été l'un des premiers systèmes neuromorphiques, utilisant une architecture numérique entièrement personnalisable et massivement parallèle composée de nombreuses petites unités (cœurs ARM). SpiNNaker facilite l'intercommunication entre les cœurs, traitant de nombreux petits messages (les impulsions). De plus, ses unités de traitement peuvent être conçues par un logiciel, ce qui permet un déploiement polyvalent de modèles de neurones au détriment de l'accélération matérielle.

En 2015, IBM a présenté TrueNorth [Ako+15], une architecture ASIC neuromorphique pionnière. Chaque puce TrueNorth comprend 1 million de neurones numériques et 256 millions de synapses réparties sur 4 096 cœurs, ce qui se traduit par une consommation d'énergie réduite de seulement 70 mW. Notamment, la TrueNorth a suscité un certain intérêt pour sa capacité à mettre en oeuvre un CSNN pour la reconnaissance d'actions [Ami+17] à l'aide d'une caméra événementielle.

En 2018, Intel a dévoilé Loihi [Dav+18], une puce neuromorphique de recherche dotée de 128 cœurs capables de modéliser 130 000 neurones numériques et 130 millions de synapses. Fabriquée selon un processus 14 nm, la puce ne consomme que 110 mW dans un cas d'utilisation réel de détection de mots-clés, comparativement aux 650 mW consommés par l'accélérateur ANN Intel Movidius [Blo+19]. Les puces Loihi peuvent être interconnectées pour augmenter le nombre de neurones et de synapses, jusqu'à 100 millions de neurones et 100 milliards de synapses avec 768 puces. Une deuxième version, Loihi 2 [Orc+21], a été lancée en 2021, offrant 1 million de neurones par puce sur une surface plus petite (31 mm^2 par rapport à 60 mm^2), une performance dix fois plus rapide, la possibilité de programmer des neurones et des impulsions à nombres entiers, tout en conservant la même consommation d'énergie. De plus, la Loihi suscite un grand intérêt dans la communauté neuromorphique, notamment grâce à la simplicité de son interface de programmation via la librairie Lava [Int22] (mentionnée en Section 2.2.6) et grâce à ses capacités d'apprentissage sur la puce elle-même. C'est pourquoi de nombreux travaux ont pu démontrer l'intérêt

des approches neuromorphiques par rapport aux méthodes conventionnelles en étant déployé sur la Loihi [Dav+21].

En 2019, Brainchip a introduit le premier processeur neuromorphique breveté, nommé Akida [Bra20]. Il a été présenté comme un accélérateur matériel pour diverses tâches d'intelligence artificielle. Brainchip propose une chaîne d'outils complète pour la conversion ANN-vers-SNN avec des impulsions entières allant jusqu'à 4 bits pour minimiser la perte de précision lors de la conversion.

Le Tableau 2.3 détaille les comparaisons entre les processeurs neuromorphiques mentionnés dans cette section. Bien que l'intérêt de déployer nos modèles CSNNs est certain pour nos travaux, il est encore difficile de se procurer de tels processeurs et ceux-ci sont encore limités vis-à-vis de la complexité des réseaux de neurones. En conséquence, nos travaux n'incluent pas de déploiement sur processeur neuromorphique, mais l'emploi de la librairie SpikingJelly [Fan+20] promet une intégration sur processeur neuromorphique (via une conversion sur Lava [Int22] - Loihi [Dav+18]), ce qui encouragerait des évaluations a posteriori.

2.2.7 Verrous Scientifiques

À travers cette Section 2.2, nous avons présenté un aperçu des travaux liés aux SNNs, en mettant en évidence les avancées dans le contexte des architectures profondes pour le traitement de la vision artificielle. Au cours de notre revue de l'état de l'art, nous avons établi les fondements conceptuels de notre étude :

- Nous avons introduit les modèles de neurones impulsionnels IF et LIF, choisis pour leur efficacité computationnelle et leur pertinence dans le développement de SNNs profonds.
- Nous avons examiné diverses techniques courantes de codages neuronaux des images statiques, essentielles pour les SNNs travaillant avec des caméras conventionnelles en vision par ordinateur.
- Nous avons formulé l'apprentissage supervisé par substitut du gradient [NMZ19] à partir du modèle de neurone IF, une méthode clé pour développer des CSNNs profonds résolvant des problèmes complexes de vision par ordinateur.
- Nous avons passé en revue les outils disponibles pour la simulation des SNNs, qu'il s'agisse de simulations sur du matériel conventionnel (CPU/GPU) ou de

l'utilisation de processeurs neuromorphiques. Nous avons également justifié le choix de SpikingJelly [Fan+20] comme outil de simulation adapté à nos besoins, tout en notant les défis actuels liés à l'accès aux processeurs neuromorphiques pour le déploiement de nos travaux.

Cependant, des limitations se dégagent de notre revue de l'état de l'art, à la fois dans nos choix de conception et dans le contexte actuel de la vision artificielle.

Tout d'abord, l'apprentissage par la BPTT et le substitut du gradient est reconnu comme étant *non local*, ce qui signifie que les paramètres du SNN sont ajustés en utilisant des informations ne provenant pas uniquement des neurones voisins. Cette caractéristique pose un défi pour le déploiement de cette technique sur du matériel neuromorphique, car de tels systèmes reposent fortement sur le principe de la localité des informations échangées. Bien que des solutions soient en cours de développement pour atténuer ce problème [Nøk16; KMN20], elles ne parviennent pas encore à égaler les performances de l'application pure de la BPTT. Ainsi, il convient de noter que les SNNs profonds entraînés par la BPTT pour la vision, y compris nos travaux, restent limités à un entraînement sur du matériel conventionnel en attendant que des solutions plus robustes soient développées.

Plus spécifiquement dans le domaine de la vision artificielle, l'utilisation du substitut du gradient a permis des avancées significatives dans le développement de SNNs profonds pour résoudre des tâches visuelles complexes (voir Section 2.2.5.4). Cependant, cette rapide progression pour résoudre une variété croissante de problèmes masque une problématique fondamentale : les comportements de ces nouveaux SNNs profonds peuvent différer de ceux des méthodes traditionnelles, telles que les ANNs ou les SNNs entraînés avec des règles établies comme la STDP. Nous soutenons qu'il est essentiel de conduire des analyses approfondies sur les SNNs profonds entraînés par la BPTT (comme celles présentées dans [CLR21; Bou+22]), car il n'y a aucune garantie que leur comportement sera similaire à celui des approches traditionnelles. Par conséquent, dans nos travaux, notre attention se porte principalement sur l'analyse des CSNNs entraînés par le substitut du gradient, afin de mieux comprendre comment différentes configurations influencent leurs mécanismes (telles que le codage neuronal, les techniques d'augmentation de données, les corruptions des capteurs, etc.).

2.3 Vision Événementielle

La **vision événementielle**, également connue sous le nom de "*event-based vision*" en anglais [Gal+20], constitue un domaine spécifique de la vision artificielle. Il englobe les algorithmes de traitement visuel conçus pour interpréter les données produites par un type particulier de capteur visuel inspiré de la biologie, appelé la "**caméra événementielle**" ("*event camera*", en anglais). Les données générées par ces capteurs, désignées sous le terme d'"**événements**", diffèrent fondamentalement des images statiques capturées par les caméras conventionnelles.

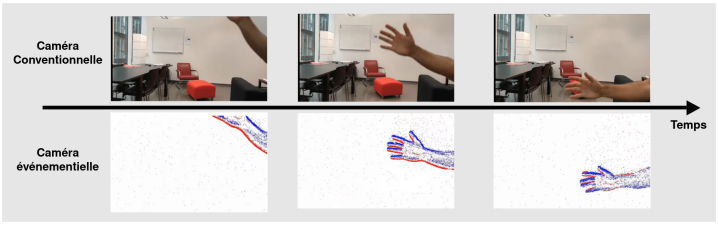
Dans cette section, nous offrons une vue d'ensemble du domaine de la vision événementielle. Cela comprend une exploration des méthodes spécifiques pour le traitement de ces flux d'événements, une présentation des tâches de vision existantes relevant de ce domaine, ainsi qu'une discussion sur les modèles d'apprentissage adaptés à ces tâches.

2.3.1 Caméra Événementielle

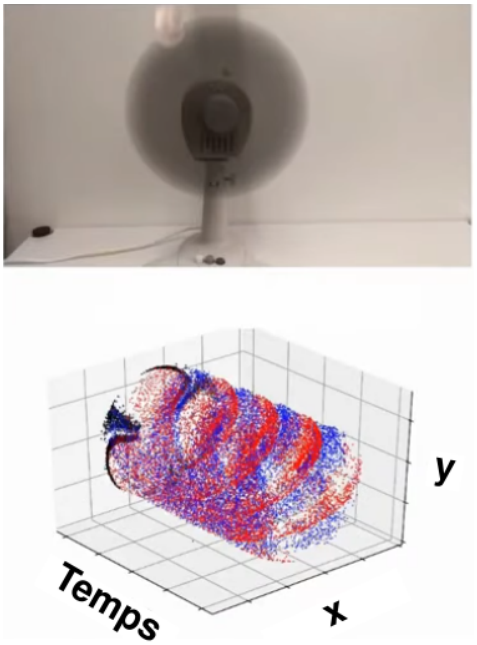
En parallèle de l'étude de systèmes de calculs tels que les SNNs, l'ingéniérie neuromorphique s'étend aussi dans le domaine de la perception visuelle, en inaugurant un nouveau type de capteur, la caméra événementielle, qui imite de près la vision biologique [Gal+20]. L'élément distinctif réside dans leur mode de fonctionnement unique, où chaque pixel de la caméra fonctionne indépendamment, déclenchant un signal, un "*événement*", uniquement en réponse aux variations de luminosité, généralement provoquées par le mouvement ou d'autres altérations visuelles [Dav+21].

En ce qui concerne la structure de données des événements, ceux-ci peuvent être considérés comme un flux asynchrone où chaque événement symbolise un changement discret, généralement positif ou négatif. De ce fait, les flux d'événements partagent de nombreuses similitudes avec les trains d'impulsions échangés par les neurones impulsionnels. Cette correspondance permet aux SNNs d'exploiter directement les événements produits par une caméra événementielle, créant ainsi une synergie prometteuse entre ces deux systèmes neuromorphiques pour résoudre efficacement les problèmes de vision (voir Section 2.2.5.4 pour plus de détails).

Dans la pratique, les flux d'événements capturés par les caméras événementielles se caractérisent par leur parcimonie, avec une concentration particulière le long des contours des objets. La Figure 2.17 montre des exemples de la sortie de ces flux d'événements.



(a) Flux d'événements visualisé sous forme de séquence d'images événementielles (voir Section 2.3.3.1). La scène capturée représente un bras bougeant de haut en bas.



(b) Flux d'événements visualisé dans le domaine spatio-temporel. La scène capturée représente un ventilateur en fonctionnement.

Figure 2.17: Exemples de flux d'événements capturés par une caméra événementielle, selon différentes représentations visuelles. On remarque que les événements générés par la caméra se situent avant tout au niveau des contours des objets en mouvement. Source : Davide Scaramuzza (https://www.youtube.com/watch?v=6Sn9-M7qXLk)

2.3.1.1 Avantages par Rapport aux Caméras Conventionnelles

En tant que nouveau type de capteur émergent, les caméras événementielles comportent plusieurs avantages par rapport aux caméras conventionnelles [Gal+20] :

Haute résolution temporelle. L'observation des variations de luminosité de la caméra événementielle se produit rapidement, grâce à des circuits analogiques, tandis que la lecture des événements s'effectue numériquement à l'aide d'une horloge de 1 MHz. En d'autres termes, les événements sont rapidement détectés et horodatés avec une précision de l'ordre de la microseconde [Reb+19b]. Par conséquent, les caméras événementielles excellent dans la capture de mouvements extrêmement rapides, évitant le flou cinétique qui affecte les caméras conventionnelles [Jia+20b].

Latence réduite. Chaque pixel fonctionne de manière autonome, éliminant le besoin d'attendre un temps d'exposition global de l'image entière ; dès qu'un changement

est détecté, il est rapidement transmis. Par conséquent, les caméras événementielles présentent une latence exceptionnellement faible, d'environ 10 μs en conditions de laboratoire et de moins d'une milliseconde dans des scénarios du monde réel.

Faible consommation d'énergie. Les caméras événementielles présentent une faible consommation d'énergie grâce à leur transmission sélective des changements de luminosité, éliminant ainsi le traitement redondant des données qu'on observe dans les caméras conventionnelles. L'énergie est principalement utilisée pour gérer les pixels en mutation. Au niveau de la puce de silicium, la plupart des caméras événementielles consomment environ 10 mW, avec des prototypes atteignant une consommation d'énergie encore plus faible, souvent inférieure à 10 μW.

Haute plage dynamique. Les caméras événementielles possèdent une plage dynamique dépassant les 120 dB, surpassant largement la plage de 60 dB des caméras conventionnelles. Cette plage étendue permet aux caméras événementielles de capturer des informations dans des conditions d'éclairage plus extrêmes (par exemple, pendant la nuit) [ZYS21]. Cette capacité résulte du fonctionnement propre et indépendant des photorécepteurs des pixels. De manière similaire à la rétine biologique, cela permet aux pixels de s'adapter à des luminosités très faibles ou très fortes.

De nombreuses applications peuvent bénéficier d'un ou de plusieurs de ces avantages, que ce soit les systèmes embarqués (pour la faible consommation énergétique) [Ami+17] ou même l'amélioration de la qualité photographique en employant une caméra conventionnelle et événementielle simultanément [Han+20].

2.3.1.2 Changement de Paradigme

Comme évoqué en Section 2.3.1, les caméras conventionnelles et événementielles ont un fonctionnement fondamentalement différent, si bien que les algorithmes de vision artificielle "classique", adapté pour traiter des images statiques, ne sont pas compatibles avec le traitement des événements. Pour profiter des avantages offerts par les caméras événementielles (détaillés en Section 2.3.1.1), il faut concevoir de nouvelles méthodes capables de fonctionner avec les flux d'événements, ce qui constitue le coeur du domaine de la vision événementielle. Par rapport aux images statiques, les algorithmes taillés pour la vision événementielle doivent tenir compte de certains enjeux :

S'adapter à des sorties asynchrones. Les caméras événementielles produisent des sorties fondamentalement différentes par rapport aux caméras conventionnelles. Les événements sont asynchrones et spatialement épars, tandis que les images statiques sont des tenseurs synchrones et denses. Par conséquent, les algorithmes de vision basés sur des images statiques ne sont pas directement applicables aux données événementielles.

Gérer les différences de détection photométrique. Les caméras événementielles se distinguent des caméras conventionnelles en ce que chaque événement transmet une information binaire sur le changement de luminosité (positif ou négatif). Ces changements dépendent à la fois de la luminosité de la scène et de l'historique des mouvements relatifs entre la scène et la caméra.

Gérer des bruits de capteurs. Tout capteur visuel est sensible à des effets de bruits à cause de plusieurs facteurs tels que le bruit dans les circuits à transistors. Cette constatation est aussi vraie pour les caméras événementielles dont les corruptions de capteur ont des effets particulier sur les flux d'événements bruités [Fen+20; HLD21].

2.3.1.3 Caméras Événementielles Existantes

Depuis le premier prototype de capteur événementiel reproduisant le fonctionnement de l'oeil, nommé la "rétine en silicium" [MM94] (en 1991), des progrès significatifs ont été réalisés [DT15; Pos+14], si bien que les caméras événementielles sont maintenant assez matures pour être commercialisées par des fabricants tels que iniVation, Prophesee, Samsung et CelePixel.

Bref Historique. iniVation a été le premier fabricant à commercialiser une caméra événementielle, nommée la DVS128 [LPD08], en 2008, offrant une résolution de (128×128) pixels. En 2011, Prophesee a dévoilé sa première caméra événementielle, nommée "ATIS" [PMW10], dotée d'une résolution de (304×240) pixels. Au fil du temps, la résolution des caméras événementielles n'a cessé d'augmenter, avec des modèles tels que le Prophesee GEN4 CD [Fin+20b], le Samsung DVS-Gen4 [Suh+20a] et le CelePixel CeleX-V [CG19a] offrant désormais des résolutions dépassant (1280×720) pixels. La résolution est limitée par la taille physique des pixels, et il reste incertain si les fabricants pourront encore améliorer la résolution des caméras événementielles, qui reste bien en deçà de celle des caméras conventionnelles. D'autre part, des avancées récentes ont conduit au développement de caméras événementielles capables de

Fabricant	iniVation			Prophesee				Samsung			CelePixel	
Caméra	DVS128	DAVIS240	DAVIS346	ATIS	Gen3 CD	GEN3 ATIS	GEN4 CD	DVS-Gen2	DVS-Gen3	DVS-Gen4	CeleX-IV	CeleX-V
Année	2008 [LPD08]	2014 [Bra+14]	2017 -	2011 [PMW10]	2017 [Pro20]	2017 [Pro20]	2020 [Fin+20a]	2017 [Son+17]	2018 [Ryu19]	2020 [Suh+20b]	2017 [GHC17]	2019 [CG19b]
Résolution ($H \times W$)	(128 × 128)	(180 × 240)	(260 × 346)	(240 × 304)	(480 × 640)	(360 × 480)	(720 × 1280)	(480 × 640)	(480 × 640)	(960 × 1280)	(640 × 768)	(800 × 1280)
Latence (μs)	12	12	20	3	40 - 200	40 - 200	20 - 150	65 - 410	50	150	10	8
Plage dynamique (dB)	120	120	120	143	> 120	> 120	> 124	90	90	100	90	120
Consommation d'énergie (mW)	23	5 - 14	10 - 170	50 - 175	36 - 95	25 - 87	32 - 84	27 - 50	40	130	-	400
Image en niveaux de gris	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	✓
Sortie IMU	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗

Table 2.4: Comparaison de certaines caméras événementielles commercialisées. Tableau adapté de [Gal+20].

capturer la couleur comme l'iniVation Color-DAVIS346 [Tav+18]. Cependant, il n'a pas été démontré de manière concluante que l'information de couleur pour les événements ait un intérêt conséquent comme c'est le cas pour les images statiques [Li+15; Mar+18].

Améliorations des Fonctionnalités. Outre l'augmentation de la résolution des caméras, les constructeurs cherchent à améliorer leurs produits en y ajoutant des fonctionnalités pertinentes pour certains cas d'utilisation. Ainsi, des caméras commercialisées telles que la DAVIS [Bra+14] proposent une sortie parallèle sous la forme d'une image en niveaux de gris, l'intégration d'une unité de mesure inertielle ("*Inertial Measurement Unit*" ou IMU, en anglais) [DVL14], ou encore la synchronisation multi-caméras [Ber06]. Dans un registre plus exploratoire, certains travaux cherchent à améliorer/corriger certaines caractéristiques des caméras événementielles en proposant des circuits de pixels modifiés. Par exemple, la caméra événementielle "CS-DVS" [Del+22], uniquement simulée par logiciel à l'heure actuelle [HLD21], propose une architecture de pixel rendant la caméra plus robuste aux changements bruités de basses fréquences. On peut également mentionner la caméra PDAVIS [Hae+23], une caméra événementielle à polarisation inspirée du fonctionnement du système visuel de la crevette mante, considéré comme l'un des plus sophistiqué du règne animal [Mar88].

Le Tableau 2.4 présente une vue d'ensemble comparative de plusieurs caméras événementielles disponibles sur le marché.

2.3.2 Formulation de la Génération des Événements

Chaque pixel d'une caméra événementielle réagit indépendamment aux changements de photocourant logarithmique $\log V$ ("luminosité") qu'il observe. Autrement dit, un pixel de coordonnées (x, y) d'une caméra événementielle de résolution $(H \times W)$ émet un événement e à un instant t dés que la différence de luminosité depuis l'événement précédent atteint un certain seuil $\pm\gamma$. L'événement e généré encode une information de "polarité" $p \in \{1, -1\}$ qui représente le signe du changement de luminosité du pixel, où

$p = 1$ représente un pixel passant du sombre (*OFF*) au clair (*ON*) et $p = -1$ représente l'inverse. Plus formellement, la polarité p est donnée par l'équation suivante:

$$p = \begin{cases} 1 & \text{if } \log V(t,x,y) - \log V(t-\Delta t,x,y) > \gamma \\ -1 & \text{if } \log V(t,x,y) - \log V(t-\Delta t,x,y) < -\gamma \end{cases} \tag{2.14}$$

, où $\log V(t,x,y)$ est la luminosité observée au pixel (x,y) à l'instant t, et Δt est la durée qui sépare l'événement e de celui qui le précède.

Par conséquent, un événement e peut être représenté par un tuple de 4 valeurs:

$$e = (x,y,p,t) \tag{2.15}$$

qui constitue la base du protocole AER (*"Adress-Event Representation"*, en anglais) [Boa04], la représentation commune de la sortie des caméras événementielles.

Pendant un intervalle de temps $\Delta\mathcal{T}$, une caméra événementielle génère un ensemble $\mathcal{E}$ de N événements asynchrones, tel que :

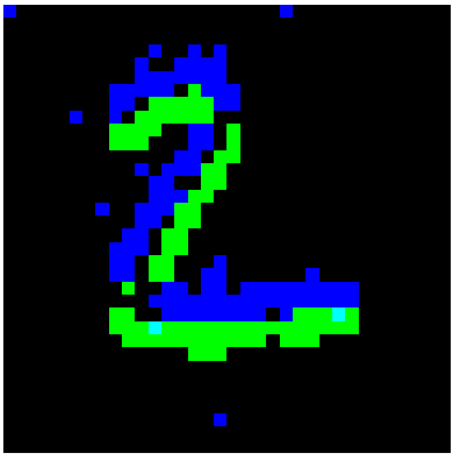
$$\mathcal{E} = \{e_i\}_{i=1}^{N} = \{(x_i,y_i,p_i,t_i)\}_{i=1}^{N} \tag{2.16}$$

.

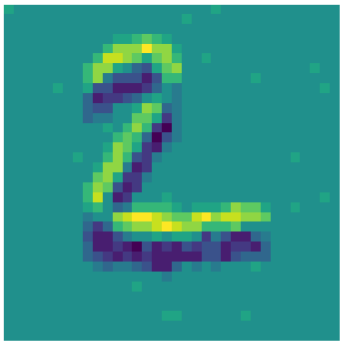
2.3.3 Représentation des Événements

Pour traiter un flux d'événements asynchrones $\mathcal{E}$ (formulé en Section 2.3.2), la grande majorité des méthodes existantes applique un pré-traitement nommé "**représentation d'événements**" ("*event representation*", en anglais). L'objectif d'une représentation d'événements est de transformer un flux d'événements asynchrones en une représentation qui exprime plus efficacement les informations de la scène observée, ou qui facilite le traitement de ces données par un certain algorithme.

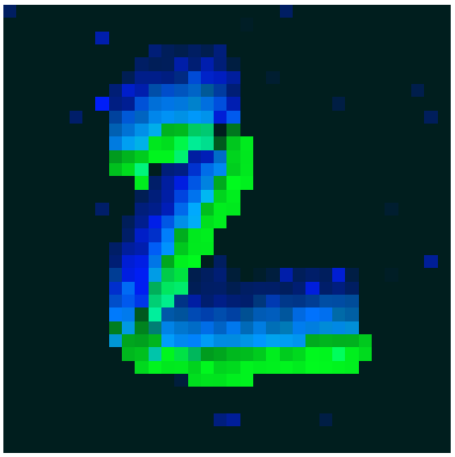
Par exemple, le modèle proposé dans [Kim+21] utilise un CNN pour classifier des flux d'événements selon l'objet observé. Comme un CNN ne peut que traiter des images (c'est-à-dire, des tenseurs synchrones de la forme $(C \times H \times W)$), l'approche comporte une étape de pré-traitement destinée à reconstruire des images à partir d'un ensemble d'événements $\mathcal{E}$.



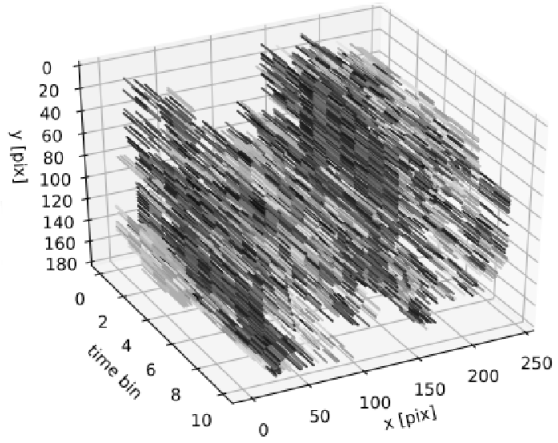
(a) Image événementielle binaire.



(b) Histogramme d'événements.



(c) Surface temporelle [Lag+16].



(d) Représentation en Voxel [Zhu+19].

Figure 2.18: Exemples de représentations d'événements.

Contrairement à la vision artificielle conventionnelle qui utilise la représentation d'images en RGB (c'est-à-dire des tenseurs à trois dimensions $3 \times H \times W$) comme représentation de référence, la représentation de flux d'événements est un domaine de recherche actif où il n'existe pas de méthode de référence. Plusieurs catégories de méthodes existent, avec leurs spécificités propres. Dans cette section, nous abordons les paradigmes de représentation d'événements populaires [Zhe+23].

2.3.3.1 Images Événementielles

Cette stratégie regroupe les méthodes qui visent à obtenir une ou plusieurs images à partir d'un flux d'événements $\mathcal{E}$. Cette stratégie est particulièrement populaire grâce à sa facilité d'utilisation, étant donné qu'elle permet d'employer des techniques de vision artificielle communes, tel que des CNNs, ViTs, etc. De plus, les modèles de l'état de l'art se basant sur les images événementielles ("*event frames*", en anglais) atteignent de très bonnes performances. Plus formellement, une image événementielle obtenue à partir d'un flux d'événements $\mathcal{E}$ est désignée par un tenseur à trois dimensions $(C \times H \times W)$, avec C étant le nombre de canaux de l'image construite (spécifique à la méthode employée).

Dans le reste de cette section, nous formalisons les techniques d'images événementielles les plus largement employées, et utilisées dans nos travaux.

Image Événementielle Binaire (Binary Event Frame)**.** À chaque pixel de coordonnées (x, y) et pour une polarité p, une image événementielle binaire encode la présence ou l'absence d'un événement dans $\mathcal{E}$ via la relation Img_{bin} tel que:

$$Img_{bin}(x, y, p) = \mathbb{1}(x, y, p) \tag{2.17}$$

, où $\mathbb{1}(x, y, p)$ est la fonction indicateur qui retourne 1 si un événement de polarité p est présent aux coordonnées de pixel (x, y) et 0 sinon. Par conséquent, l'image événementielle obtenue est de dimensions $(2 \times H \times W)$ où $C = 2$ correspond aux deux valeurs de polarité possibles (-1 et 1). La Figure 2.18a illustre un exemple d'image événementielle binaire. Bien que très populaires, les images événementielles binaires ont le défaut de "saturer" facilement si le nombre d'événements accumulés est trop grand. On parle de "saturation" d'une représentation d'événements lorsque le nombre d'événements représentés est si grand que les motifs exprimés deviennent inutiles pour l'extraction de caractéristiques.

Histogramme d'Événements (Event Histogram)**.** Plutôt que d'encoder uniquement la présence et l'absence d'événements, un histogramme d'événements représente le nombre d'événements capturés pour le même pixel via la fonction *hist*:

$$hist(x, y, p) = Count(x, y, p) \tag{2.18}$$

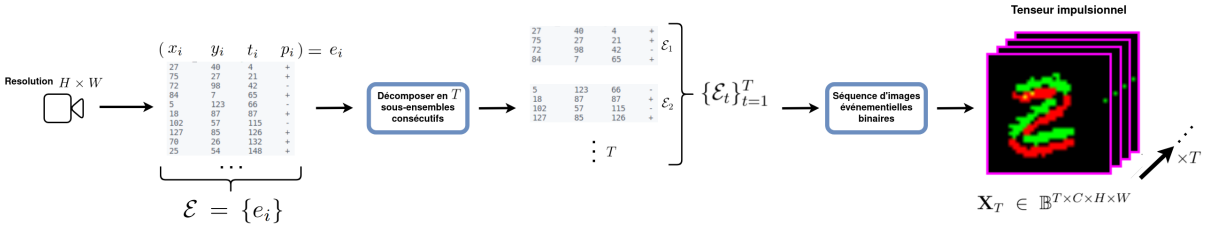


Figure 2.19: Procédé de création d'une séquence d'images événementielles binaires $\mathbf{X}_T$, aussi appelé "tenseur impulsionnel".

, où $Count(x, y, p)$ est la fonction de comptage d'événements qui retourne le nombre d'événements de polarité p et de coordonnées (x, y) dans $\mathcal{E}$. La Figure 2.18b illustre un exemple d'histogramme d'événements. Cette technique risque aussi de saturer en intégrant un trop grand nombre d'événements, car certains pixels contenant beaucoup d'événements peuvent montrer des valeurs démesurées par rapport à des parties de l'image moins actives. Cependant, les histogrammes d'événements sont plus robustes à ce phénomène que les images événementielles binaires.

Séquence d'Images Événementielles. Une faiblesse notable des images événementielles est la perte d'information temporelle des événements accumulés, vu que l'information d'horodatage (la valeur t d'un événement e) est perdue. De ce fait, seule l'information spatiale du flux d'événements est conservée. Cependant, il est possible de mitiger ce problème en extrayant plusieurs images événementielles consécutives à partir du même flux d'événements afin de traiter la séquence obtenue avec des algorithmes vidéo (par exemple, des 3D-CNNs) [Ber+23]. Dans ce cas, une séquence de T images événementielles est construite à partir des sous-ensembles d'événements $\mathcal{E}_t$:

$$\{\mathcal{E}_t\}_{t=1}^{T} \text{ , où } \mathcal{E}_t \subseteq \mathcal{E} \tag{2.19}$$

Avec ce procédé, le principal enjeux reste de bien définir les sous-ensembles d'événements qui seront utilisés, afin d'extraire un maximum d'informations spatio-temporelles de l'ensemble d'événements original. Par exemple, si les sous-ensembles sont trop proches les uns des autres dans le temps, il se peut que les images événementielles obtenues soient quasiment identiques et donc redondantes. Cette problématique reste un problème ouvert dans le domaine [Zhe+23], sans solution de référence établie. Pour la suite de ce manuscrit, on désigne par $\mathbf{X}_T \in \mathbb{B}^{T \times C \times H \times W}$ une séquence de T images événementielles binaires, aussi connue sous le nom de "**tenseur impulsionnel**" ou "**tenseur d'impulsions**". La Figure 2.19 illustre le procédé de création de ce tenseur d'impulsions.

2.3.3.2 Surface Temporelle

Le paradigme des "Surfaces Temporelles" ("*Time Surfaces*", en anglais) regroupe les méthodes visant à construire des surfaces (images, patches de voisinage de pixels, etc.) à partir de l'information temporelle des événements, contrairement aux images événementielles qui se concentrent sur l'apparition des événements sans conserver leurs horodatages (la valeur t dans un événement e). Initialement proposé par [Ben+13] avec la représentation SAE ("Surface of Active Events"), le procédé typique d'une méthode de surfaces temporelles est de faire correspondre un flux d'événements à une surface spatio-temporelle qui trace l'activité des voisinages de pixels proches des derniers événements apparus. De ce fait, les motifs exprimés par cette surface correspondent à des zones contenant du mouvement.

Toutefois, les représentations en surfaces temporelles doivent tenir compte d'un enjeu spécifique aux horodotages des événements : ceux-ci augmentent sans cesse au fur et à mesure que le temps passe [Lin+20], allant potentiellement à l'infini. Pour éviter que cette augmentation monotone des horodotages n'entraîne de phénomène de saturation, la solution de prédilection consiste à normaliser les surfaces temporelles, soit avec des méthodes de normalisation basiques [AC18; Lag+16; Afs+19] (par exemple, min-max [PS15]), soit avec des approches plus évoluées modifiant le comportement de la surface temporelle [Sir+18; Kim+21; Man+19].

Un exemple de surface temporelle, basé sur l'approche "HOTS" [Lag+16], est illustré en Figure 2.18c.

2.3.3.3 Représentations en Voxels

Les représentations en voxels consistent à convertir les événements bruts en plages temporelles au sein de grilles de voxels dans un espace 3D. Ainsi, la troisième dimension de cet espace exprime la dimension temporelle. La précision d'une telle représentation dépend avant tout de la taille allouée à cet espace 3D : plus l'espace défini est grand, plus les voxels sont placés avec précision vis-à-vis de l'ordre des événements qu'ils représentent.

Le concept d'une grille de voxels spatio-temporels a été introduit initialement dans [Zih+18], qui a utilisé une méthode d'accumulation pondérée linéaire pour améliorer la résolution temporelle. Des travaux ultérieurs, tels que [Zhu+19] et [Reb+19a], ont également adopté cette approche de grille de voxels spatio-temporels.

Plus récemment, [Bal+22] a introduit un volume d'événements ordonnés dans le temps ("*TORE*") conçu pour préserver efficacement les données temporelles brutes des impulsions avec une perte minimale d'information, tandis que [CMT22] propose la méthode des "Voxel Cubes", où chaque voxel regroupe plusieurs plages de temps de second ordre afin d'exprimer une meilleure résolution temporelle que la méthode en voxels [Zhu+19] originale.

Un exemple de représentation en voxels (plus précisément, "Voxel Grid" [Zhu+19]) est montré en Figure 2.18d.

2.3.3.4 Représentations en Graphes

Dans le but de préserver la parcimonie des événements, les approches basées sur les graphes [Den+22; Bi+20; Bi+19; Li+21a] restructurent les flux d'événements bruts se produisant dans une période de temps définie en un réseau de nœuds interconnectés. Ainsi, une telle représentation en graphe peut être traitée à l'aide d'un réseau de neurones adaptés, tel qu'un CNN à graphes, comme proposé par [Bi+20; Bi+19]. Ce type de représentation d'événements possède l'avantage de conserver la cohérence spatio-temporelle du flux d'événements.

Les approches plus récentes suivant ce paradigme de représentation, telles que [Den+22], proposent d'améliorer la construction et le traitement du graphe des événements. Ainsi, [Den+22] propose un modèle CNN à graphes basé sur des voxels [Zhu+19] pour mieux tirer profit de la parcimonie des événements. D'autre part, [Li+21a; SGS22] démontrent la possibilité de réaliser un processus asynchrone en utilisant des graphes d'événements, ce qui a pour effet d'améliorer grandement la rapidité d'exécution tout en créant un algorithme de vision capable de traiter les événements au fur et à mesure.

2.3.3.5 Représentations Entraînables

Les catégories de représentation d'événements citées précédemment consistent à mettre au point un pré-traitement visant à représenter au mieux les informations visuelles contenues dans les événements. Cependant, concevoir un traitement fixe pour exposer certaines des caractéristiques des événements n'est pas sans rappeler la philosophie derrière les descripteurs calculés pour les algorithmes traditionnels en vision artificielle (SIFT [Lin12], LBP [SJE14], etc) : les caractéristiques exposées sont définies par la méthode employée, sans certitude qu'elle soient adaptées au problème de vision

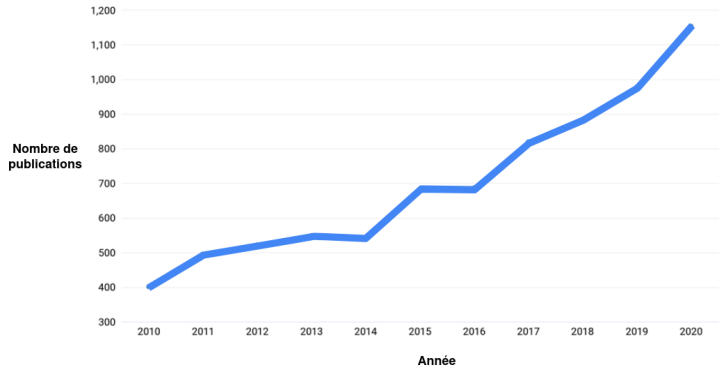


Figure 2.20: Nombre de publications par année dans le domaine de la vision événementielle. Source : Guillermo Gallego et al., CVPR 2023 Workshop on Event-based Vision.

visé. C'est pourquoi plusieurs travaux [Geh+19; Can+20; NPL16] se concentrent sur l'élaboration de méthodes de représentations d'événements pouvant être entraînées durant la phase d'apprentissage, de sorte que la représentation obtenue soit optimisée pour le problème de vision événementielle à traiter.

Parmi ces représentations d'événements dites "entraînables", la méthode pionnière [Geh+19], nommée "EST", vise à créer une représentation en tenseurs (donc utilisables par des algorithmes de vision traditionnels comme les CNNs) en utilisant des opérations différentiables, ce qui permet aux paramètres de cette méthode d'être optimisés de bout en bout par la rétropropagation. Les avancées dans ce paradigme consistent en l'intégration de traitements spatio-temporels supplémentaires dans la création de la représentation entraînable, tels que des RNNs [Can+20; NPL16].

2.3.3.6 Traitements Direct

On désigne par "traitement direct" l'ensemble des méthodes qui traitent les événements directement, sans passer par une étape de pré-traitement comme celles mentionnées précédemment. On parle ainsi d'algorithmes de vision capables de traiter le flux d'événements bruts et asynchrones pour obtenir la prédiction voulue.

Les exemples typiques de ce type d'algorithme sont les SNNs, dont le fonctionnement asynchrone par des impulsions binaires est parfaitement compatible avec le traitement des événements. Cette stratégie suscite un grand intérêt dans le domaine de la vision événementielle, comme décrit dans la Section 2.2.5.4.

2.3.4 Tâches de Vision et Bases de Données Existantes

Avec la démocratisation croissante des caméras événementielles depuis leur entrée sur le marché [LPDo8], le domaine de la vision événementielle attire toujours plus l'attention de la communauté scientifique, comme le montre la Figure 2.20. Cette expansion de la vision événementielle a naturellement conduit à une utilisation de plus en plus diversifiée des caméras événementielles. Par conséquent, plusieurs tâches de vision événementielle bien établies existent désormais au sein de la communauté scientifique, chacune ayant sa ou ses bases de données de référence pour évaluer les méthodes conçues par les pairs.

Dans cette section, nous débutons par une revue de plusieurs problématiques populaires en vision événementielle afin de donner un aperçu de la diversité des applications des caméras événementielles. Ensuite, nous introduisons deux catégorisations des bases de données événementielles existantes, en fonction des caractéristiques de leurs échantillons. Enfin, en nous basant sur ces classifications, nous procédons à une synthèse des bases de données populaires en vision événementielle.

2.3.4.1 Résumé de Tâches de Vision Existantes

Au fil des années, les chercheurs ont exploité les avantages des caméras événementielles dans diverses applications, engendrant différentes problématiques de vision où des méthodes de l'état de l'art sont évaluées et comparées. Dans le reste de cette section, nous présentons succinctement certaines des problématiques couramment abordées en vision événementielle, en mettant en évidence celles qui sont liées à nos travaux.

Reconnaissance d'Objets. L'un des premiers domaines d'application des caméras événementielles [Orc+15] concerne la reconnaissance d'objets, qui vise à prédire la catégorie d'un objet capturé en se basant sur le flux d'événements en entrée. Comme c'est le cas pour la vision artificielle conventionnelle, la reconnaissance d'objets demeure un sujet de recherche actif en vision événementielle [Fan+21b; Li+22b; Tan+22c], et elle offre une grande variété de bases de données, avec divers degrés de complexité, sur lesquelles comparer les algorithmes de l'état de l'art [Orc+15; Li+17; Kim+21].

Reconnaissance d'Actions. Similaire à la reconnaissance d'objets, la reconnaissance d'actions implique la classification du flux d'événements en fonction de l'action observée dans la scène capturée, comme les gestes humains [Ami+17; Liu+21a] ou la lecture sur les lèvres [Tan+22a]. Cette problématique diffère principalement de la

reconnaissance d'objets en ce qu'elle vise à classifier des mouvements spécifiques au lieu d'objets. Cependant, en raison de la nature asynchrone des capteurs événementiels, il n'est pas rare que les mêmes algorithmes de vision soient utilisés pour la reconnaissance d'objets et d'actions [Li+22b; Gu+21], ce qui est rarement le cas en vision artificielle classique, où la reconnaissance d'actions nécessite généralement un traitement vidéo distinct [Tra+15].

Détection d'Objets. La détection d'objets vise à localiser précisément et à classifier les objets d'intérêt présents dans une scène capturée. Cette tâche est critique dans de nombreux contextes, par exemple, pour la détection d'obstacles sur la route [Zha+18a]. Les caméras événementielles offrent une solution novatrice aux défis posés par la détection d'objets, notamment le flou cinétique et les conditions d'éclairage extrêmes. Par conséquent, des détecteurs basés sur les événements ont été développés pour relever ces défis, en particulier dans des environnements visuels exigeants [Lia+22]. L'une des applications les plus populaires de la détection d'objets par caméra événementielle est la navigation autonome [De +20; Per+20]. Les avantages des caméras événementielles, tels qu'une plage dynamique élevée et une réactivité accrue, en font des outils puissants pour améliorer la sécurité des systèmes autonomes. Une autre application émergente mérite d'être mentionnée, à savoir la détection de personnes dans des environnements divers [Mia+19; IKA23; Bor+23]. Les caméras événementielles exploitent leur sensibilité aux mouvements pour détecter plus efficacement les personnes qui se déplacent.

Tracking d'Objets. Le tracking d'objets va au-delà de la simple détection à un instant donné, car il implique de suivre la position des objets détectés dans la scène au fil du temps. Lorsqu'on utilise des caméras conventionnelles, le tracking d'objets peut être facilement perturbé, notamment lorsque les objets se déplacent rapidement, provoquant du flou cinétique. Les caméras événementielles se révèlent particulièrement efficaces pour résoudre ce genre de problème [Jia+20a]. Bien que de nombreuses bases de données de référence en matière de tracking d'objets événementiel proviennent de contextes très diversifiés [Wan+21; Zha+21; Tan+22b], elles incluent fréquemment des séquences mettant en scène des objets en mouvement rapide. Cela permet d'évaluer la capacité des algorithmes de vision événementielle à gérer le flou cinétique [Mes+23].

Segmentation Sémantique. La segmentation sémantique [BMD21] consiste à attribuer une catégorie spécifique à chaque pixel d'une image. Cette tâche est d'une grande importance en vision artificielle, car elle est utilisée dans de nombreuses appli-

cations, notamment dans le domaine de la conduite autonome [Sia+18]. L'utilisation de caméras événementielles pour la segmentation sémantique permet d'exploiter leur haute plage dynamique pour prédire des cartes de segmentation même dans des conditions d'éclairage difficiles (par exemple, la conduite de nuit [Sun+22]). La première méthode de segmentation sémantique basée sur des flux d'événements, EV-SegNet [AM19], a montré des performances prometteuses sur des bases de données événementielles capturées dans un contexte de conduite autonome [Bin+17; AM19]. Étant donné le manque d'informations spatiales dans les événements par rapport aux images statiques, de nombreuses approches en segmentation événementielle combinent les deux modalités (images RGB et événements) pour améliorer les performances [Sun+22; ZYS21].

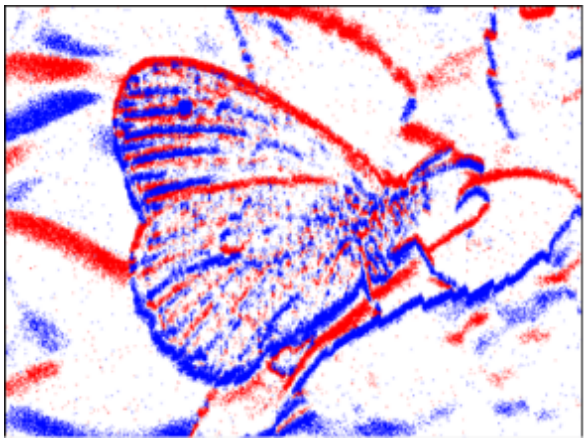
Estimation du Flux Optique. L'estimation du flux optique implique le calcul de la vélocité des objets présents dans la scène capturée. Alors que, pour les caméras conventionnelles, ceci est fait en réalisant une comparaison entre deux images consécutives d'une vidéo, les caméras événementielles permettent l'exploitation de l'information de mouvement exprimée par les événements. Deux approches principales sont envisageables pour estimer le flux optique à l'aide d'une caméra événementielle : **(1)** les méthodes traditionnelles basées sur la dynamique temporelle des événements [GRS18; SK19; SAG22], principalement pour le flux optique épars [PC21], qui peuvent être efficacement mises en œuvre sur du matériel spécialisé [LD22] ; et **(2)** les approches basées sur des réseaux de neurones (ANN [Zhu+18a] ou SNN [Lee+20]) pour estimer un flux optique généralement dense, qui sont formées à l'aide d'apprentissage supervisé [Kep+20] ou non-supervisé [Zhu+18a]. Cette tâche de vision revêt une grande importance dans de nombreuses applications car elle fournit des informations de bas niveau sur les mouvements dans la scène capturée.

Estimation de Profondeur. L'estimation de profondeur constitue un domaine de recherche majeur en vision événementielle [Gal+20], et elle peut être abordée selon diverses configurations (stéréo [Kog+11] ou monoculaire [KLD16], basée sur la géométrie épipolaire [Ben+11] ou grâce à des modèles d'apprentissage [Ran+21], impliquant ou non l'interaction avec la scène capturée [MCG15; SNB15], ...). Dans le contexte des réseaux de neurones profonds, deux grandes approches se dégagent : les approches monoculaires et stéréo. Les approches monoculaires pour l'estimation de profondeur se concentrent soit sur la prédiction de cartes de profondeur à partir des événements seuls [HGS20] (ce qui implique une perte significative d'informations spatiales par

rapport à l'utilisation d'images statiques), soit elles combinent l'emploi de caméras conventionnelles et événementielles pour améliorer les performances [Geh+21a]. D'autre part, l'estimation de profondeur stéréo, réalisée avec deux caméras ou plus, exploite la correspondance des événements provenant de ces caméras dans son processus d'apprentissage [Ran+21; Tul+19]. De manière similaire à l'estimation de profondeur monoculaire, les modèles d'apprentissage en stéréo peuvent bénéficier de l'intégration de modalités supplémentaires (typiquement, des images statiques) [MYC21].

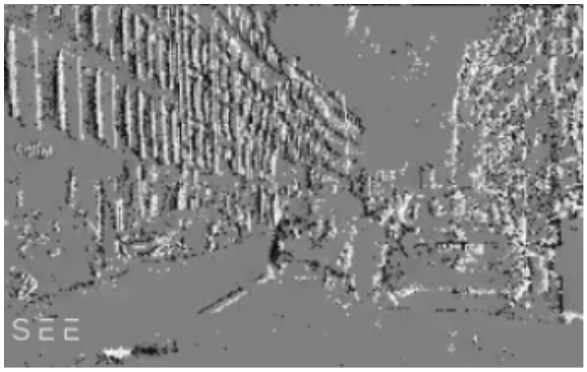
Estimation de Pose. La capacité des caméras événementielles à capturer efficacement le mouvement est un avantage important dans le cadre de l'estimation de pose de sujets humains, permettant de capter facilement les changements de poses de ceux-ci [Zou+21]. Cela est d'autant plus intéressant lorsque le sujet exécute des mouvements rapides, par exemple dans des contextes d'applications sportives [Wan+19]. La première approche basée sur l'apprentissage profond pour l'estimation de pose événementielle est présentée dans [Cal+19]. Cette méthode résout la tâche en exploitant les flux d'événements de plusieurs caméras événementielles calibrées depuis différentes perspectives. D'autres méthodes se concentrent sur l'utilisation d'une seule caméra événementielle, soit en la couplant à une caméra conventionnelle [Xu+20], soit en limitant l'usage d'images statiques à une seule image au début de la séquence pour extraire les informations spatiales initiales [Zou+21].

Améliorations et Restauration d'Images. L'expression "amélioration et restauration d'images" désigne l'ensemble des tâches en vision événementielle visant à améliorer la qualité des images statiques en exploitant les avantages des événements (notamment, une plage dynamique élevée et une faible latence temporelle) [Zhe+23]. Ces problématiques en vision événementielle sont souvent résolues en utilisant des modèles d'apprentissage profond. Parmi ces tâches, citons : l'interpolation d'images d'une séquence vidéo [Tul+21], qui consiste à augmenter artificiellement le taux d'images par seconde dans une vidéo ; la reconstruction d'événements en vidéo [Zhu+22a], qui vise à reconstituer une séquence vidéo avec une résolution temporelle élevée ; le défloutage d'images [Kim+22a], qui exploite les contours nets des événements pour corriger les effets de flou sur une image statique ; et l'augmentation de la plage dynamique d'une vidéo en utilisant des événements [WHY+19].

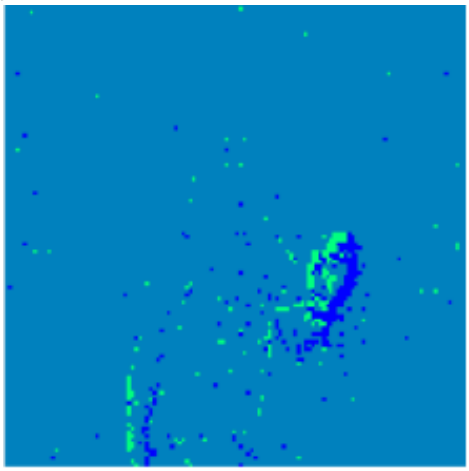


(a) Acquisition simulée et comportement statique. Base de données d'origine : N-ImageNet [Kim+21].

(b) Acquisition simulée et comportement dynamique. Base de données d'origine : DVS-UCF50 [Hu+16].



(c) Acquisition réelle et comportement statique. Base de données d'origine : Gen1 [De +20].



(d) Acquisition réelle et comportement dynamique. Base de données d'origine : DVSGesture [Ami+17].

Figure 2.21: Exemples des catégorisations de bases de données événementielles : par acquisition réelle/simulée et à comportement statique/dynamique.

2.3.4.2 Catégorisation des Bases de Données selon leur Acquisition

Il existe différentes approches pour la création de bases de données événementielles, en particulier en ce qui concerne la méthode d'acquisition des événements. La méthode la plus évidente consiste à capturer diverses scènes à l'aide d'une caméra. Cependant, cette approche demande des efforts considérables (par exemple, pour la capture et l'annotation des éléments de la base de données) [Kov+16], ce qui peut potentiellement ralentir le développement des méthodes en vision événementielle. Certains travaux adoptent des modes d'acquisition alternatifs [Orc+15; Kim+21; Hu+16] dans le but d'accélérer la création de bases de données événementielles requises pour entraîner

leurs modèles d'apprentissage. C'est pourquoi, il est possible de catégoriser les bases de données événementielles de l'état de l'art selon la méthode d'acquisition des événements. Nous pouvons distinguer deux types d'acquisition : **réelle** et **simulée**.

Acquisition Réelle. L'expression "acquisition réelle" fait référence à la méthode la plus directe pour créer une base de données, impliquant la capture de diverses scènes du monde réel à l'aide d'une caméra événementielle. Cette méthode vise à obtenir des flux d'événements réalistes dans des situations "dans la nature" ("*in the wild*", en anglais). En théorie, cette approche est à privilégier car elle se rapproche le plus des conditions d'utilisation authentiques d'une caméra événementielle. Ainsi, les performances d'un algorithme de vision événementielle évalué sur ce type de base de données sont plus susceptibles de refléter fidèlement ses performances dans le monde réel. Cependant, tout comme dans le cas des bases de données d'images statiques, la création d'une base de données à partir de zéro selon cette méthode s'avère très coûteuse en termes de temps et de ressources [Kov+16]. Cette démarche implique potentiellement des installations complexes visant à reproduire le contexte de la tâche de vision, telles que l'installation et la calibration d'une caméra sur un véhicule pour la conduite autonome [Geh+21b], la calibration d'autres équipements complémentaires [Cha+23], ou encore l'annotation des données capturées. Dans un domaine relativement récent comme la vision événementielle, ces défis peuvent constituer un obstacle à la diversification des tâches et des contextes d'application pour les caméras événementielles. En effet, chaque nouvelle application exige des efforts considérables pour construire la base de données nécessaire. Ces dernières années, on observe une augmentation du nombre de bases de données en acquisition réelle de grande envergure, couvrant divers domaines d'application, tels que la navigation autonome en environnement urbain [Bin+17; Hu+20; Geh+21b], la reconnaissance d'actions [Ami+17; Liu+21a; Wan+22a], la reconnaissance labiale [Tan+22a], la détection de personnes [Bor+23], etc. Les Figures 2.21c et 2.21d montrent des exemples de flux d'événements obtenus par acquisition réelle.

Acquisition Simulée. Les bases de données qualifiées d'"acquisition simulée" regroupent celles où les flux d'événements ont été générés dans des conditions simulées, c'est-à-dire qui ne correspondent pas à une capture dans un environnement naturel. En général, cette méthode implique la génération de flux d'événements à partir de vidéos ou d'images statiques. Bien que les échantillons ainsi obtenus puissent différer de ceux capturés dans le monde réel, la génération synthétique d'événements à partir

de données issues de caméras conventionnelles facilite considérablement la création de vastes bases de données événementielles. En effet, l'acquisition simulée permet de réutiliser une base de données d'images ou de vidéos (déjà annotée), ce qui permet d'obtenir de nombreux flux d'événements annotés avec un minimum d'efforts. En particulier, cette approche a permis aux premières solutions de vision événementielle [Orc+15; Li+17; Hu+16] d'acquérir rapidement des bases de données événementielles pour évaluer leurs performances. Ces bases de données pionnières ont été créées en capturant, avec une caméra événementielle, des écrans affichant des échantillons provenant d'ensemble de données d'images statiques. Même aujourd'hui, ces bases de données événementielles sont largement utilisées, et ce procédé d'acquisition est encore employé dans des travaux récents [Kim+21]. Cependant, cette approche nécessite une installation physique complexe, comme par exemple l'assemblage d'un écran et d'un capteur événementiel calibré sur un support physique [Orc+15]. Pour simplifier l'acquisition simulée, des travaux plus récents [RGS18; HLD21; Zhu+21; Geh+20] se sont concentrés sur la conversion de vidéos en événements, visant à générer des flux d'événements à partir de vidéos à l'aide d'outils logiciels, plus faciles à mettre en place qu'une installation physique. Ces outils logiciels ont considérablement progressé en termes de fonctionnalités, et ils peuvent désormais être utilisés dans des logiciels de création d'environnements 3D [RGS18], par exemple. Certains outils [HLD21] parviennent à générer des flux d'événements réalistes, en utilisant des modèles d'interpolation de vidéos [Jia+18] et en intégrant des bruits de capteurs [HLD21]. Des exemples de flux d'événements obtenus par acquisition simulée sont montrés dans les Figures 2.21a et 2.21b.

2.3.4.3 Catégorisation des Bases de Données selon leur Dynamique

Outre la méthode d'acquisition utilisée, les bases de données événementielles peuvent être classées en fonction de la dynamique spatio-temporelle des scènes capturées. Comme expliqué dans la Section 2.3.1, une caméra événementielle génère des événements principalement le long des contours des objets en mouvement par rapport à elle. Ainsi, si la caméra reste immobile, seuls les objets en mouvement sont enregistrés sous forme d'événements. En revanche, lorsque la caméra elle-même est en mouvement, tout son environnement apparaît en mouvement par rapport à elle, et les événements générés ressemblent grossièrement à une carte de contours de la scène entière (comme illustré en Figure 2.21c). En conséquence, les bases de données événementielles peuvent contenir des flux d'événements présentant des caractéristiques globales et une temporalité très différentes en fonction de la manière dont la caméra capture la scène. Dans la

littérature, nous identifions deux principaux types de dynamiques spatio-temporelles : les bases de données à comportement "statique" et celles à comportement "dynamique".

Bases de Données à Comportement Dynamique. Le terme "dynamique" ici fait référence aux bases de données où la forme des objets capturés évolue au fil du temps. Habituellement, ce type de données est obtenu lorsque la caméra événementielle reste immobile, ce qui permet de capturer uniquement les objets en mouvement dans la scène (par exemple, une personne réalisant des actions devant une caméra [Ami+17; Liu+21a], le déplacement de satellite en orbite dans le ciel [SF23], la détection de chutes [Wan+23]). Ces bases de données mettent en évidence les avantages de l'utilisation de caméras événementielles fixes dans l'espace, car cela permet de distinguer clairement le mouvement du reste de la scène. Les échantillons de ces bases de données consistent généralement en des flux d'événements d'une durée de plusieurs secondes (pour capturer l'intégralité de l'action) et les algorithmes de vision événementielle développés pour résoudre ces tâches tirent souvent parti du traitement spatio-temporel pour obtenir de meilleurs résultats (SNN [ICL21], 3D-CNN [Ber+23], ...). Les Figures 2.21b et 2.21d présentent des exemples de flux d'événements provenant de bases de données à comportement dynamique.

Bases de Données à Comportement Statique. Cette catégorie englobe les bases de données événementielles dont les flux d'événements ne représentent pas uniquement le mouvement, mais plutôt ressemblent davantage à des cartes de contours de la scène capturée [Orc+15; Sir+18; De +20]. De telles bases de données sont généralement obtenues en utilisant des caméras qui se déplacent dans l'espace, ce qui est courant dans de nombreuses applications (navigation autonome en environnement urbain [Sir+18; Bin+17; Hu+20; Geh+21b], robotique [Cha+23], acquisition simulée d'images statiques [Orc+15; Li+17; Kim+21], etc.). Plus concrètement, les flux d'événements générés ont une très courte durée (par exemple, 100 ms), et par conséquent, la forme des objets dans la scène ne change pas significativement sur cette courte période. On peut considérer que les échantillons d'une base de données statique n'expriment pas d'information temporelle significative, car l'objectif principal est d'extraire uniquement l'information spatiale de la scène (par exemple, à l'aide de 2D-CNNs [Sir+18], de descripteurs [Ram+19], ...). Les Figures 2.21a et 2.21c présentent des exemples d'échantillons provenant de bases de données à comportement statique.

Tâche	Nom	Année		Acquisition	Comportement
Reconnaissance d'Objets	N-MNIST [Orc+15]	2015	Adaptation de MNIST [LeC+98]	Simulée	Statique
	N-Caltch101 [Orc+15]	2015	Adaptation de Caltech101 [FFP04]	Simulée	Statique
	CIFAR10-DVS [Li+17]	2017	Adaptation de CIFAR10 [KH+09]	Simulée	Statique
	N-CARS [Sir+18]	2018	Classification voiture/personne dans un environnement urbain	Réelle	Statique
	N-ImageNet [Kim+21]	2021	Adaptation de ImageNet [Den+09]	Simulée	Statique
	ASL-DVS [Bi+20]	2020	Reconnaissance de langue des signes	Réelle	Statique
Reconnaissance d'actions	DVS-UCF-50 [Hu+16]	2016	Adaptation de UCF-50 [RS13]	Simulée	Dynamique
	DVSGesture [Ami+17]	2017	Reconnaissance de gestes humains	Réelle	Dynamique
	DailyAction-DVS [Liu+21a]	2021	Reconnaissance de gestes humains	Réelle	Dynamique
	HARDVS [Wan+22a]	2022	Reconnaissance de gestes humains	Réelle	Dynamique
	DVS-Lip [Tan+22a]	2019	Reconnaissance labiale	Réelle	Dynamique
	NEFER [Ber+23]	2023	Reconnaissance d'Expressions Humaine	Réelle	Dynamique
	Event-YawDD [Kie+23]	2023	Reconnaissance de bâillement en voiture	Simulée	Dynamique
Détection d'objets	Gen1 [De +20]	2020	Détection pour la conduite automatique	Réelle	Statique
	1Mpx Detection [Per+20]	2020	Détection pour la conduite automatique	Réelle	Statique
	PAF [Mia+19]	2019	Détection de personnes	Réelle	Statique
	Pedro [Bor+23]	2023	Détection de personnes	Réelle	Statique
	NU-AIR [IKA23]	2023	Détection de personnes et véhicules par un drône aérien	Réelle	Statique
Tracking d'objets	DVS-VOT15 [Hu+16]	2016	Adaptation de VOT15 [Kri+16]	Simulée	Dynamique
	FE240hz [Zha+21]	2021	Tracking d'objets filmés par une caméra DAVIS (images + événements)	Réelle	Dynamique
	VisEvent [Wan+21]	2021	Tracking d'objets divers combiné avec des images statiques	Réelle	Dynamique
Segmentation sémantique	DDD17 [Bin+17]	2017	Segmentation pour la conduite autonome	Réelle	Statique
	DDD20 [Hu+20]	2020	Évolution de DDD17 [Bin+17]	Réelle	Statique
	DSEC-Semantic [Geh+21b; Sun+22]	2022	Segmentation pour la conduite autonome	Réelle	Statique
Estimation de flux optique	MVSEC [Zhu+18b]	2018	Contextes de navigations autonomes (terrestre et aérienne)	Réelle	Statique
	DSEC [Geh+21b]	2021	Contexte de conduite en environnement urbain	Réelle	Statique
Estimation de profondeur	MVSEC [Zhu+18b]	2018	Estimation de profondeur stéréo pour la navigation terrestre et aérienne	Réelle	Statique
	DSEC [Geh+21b]	2021	Estimation de profondeur stéréo pour la conduite en environnement urbain	Réelle	Statique
	VIVID++ [Lee+22]	2022	Estimation de profondeur dans des conditions d'illumination difficiles	Réelle	Statique
Estimation de pose	DHP19 [Cal+19]	2019	Poses humaines en 3D capturées par 4 caméras événementielles synchronisées	Réelle	Dynamique
	EventCap [Xu+20]	2020	Poses humaines en 3D capturée par une caméra événementielle et une image statique initiale	Réelle	Dynamique
	SynEventHPD [Zou+23a]	2022	Base de données de grande échelle convertie à partir de nombreuses bases d'images statiques	Simulée	Dynamique
Améliorations et Restauration d'images	HQF [Sto+20]	2020	Reconstruction d'images	Réelle	-
	DVSNoise20 [Bal+20]	2020	Correction de bruits et super-résolution	Réelle	-
	IJRR [Mue+17]	2017	Reconstruction de vidéos	Réelle	-

Table 2.5: Liste de bases de données événementielles de l'état de l'art, avec leurs catégorisations (acquisition simulée/réelle et comportement statique/dynamique) et les contextes applicatifs.

2.3.4.4 Synthèse des Bases de Données de Référence

La Table 2.5 résume certaines des bases de données événementielles majeures de l'état de l'art, en mettant en relation les types de problèmes de vision (voir Section 2.3.4.1) et les catégories de bases de données (voir les Sections 2.3.4.2 pour l'acquisition simulée ou réelle, et 2.3.4.3 pour le comportement statique ou dynamique).

D'abord, nous observons que les problèmes de vision événementielle sont souvent liés à un type de dynamique spatio-temporelle. Par exemple, la segmentation sémantique est évaluée principalement sur des bases de données à *comportement statique* [Bin+17; Hu+20; Geh+21b], car elle repose sur l'information spatiale. Les bases de données à comportement dynamique ne sont pas adaptées car elles ne permettraient de segmenter que les objets en mouvement. À l'inverse, la reconnaissance d'actions utilise des bases de données à *comportement dynamique*, car elles captent les mouvements essentiels effectués par le sujet humain pour la reconnaissance. Ainsi, la catégorisation du comportement spatio-temporel est souvent liée au contexte de la tâche de vision.

Ensuite, nous remarquons que les bases de données événementielles récentes privilégient généralement une *acquisition réelle* par rapport à une acquisition simulée [Wan+22a; De +20]. Cela résulte de l'accessibilité croissante des caméras événementielles et de l'intérêt croissant de la communauté pour cette technologie. En outre, les bases de données à acquisition simulée sont généralement des travaux moins récents où l'adaptation de bases de données d'images statiques était un moyen populaire d'obtenir rapidement des ensembles de données événementielles à moindre effort [Orc+15; Li+17]. Néanmoins, l'acquisition simulée reste pertinente pour de nombreux travaux [Kie+23; Mue+17], grâce à des outils de conversion "vidéo-vers-événements" simples [HLD21], permettant d'explorer rapidement le potentiel des caméras événementielles dans de nouveaux contextes.

Enfin, notre revue des bases de données événementielles révèle une diversification croissante des contextes d'application. Alors que les premiers travaux se concentraient principalement sur la reconnaissance générique d'objets (N-MNIST [Orc+15], CIFAR10-DVS [Li+17], ...), la vision événementielle s'étend désormais à des problèmes plus appliqués (conduite autonome [Bin+17; Geh+21b], reconnaissance labiale [Ber+23]) dans divers domaines (sécurité routière [Kie+23], aérospatial [SF23], ...). Cette tendance témoigne de la croissance de la vision événementielle et de son utilité dans de nombreux domaines.

2.3.5 Verrous Scientifiques

Cette Section 2.3 a présenté une vue d'ensemble de la vision événementielle. Nous y avons résumé les avancées clés dans ce domaine, notamment les caméras événementielles, les méthodes de représentation des événements, et les problématiques de vision événementielle abordées dans l'état de l'art. De plus, nous avons formulé des concepts fondamentaux pour nos travaux, tels que la représentation par images événementielles binaires et la génération d'événements, tout en définissant des termes essentiels, notamment les catégorisations des bases de données événementielles.

Tout au long de cette revue de la vision événementielle, nous avons identifié des problèmes ouverts dans l'état de l'art, notamment deux défis spécifiques que nous visons à résoudre dans nos travaux. Le reste de cette section expose ces défis scientifiques que nous avons identifiés.

Premièrement, nous avons constaté une grande diversité dans les techniques de représentation des événements dans l'état de l'art, comme en témoignent les différents paradigmes présentés en Section 2.3.3. En ce qui concerne le traitement des événements par des algorithmes de vision "classique" (c'est-à-dire des algorithmes généralement destinés au traitement d'images statiques), plusieurs approches sont envisageables. Parmi celles-ci, on peut citer les images événementielles, qui capturent les informations spatiales d'un flux d'événements, ou les surfaces temporelles, qui mettent davantage en évidence la temporalité des événements. Néanmoins, il est encore rare de voir des représentations d'événements simples, telles que les images événementielles, intégrer une forme d'information temporelle sans avoir recours à des séquences d'images, ce qui peut être coûteux en termes de mémoire. Nous formulons donc l'hypothèse que l'intégration d'informations temporelles dans les images événementielles pourrait offrir des avantages significatifs pour les algorithmes de vision classique adaptés aux événements.

Deuxièmement, bien que la vision événementielle s'étende de plus en plus vers de nouvelles problématiques et des contextes d'application variés, la création des bases de données événementielles nécessaires à ces évolutions reste un défi majeur. Cette tâche exige des ressources considérables en termes de temps et de moyens pour capturer et annoter les flux d'événements à l'aide de caméras événementielles. Alors que la conversion "vidéo-vers-événements" s'avère être une méthode efficace pour obtenir rapidement des ensembles de données événementielles, la question de la cohérence entre les événements synthétiques et les conditions réelles de déploiement des algorithmes

de vision événementielle demeure. Plutôt que de s'appuyer massivement sur des données simulées, une approche intéressante consisterait à permettre un entraînement plus efficace des modèles de vision événementielle sur des ensembles de données événementielles réelles plus restreints, à condition de mettre au point des techniques permettant de réduire les besoins en données annotées pour l'apprentissage des modèles.

2.4 Conclusion

Dans ce Chapitre 2, nous avons offert une vue d'ensemble des travaux de l'état de l'art relatifs aux sujets cruciaux pour notre recherche. Nous avons commencé par récapituler brièvement les évolutions majeures des ANNs dans le contexte de l'apprentissage profond en vision artificielle, en mettant en lumière les enjeux liés à leur consommation énergétique. Ensuite, nous avons introduit les SNNs comme une solution prometteuse pour concevoir des modèles d'apprentissage profond à faible consommation énergétique. Nous avons également mentionné les défis actuels associés à la conception des SNNs pour la vision artificielle. Notre exploration s'est ensuite approfondie dans le domaine émergent de la vision événementielle, une sous-discipline de la vision artificielle axée sur le traitement de flux d'événements asynchrones, compatibles avec les SNNs. Nous avons abordé des aspects essentiels de ce domaine, tout en identifiant des défis scientifiques pertinents, en particulier en ce qui concerne la conception et l'entraînement de méthodes basées sur les réseaux de neurones profonds. Grâce à cette compréhension approfondie des technologies évoquées (ANN, SNN, et flux d'événements), nous sommes maintenant en mesure de discerner les pistes de recherche qui revêtent une importance particulière pour nos travaux.

Une première voie de recherche que nous explorons aborde la question des représentations d'événements pour la vision événementielle en utilisant des approches traditionnelles, celles qui traitent principalement des images statiques. Comme exposé en Section 2.3.5, la représentation en images événementielles demeure très courante dans l'état de l'art [Zhe+23] en raison de sa facilité d'utilisation et de sa simplicité de mise en œuvre. Cependant, cette représentation omet complètement la dimension temporelle des événements, contrairement à des représentations potentiellement plus complexes, telles que les surfaces temporelles. Cette lacune met en évidence l'importance d'incorporer de manière efficiente des informations temporelles dans les images événementielles. Cela permettrait de tirer parti de l'apport d'informations spatio-temporelles pour améliorer les performances des algorithmes conventionnels, notamment les 2D-CNNs. Dans notre travail présenté au Chapitre 3, nous cherchons à résoudre cette problématique en développant une représentation en images événementielles qui intègre efficacement la dimension temporelle des événements.

Ensuite, une deuxième piste de recherche que nous envisageons se focalise sur les SNNs profonds, spécifiquement entraînés avec la BPTT et le substitut du gradient. Comme mentionné dans la Section 2.2.7, nous avons constaté l'efficacité de cette règle

d'apprentissage pour la conception d'architectures SNN profondes, notamment les CSNNs, et leur adoption croissante pour résoudre une gamme toujours plus étendue de tâches de vision artificielle. Toutefois, il est essentiel de noter que ces nouveaux modèles SNNs ne réagissent pas nécessairement de la même manière que les technologies plus anciennes examinées dans l'état de l'art, telles que les ANNs ou les SNNs entraînés par STDP. Par conséquent, des investigations approfondies sur le comportement des SNNs profonds, en tenant compte de divers choix de conception fondamentaux, devraient être menées simultanément à leur développement pour aborder de manière efficace une multitude de problèmes de vision. Dans nos travaux exposés au Chapitre 4, nous empruntons cette voie en élaborant des modèles CSNN profonds adaptés à des nouvelles tâches de vision artificielle, qu'il s'agisse d'images statiques ou de flux d'événements. En parallèle de la résolution de ces nouvelles problématiques de vision, nous entreprenons des analyses expérimentales approfondies sur divers aspects fondamentaux des SNNs. Cela englobe des considérations telles que les schémas de codages neuronaux pour traiter les images statiques ainsi que la robustesse des modèles face aux bruits des capteurs.

Enfin, la troisième voie de recherche que nous explorons aborde la problématique soulevée dans la Section 2.3.5, concernant les défis liés à l'entraînement de modèles profonds en vision événementielle qui requièrent une quantité significative de données annotées. Même si la solution consistant à générer des flux d'événements synthétiques permet d'obtenir rapidement de vastes bases de données annotées, les données événementielles ainsi obtenues peuvent différer considérablement du contexte d'application réel dans lequel l'algorithme de vision doit opérer. Dans nos travaux exposés au Chapitre 5, nous adoptons une approche différente pour résoudre ce problème. Au lieu de synthétiser des événements annotés, nous cherchons à réduire la nécessité de données annotées pour l'entraînement de modèles d'apprentissage. Notre démarche repose sur un pré-entraînement initial d'un modèle ANN ou SNN profond, lui permettant d'être ajusté ultérieurement avec un volume de données limité dans le cadre d'une nouvelle application en vision événementielle. Cette approche vise à minimiser les efforts requis pour la création d'une base de données en acquisition réelle, tout en conservant des performances satisfaisantes pour l'algorithme de vision.

Pour conclure, ces trois pistes de recherche offrent des moyens de résoudre diverses limitations des approches neuromorphiques en vision artificielle. Elles partagent toutes un objectif commun : améliorer notre compréhension de ces technologies émergentes tout en proposant des solutions visant à renforcer leur aptitude à résoudre les défis

de vision contemporains.

3

Bina-Rep : une Méthode Simple et Efficace pour la Représentation d'Événements en Images

Sommaire

Comme expliqué en Section 2.3.3.1, les représentations en images événementielles sont construites à partir de l'apparition des événements pour chaque pixel, entraînant la perte de l'information temporelle des événements accumulés [Zhe+23]. Plus précisément, les deux techniques d'images événementielles principales, les images événementielles binaires [KSK09] et histogrammes d'événements [Maq+18], encodent avant tout l'information spatiale, mais il n'est plus possible de savoir dans quel ordre ces événements ont eu lieu à partir des valeurs de pixels dans l'image obtenue.

Une solution largement employée consiste à décomposer l'ensemble d'événements $\mathcal{E}$ en sous-ensembles consécutifs afin de créer une séquence d'images événementielles [WHY+19]. Cependant, cela nécessite de traiter plusieurs images avec l'algorithme de reconnaissance sous-jacent, ce qui pose un nouveau problème de lourdeur en termes de mémoire et puissance de calcul.

Dans ce chapitre, nous proposons une méthode de représentation d'événements en images événementielles, nommée *"Bina-Rep"*, qui a la particularité d'incorporer de l'information temporelle tout en étant plus compacte qu'une séquence d'images. Pour fournir une comparaison équitable avec d'autres méthodes de représentation d'événements, nous concevons un 2D-CNN basique de référence et évaluons ses performances en classification avec Bina-Rep et d'autres techniques populaires de l'état de l'art. Nos différentes expérimentations montrent que Bina-Rep atteint des résultats compétitifs en termes de précision ainsi qu'en robustesse contre des corruptions communes de caméras événementielles.

Ce chapitre est organisé de la manière suivante : d'abord, nous formulons, en Section 3.1, la méthodologie proposée pour obtenir des images événementielles "Bina-Rep". Ensuite, nous définissons, en Section 3.2, le modèle 2D-CNN de référence qui est utilisé dans nos expérimentations avec diverses représentations d'événements. En Section 3.3, nous détaillons les résultats de nos expérimentations, puis apportons nos conclusions en Section 3.4.

3.1 Méthode : Bina-Rep

"Bina-Rep" (pour *"**Bina**ry-**Rep**resentation"*) est une approche qui consiste à construire des images événementielles (que l'on appelle "images événementielles Bina-Rep") avec pour objectif d'encoder une forme d'information temporelle en restant efficace en termes de calcul et de mémoire. Autrement dit, la méthode proposée vise à réduire les calculs nécessaires pour incorporer l'information temporelle du flux d'événements, tout en assurant que l'image événementielle en sortie soit compacte pour le traitement par un algorithme de vision.

3.1.1 Images Événementielles Bina-Rep

La création d'une image événementielle Bina-Rep se base sur une séquence d'images événementielles binaires successives. Comme expliqué en Section 2.3.3.1, on définit une séquence de T sous-ensembles d'événements $\{\mathcal{E}_t\}_{t=1}^{T}$ d'un flux d'événements donné $\mathcal{E}$, pour reconstruire une séquence d'images événementielles binaires $\mathbf{X}_T \in \mathbb{B}^{T\times C\times H\times W}$ où $C = 2$ est le nombre de canaux correspondant aux deux polarités de la caméra. Cette séquence $\mathbf{X}_T$ peut aussi être appelée "**tenseur d'impulsions**" ou "**tenseur impulsionnel**".

Dans cette séquence d'images binaires $\mathbf{X}_T$, on retrouve une suite de T bits consécutifs pour chaque indice coordonnées-polarité (x, y, p) tel que :

$$\mathbf{X}_T[x, y, p] = \{\mathbb{1}(x, y, p)\}_{t=1}^{T} \tag{3.1}$$

L'idée principale dans Bina-Rep consiste à réinterpréter ces suites de T bits en une représentation numérique à T-bits pour générer l'image événementielle Bina-Rep, de manière similaire à [Inn+21] mais en prenant en compte les deux polarités d'une caméra événementielle. Après une étape de normalisation [PS15], la sortie finale est désignée par $X_T^{\text{BinaRep}} \in \mathbb{R}^{C\times H\times W}$. La Figure 3.1 montre une illustration du processus de création d'une image Bina-Rep.

En guise de cas particulier, une image événementielle Bina-Rep avec $T = 1$ est équivalente à une image événementielle binaire, tel que défini dans la Section 2.3.3.1.

Séquence d'Images Bina-Rep. De la même manière que pour les autres méthodes d'images événementielles, les images Bina-Rep peuvent être générées sur des sous-

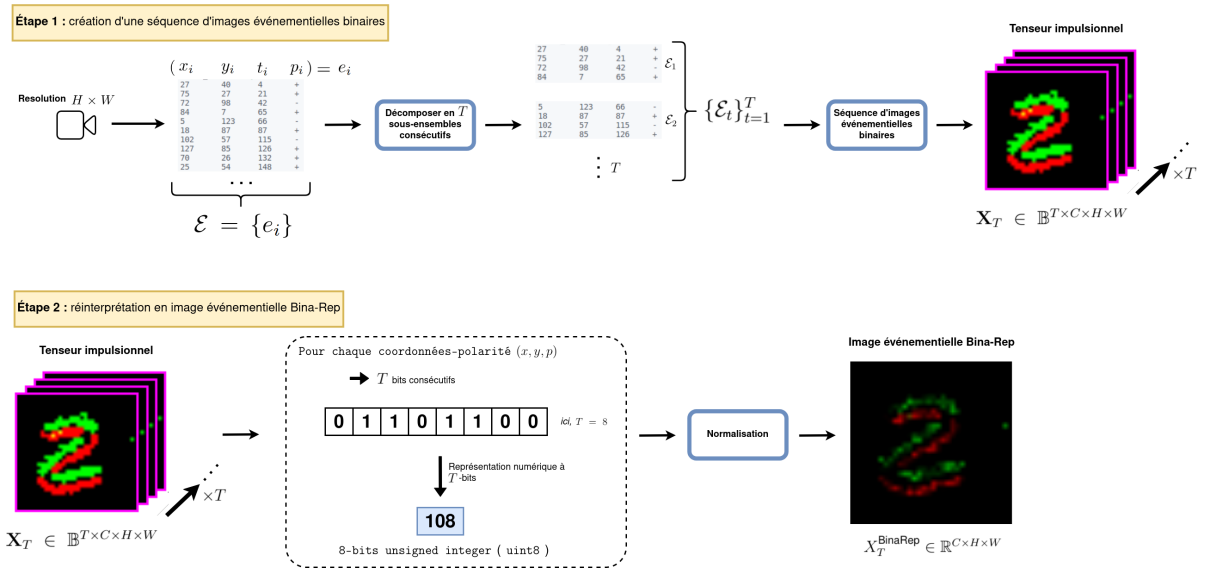


Figure 3.1: Création d'une image événementielle Bina-Rep.

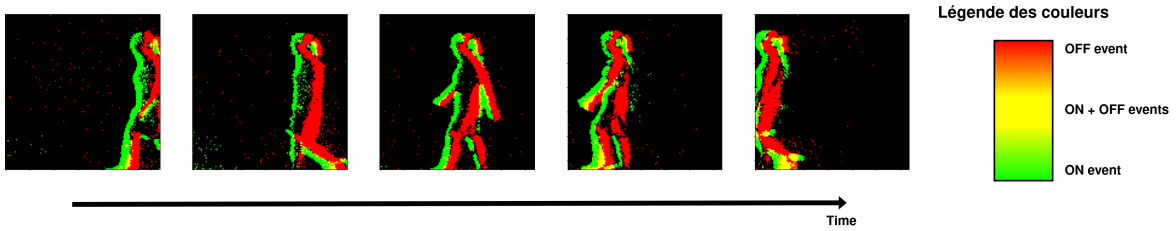
ensembles d'événements consécutifs, donnant une séquence d'images Bina-Rep. La séquence d'images Bina-Rep est désignée par $\mathbf{X}_{K,T}^{\text{BinaRep}} \in \mathbb{R}^{K \times C \times H \times W}$, où K correspond au nombre d'images Bina-Rep créées. Par conséquent, générer une séquence de K images Bina-Rep en représentation à T-bits nécessite $K \times T$ images événementielles binaires.

3.1.2 Différences avec les Représentations Similaires

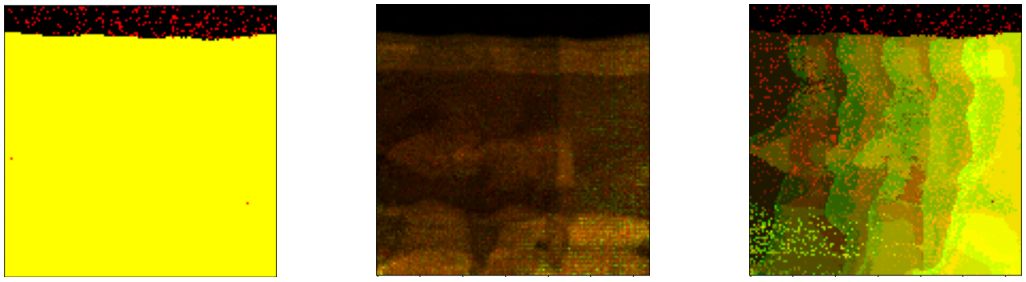
La méthode proposée se différencie des autres approches d'images événementielles par l'information qui est encodée pour chaque pixel. Comme discuté en Section 2.3.3.1, chaque pixel d'une image événementielle binaire encode la présence ou l'absence d'événement, alors que les pixels d'un histogramme d'événements retiennent le nombre d'événements apparus dans le flux original. En comparaison, un pixel d'une image Bina-Rep encode :

1. **La présence d'événement :** si la valeur d'un pixel est supérieure à 0, un événement est présent dans le flux original.

2. **Une approximation du nombre d'événements :** le nombre d'événements est indirectement encodé, et peut-être compris entre 0 et T. Au-delà, la méthode est *"saturée"*, c'est-à-dire que la représentation numérique de T-bits est limitée. C'est pourquoi il faut mentionner que l'expression du nombre d'événements est bien plus limitée que pour les histogrammes d'événements.



(a) Visualisation de la scène sous forme de séquence d'images événementielles.



(b) Image événementielle binaire. **(c)** Histogramme d'événements. **(d)** Image événementielle Bina-Rep.

Figure 3.2: Mise en lumière de l'intérêt de Bina-Rep par rapport aux autres images événementielles. La séquence utilisée provient de [Liu+21a] et illustre un sujet humain qui marche de la droite vers la gauche.

3. **L'ordre d'apparition des événements :** les valeurs des pixels changent en fonction de l'ordre d'apparition des événements dans la séquence d'images binaires $\mathbf{X}_T$. C'est dans cet ordre d'apparition encodé que réside une approximation de l'information temporelle du flux d'événements original.

Mise en Lumière par l'Exemple. La Figure 3.2 montre un exemple typique de l'intérêt d'employer Bina-Rep, par rapport à une représentation en une image événementielle binaire ou un histogramme d'événements. Le flux d'événements représente un sujet humain qui marche de la droite vers la gauche (visualisé en Figure 3.2a). Par conséquent, le motif spatio-temporel exprimé est un mouvement, relativement long ($\approx$ 2.2 secondes), de la droite vers la gauche. Comme montré en Figure 3.2b, une accumulation en une seule image événementielle binaire n'est pas exploitable, car l'action est trop longue et entraîne un phénomène de saturation. L'histogramme d'événements (montré en Figure 3.2c) est moins sensible à ce phénomène, mais la perte de la dimension temporelle des événements crée un motif qui ne permet plus de constater l'action capturée. En comparaison, l'utilisation de Bina-Rep (montré en Figure 3.2d) permet de voir distinctement des "instants" dans l'action (et donc, l'évolution du motif capturé dans le temps), grâce à l'ordre d'arrivée encodé par les pixels.

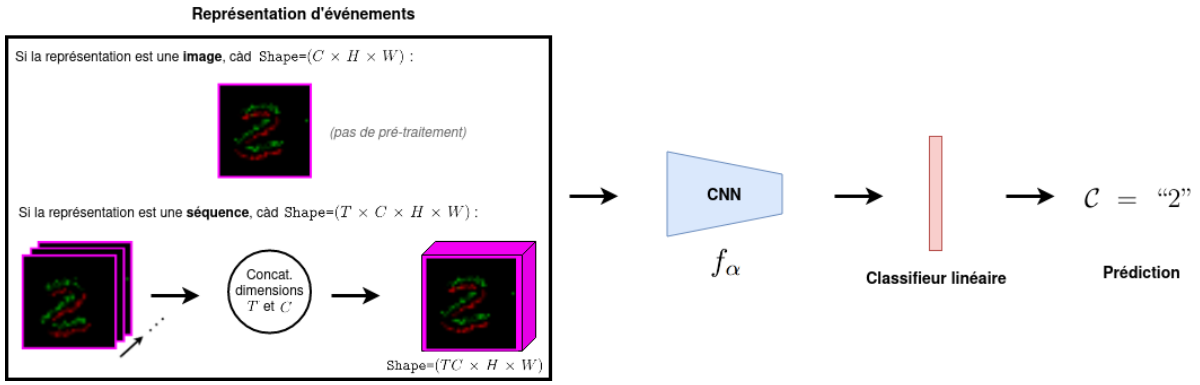


Figure 3.3: Modèle 2D-CNN de référence utilisé dans nos expérimentations avec différentes techniques de représentations d'événements. Un pré-traitement est appliqué sur les représentations en séquence afin que le tenseur obtenu ait trois dimensions, adéquat au traitement par un 2D-CNN.

3.2 Méthode : Réseau de Neurones Convolutif de Référence

Pour étudier les impacts de Bina-Rep en comparaison à d'autres techniques de représentations d'événements dans l'état de l'art, nous concevons un modèle de réseau de neurones de référence visant à réaliser des tâches de classification sur des bases de données événementielles communément utilisées [Ami+17; Orc+15; Li+17; Sir+18].

Comme illustré dans la Figure 3.3, le modèle de référence est composé d'un 2D-CNN $f_\alpha(\cdot)$ pour extraire un vecteur de caractéristiques (de manière similaire à la formulation décrite en Section 2.1.2), et d'une couche linéaire en fin du 2D-CNN pour obtenir un score de classification entre $\mathcal{C}$ catégories (propre à la base de données utilisée). Le 2D-CNN employé est une architecture ResNet-18 [He+16]. Enfin, le modèle est entraîné en utilisant la fonction de **coût de cross-entropie** (*"cross-entropy loss"*, en anglais).

Dans nos expérimentations, certaines représentations évaluées sont constituées de **séquences** d'images, ce qui signifie que le tenseur en entrée du modèle peut être de 4 dimensions (de la forme $T \times C \times H \times W$, où T est le nombre d'images dans la séquence). Comme un 2D-CNN ne peut traiter que des images, dans la forme de tenseurs à trois dimensions (par exemple, une image RGB est de la forme $3 \times H \times W$), une opération optionnelle de concaténation des dimensions T et C est appliquée si nécessaire, comme montré dans la partie gauche de la Figure 3.3. En conséquence, le modèle de référence présenté accepte des séquences d'images événementielles sous la forme $TC \times H \times W$.

3.3 Expérimentations

Nous testons notre méthode par rapport à des approches de l'état de l'art en utilisant le modèle 2D-CNN de référence décrit dans la Section 3.2. Nous évaluons la performance de notre méthode sur différentes bases de données événementielles en classification. Outre la précision pure du modèle pour une représentation d'événements donnée, nous étudions la robustesse de ce modèle contre des corruptions des flux d'événements.

3.3.1 Configuration des Expériences

3.3.1.1 Représentations d'Événements Étudiées

Comme Bina-Rep est une représentation d'événements catégorisée comme une "image événementielle", nous évaluons également les images événementielles binaires et les histogrammes d'événements. Concernant les images événementielles binaires, nous employons celles-ci en séquence d'images pour assurer une comparaison équitable car notre approche se base elle-même sur des séquences d'images binaires. Cependant, l'histogramme d'événements est appliqué qu'en une seule image et pas en séquences, étant donné que c'est une manière commune d'utiliser cette représentation.

En plus de la comparaison avec des techniques d'images événementielles, nous évaluons Voxel Grid [Zhu+19], une technique de représentation en voxels, pour étudier les différences entre Bina-Rep et une technique de référence de représentation dans un espace en trois dimensions.

Concernant Bina-Rep, nous évaluons son efficacité en utilisant une configuration à une seule image, ainsi qu'en séquences d'images. Nous adoptons une représentation en nombre entier à 8 bits non signés ("`8-bits unsigned integer`" ou "`uint8`") pour toutes les expériences, soit $T = 8$.

Pour finir, la Table 3.1 reprend les méthodes de représentation évaluées et précise leurs paramètres.

3.3.1.2 Bases de Données Employées

Comme expliqué en Section 2.3.4, les bases de données événementielles peuvent être catégorisées soit selon la méthode d'acquisition (*"réelle"* ou *"simulée"*) des flux d'événements, soit selon la dynamique de ceux-ci (*"comportement statique"* ou *"comportement dynamique"*). Nous sélectionnons des bases de données populaires en classification

Représentation d'Événements	C	T	# de canaux (*ex.* TC)
Voxel Grid [Zhu+19]	-	10	10
Images événementielles binaires [KSK09]	2	10	20
Histogramme d'événements [Maq+18]		1	2
Bina-Rep (contrib.)		1	2
Bina-Rep (contrib.)		3	6

Table 3.1: Résumé des paramètres des représentations d'événements étudiées.

Base de Données	Résolution	Durée	#Échantillons Total	Entraînement	Validation	Classes
N-MNIST [Orc+15]	34×34	≈ 0.3s	70000	60000	10000	10
CIFAR10-DVS [Li+17]	128×128	≈ 1.2s	10000*	8000*	2000*	10
N-CARS [Sir+18]	304×240	≈ 0.1s	24029	15422	8607	2
DVSGesture [Ami+17]	128×128	≈ 6s	1342	1078	264	11

Table 3.2: Récapitulatif des bases de données étudiées. * : décomposition non-officielle de la base de données en ensembles d'entraînement et validation, car il n'existe pas de décomposition officielle.

selon ces catégories. La Table 3.2 reprend les statistiques des bases de données utilisées.

Nous utilisons les bases de données suivantes :

- **CIFAR10-DVS** [Li+17] et **N-MNIST** [Orc+15] **:** ces bases de données, caractérisées par une **acquisition simulée** et un **comportement statique**, sont historiquement les plus utilisées pour évaluer les performances de modèles en classification et donc restent un bon moyen d'évaluer les performances d'une nouvelle approche. Néanmoins, de par son "acquisition simulée", ces flux d'événements diffèrent fortement d'un scénario d'utilisation réel de caméra événementielle. C'est pourquoi des évaluations supplémentaires avec d'autres bases de données sont requises.

- **N-CARS** [Sir+18] **:** caractérisée par des flux d'événements en **acquisition réelle** mais au **comportement statique**, N-CARS est une base de données événementielles qui est représentative d'un grand nombre de bases de données existantes dans l'état de l'art [De +20; Per+20; Geh+21b; Bin+17] (à savoir, la capture de données pour la conduite de voitures autonomes) et illustre un cas d'application courant de vision événementielle.

- **DVSGesture** [Ami+17] **:** cette base de données événementielles est composée de flux d'événements en **acquisition réelle** avec un **comportement dynamique**. Concrètement, elle consiste à faire la classification d'actions exécutées par des sujets humains filmés grâce à une caméra DVS immobile. Dans le cadre de notre analyse, l'étude de DVSGesture est particulièrement intéressante, car elle permet d'étudier l'efficacité d'une représentation d'événements à exprimer des informations spatio-temporelles qui changent dans le temps.

3.3.1.3 Autres Détails d'Implémentation

Les expériences sont implémentées avec la librairie PyTorch [Pas+19] en utilisant un GPU NVIDIA Tesla P100. Les flux d'événements sont traités via la librairie Tonic [Len+21]. Tous les modèles ont été entraînés à partir d'une initialisation aléatoire (c'est-à-dire sans pré-entraînement) pendant 60 époques. Pour chaque base de données, nous avons redimensionné la résolution de la caméra à $(H \times W) = (224 \times 224)$. Nous avons utilisé l'optimiseur Adam [KB14] avec un taux d'apprentissage initial de 0.001, que nous avons divisé par 10 toute les 15 époques. L'implémentation naïve de Bina-Rep est disponible publiquement dans la librairie Tonic [Len+21] : `https://github.com/neuromorphs/tonic`.

3.3.2 Comparaison des Représentations d'Événements

La Table 3.3 détaille les résultats des expérimentations pour chaque base de données. En général, l'approche proposée obtient des résultats compétitifs pour toutes les bases de données. Plus précisément, elle surpasse les autres méthodes sur N-MNIST [Orc+15] et présente la deuxième meilleure performance sur N-CARS [Sir+18] et DVSGesture [Ami+17]. Nous observons généralement de meilleures performances avec plusieurs images ($T = 3$), ce qui peut s'expliquer par une décomposition plus précise du flux original d'événements.

Les résultats observés pour DVSGesture [Ami+17] illustrent bien l'intérêt d'étudier une base de données à comportement dynamique. En effet, nous constatons de grandes variations dans les performances selon la représentation d'événements utilisée, alors que les écarts de performance dans les autres bases de données (dont le comportement est statique) restent faibles. L'utilisation de séquence d'images Bina-Rep ($T = 3$) atteint la deuxième meilleure précision, ce qui montre l'intérêt de Bina-Rep pour ce cas d'utilisation. Notre approche avec $T = 3$ obtient également de meilleures

Représentation	N-MNIST [Orc+15]	CIFAR10-DVS [Li+17]	N-Cars [Sir+18]	DVSGesture [Ami+17]
Voxel Grid [Zhu+19]	99.43	**94.00**	90.71	79.55
Images événementielles binaires [KSK09]	99.36	93.45	**92.48**	82.95
Histogramme d'événements [Maq+18]	99.14	93.65	91.77	**90.91**
Bina-Rep $T = 1$ (contrib.)	99.34	92.4	92.04	80.68
Bina-Rep $T = 3$ (contrib.)	**99.52**	93.25	90.74	87.88

Table 3.3: Comparaison des représentations d'événements étudiées en utilisant le modèle CNN de référence présenté en Section 3.2.

performances que les images binaires avec $T = 10$, confirmant la meilleure expressivité de Bina-Rep tout en restant plus compacte que des séquences d'images binaires. De manière intéressante, l'histogramme d'événements surpasse les autres méthodes alors que cette représentation ne prend pas en compte la dimension temporelle. Ce phénomène peut être expliqué par le fait que les motifs créés par l'histogramme sont déjà efficaces pour discriminer les classes entre elles, d'autant plus que les histogrammes d'événements sont moins susceptibles de présenter des valeurs saturées. D'autre part, la configuration à une seule image Bina-Rep ($T = 1$) obtient le pire score de précision. Cela peut s'expliquer par le fait qu'une seule image Bina-Rep ne peut pas intégrer une grande quantité d'événements provenant de séquences temporelles plus longues sans saturer la représentation en 8 bits des pixels.

3.3.3 Comparaison avec les Méthodes Existantes

Nous mettons en perspective les résultats obtenus par le modèle de référence entraîné sur des images Bina-Rep avec les résultats observés dans l'état de l'art (montré en Tableau 3.4). De manière générale, nous observons que notre méthode surpasse les travaux précédents sur plusieurs ensembles de données, démontrant l'efficacité de notre modèle simple pour effectuer des tâches de reconnaissance. Cependant, il est important de mentionner que les modèles utilisés peuvent différer considérablement d'une méthode à l'autre. Par exemple, DiST [Kim+21] utilise un 2D-CNN ResNet-34, tandis que HATS [Sir+18] est basé sur un SVM linéaire.

3.3.4 Analyse de Robustesse aux Corruptions

En plus des scores de précision obtenus sur des "cas idéaux" (c'est-à-dire des cas où les flux d'événements ne sont pas corrompus), il est important d'étudier la robustesse d'une méthode de représentation d'événements contre des entrées altérées. C'est

Méthode	# Paramètres (approx.)	N-MNIST [Orc+15]	CIFAR10-DVS [Li+17]	N-Cars [Sir+18]
HATS [Sir+18]	-	99.1	52.4	90.2
PLIF SNN [Fan+21b]	17M	**99.6**	74.8	-
DiST [Kim+21]	21M	-	62.57	90.80
EvS-B [Li+21a]	-	-	68.0	**93.1**
Bina-Rep $T = 1$ (ours)	11M	99.34	<u>92.4</u>	<u>92.04</u>
Bina-Rep $T = 3$ (ours)	11M	<u>99.52</u>	**93.25**	90.74

Table 3.4: Comparaison des performances de notre approche (modèle 2D-CNN de référence + Bina-Rep) **avec les résultats de travaux existants.**

pourquoi nous analysons les impacts de corruptions communes de vision événementielle sur les performances. Les techniques de représentation d'événements analysées sont les mêmes que celles étudiées en Section 3.3.2.

3.3.4.1 Cadre d'Expérimentations

Dans cette analyse de robustesse, nous employons la base de données N-CARS [Sir+18]. Après l'entraînement du modèle de référence en utilisant l'ensemble d'apprentissage, l'ensemble de validation est altéré par une corruption spécifiée. Inspiré par [HD19], on définit 5 niveaux croissants de sévérité pour une corruption donnée, où chaque niveau de sévérité augmente l'intensité de la corruption.

Deux corruptions communes sont employées :

- **Bruit d'activité de fond** (*"background activity noise"*, en anglais). Cette corruption apparaît à cause du bruit thermique et du courant de fuite de jonction dans le capteur [Gal+20]. Il en résulte la production d'événements parasites, qui ne sont liés à aucun changement d'intensité dans la scène [Fen+20]. Pour simuler ce bruit, un certain pourcentage d'événements est ajouté selon une distribution gaussienne dans le flux en entrée. Ce pourcentage varie en fonction du niveau de sévérité utilisé. Il faut noter que le bruit d'activité de fond est déjà observé nativement dans une caméra événementielle, c'est pourquoi cette corruption est destinée à renforcer ce phénomène et non à l'introduire.

- **Occultation** (*"occlusion"*, en anglais). Cette corruption simule la présence d'un objet qui occulte une partie de la scène. Ici, cette simulation est faite en supprimant les événements dans une case située au centre de la caméra. Le niveau de

Corruption	Niveau de sévérité				
	1	2	3	4	5
Bruit d'activité de fond	0.5%	0.8%	1.0%	2.0%	3.0%
Occultation	35%	45%	50%	60%	70%

Table 3.5: Valeurs des paramètres des corruptions étudiées (bruit d'activité de fond et l'occultation) **pour chaque niveau de sévérité**.

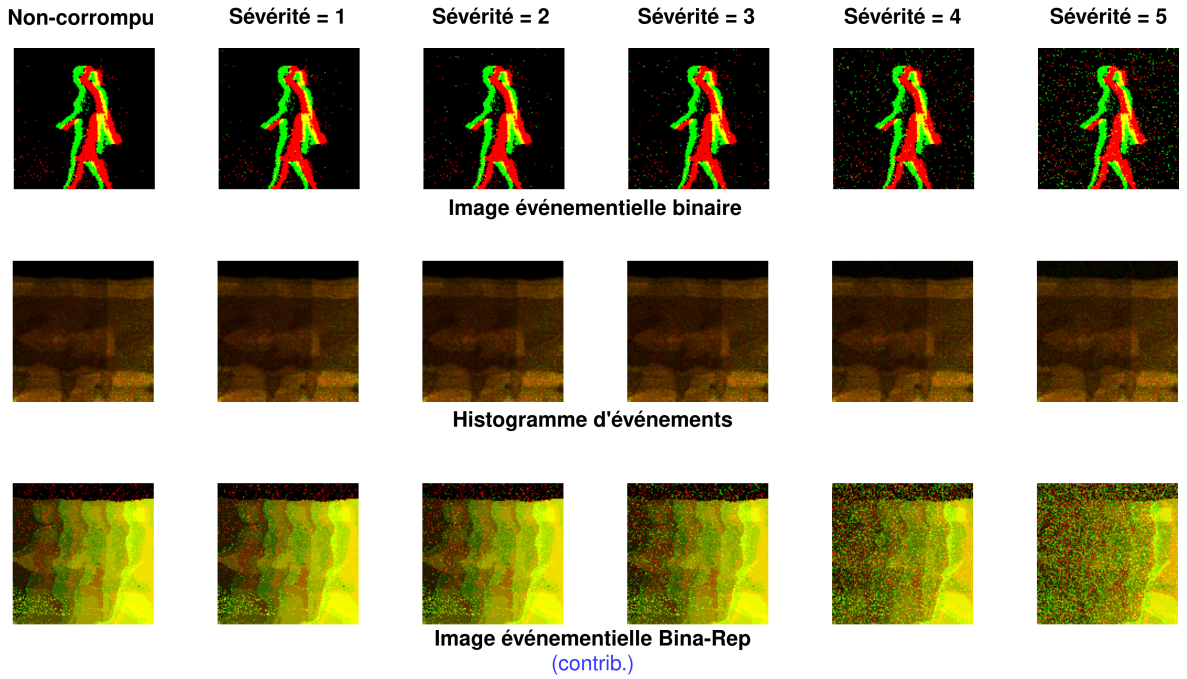


Figure 3.4: Impact du bruit d'activité de fond pour chaque niveau de sévérité.

sévérité fait varier le pourcentage de surface de l'image occultée.

La Table 3.5 détaille les valeurs des paramètres influencés par le niveau de sévérité pour les deux corruptions expliquées. Les Figures 3.4 et 3.5 illustrent un exemple du bruit d'activité de fond et de l'occultation, respectivement, pour chaque niveau de sévérité.

Pour estimer quantitativement la robustesse d'un modèle avec une représentation d'événements donnée, on emploie un score de **baisse de précision relative** (*"Relative Accuracy Drop"*, en anglais) qui mesure la perte de précision relative du modèle sur des données corrompues par rapport à la précision obtenue sur les données non-altérées. L'Annexe A.1 détaille et formule le calcul de cette métrique (équation A.1). Finalement, on reporte l'évolution du score de baisse de précision relative en fonction du niveau de sévérité (de 1 à 5).

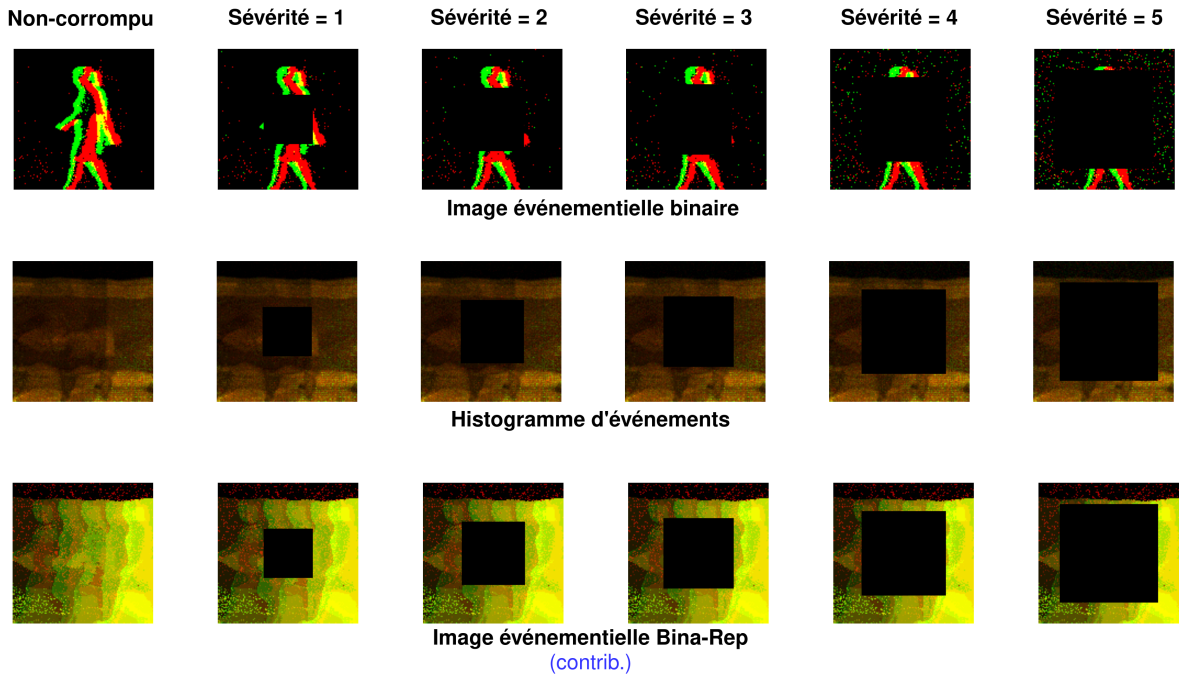
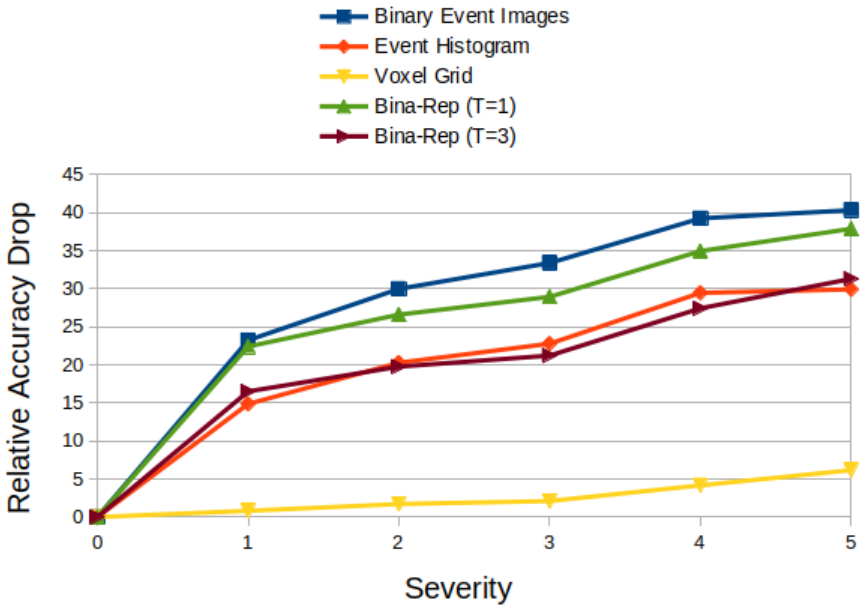


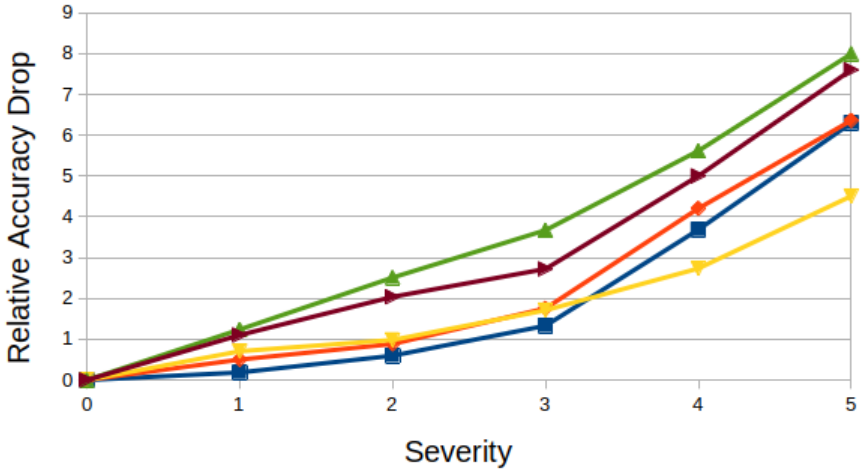
Figure 3.5: Impact de l'occultation pour chaque niveau de sévérité.

3.3.4.2 Résultats

Bruit d'Activité de Fond. Les résultats de robustesse face au bruit d'activité de fond sont montrés en Figure 3.6a. Nous observons la même évolution et des scores similaires pour toutes les méthodes, à l'exception de Voxel Grid [Zhu+19] qui est extrêmement robuste contre ce type de corruption. D'autre part, Bina-Rep présente de meilleures performances que les images événementielles binaires dans les paramètres $T = 1$ et se comporte de manière similaire à l'histogramme d'événements [Maq+18] avec $T = 3$, ce qui suggère une plus grande robustesse en augmentant le nombre d'images Bina-Rep. Cela peut être expliqué par le fait qu'une image Bina-Rep exprime l'ordre d'arrivée des événements en plus de leur occurrence (comme expliqué dans la Section 3.1). Par conséquent, une seule image Bina-Rep sature en raison des événements de fond, ce qui nuit aux performances. Ce phénomène de saturation est facilement observé sur la Figure 3.4. En utilisant une séquence au lieu d'une seule image, cette influence est réduite et nous obtenons des performances similaires à celles de l'histogramme d'événements qui est moins susceptible de saturer face à une augmentation du nombre d'événements. D'autre part, on remarque que Bina-Rep est plus robuste aux corruptions que des séquences d'images binaires, soulignant le fait que la ré-interprétation de séquences d'images binaires en images Bina-Rep permet de limiter l'impact du bruit d'activité de fond sans ajouter de calculs supplémentaires.



(a) Bruit d'Activité de Fond



(b) Occultation

Figure 3.6: Scores de baisse de précision relative ("*Relative Accuracy Drop*") **pour chaque représentation d'événements étudiée**.

Occultation. Les résultats de robustesse face à l'occultation sont présentés en Figure 3.6b. Toutes les méthodes semblent être assez robustes face à ce type de corruption (les scores de baisse de précision relative restent inférieurs à 9%). La même évolution est observée également. Voxel Grid montre toujours une grande robustesse pour des niveaux de sensibilité élevés, mais suit la même tendance que les autres représentations. Bina-Rep présente des scores légèrement plus faibles quelque soit le nombre d'images utilisées ($T = 1$ ou $T = 3$), mais reste compétitif par rapport aux autres méthodes de représentation d'événements.

Observations Générales. Bien que Bina-Rep ne montre pas de robustesse supérieure aux autres représentations étudiées, il est encourageant de constater des résultats compétitifs. De plus, les expérimentations soulignent des bons résultats par rapport au bruit d'activité de fond, ce qui révèle l'intérêt de la méthode contre ce type de corruption. Nos expérimentations montrent aussi l'intérêt d'utiliser des images Bina-Rep en séquences et non seules pour améliorer la robustesse, car une seule image a tendance à saturer à cause de l'ajout des événements bruités. En dehors de Bina-Rep, nos expérimentations permettent de souligner la grande robustesse des représentations basées sur les Voxels (et donc basées sur un espace en trois dimensions). En effet, nous observons de très faibles scores de baisse de précision relative pour Voxel Grid, ce qui n'est pas explicitement suspecté dans l'article original de cette méthode [Zhu+19].

3.4 Conclusion

Dans ce chapitre, nous avons traité la problématique de la représentation des événements dans le contexte du traitement par des algorithmes de vision artificielle conventionnelle. Plus précisément, nous avons introduit Bina-Rep, une nouvelle méthode de représentation en images événementielles qui se distingue des autres méthodes similaires en encodant une forme d'information temporelle (c'est-à-dire l'ordre d'arrivée des événements par pixel). Le mécanisme de Bina-Rep repose sur une séquence de T images événementielles binaires qui est réinterprétée en une représentation à T bits. Ainsi, notre approche ne nécessite pas de calculs supplémentaires et constitue principalement une réinterprétation des mêmes données.

À travers notre étude expérimentale menée sur des tâches de classification, nous avons démontré que Bina-Rep obtient des résultats compétitifs, notamment pour la représentation d'informations spatio-temporelles. De plus, notre étude sur la robustesse face aux corruptions courantes des capteurs événementiels a révélé que Bina-Rep atténue efficacement l'impact du bruit d'activité de fond, en particulier dans des configurations impliquant plusieurs images.

Néanmoins, on observe une difficulté liée aux hyperparamètres en observant les résultats les plus faibles obtenus avec notre approche. La construction d'une image Bina-Rep implique de déterminer la valeur d'un hyperparamètre T lié à la représentation en nombre à T-bits. Trouver une valeur appropriée pour T en fonction du flux d'événements à traiter n'est pas une tâche facile. Si T est trop petit, on observe la saturation de l'image événementielle, ce qui rend les motifs exprimés par la méthode inefficaces. D'un autre côté, si T est trop grand, cela augmente considérablement les besoins en mémoire de la méthode. Une bonne valeur de T est notamment influencée par le nombre d'événements par pixel dans un flux donné. De plus, si les images Bina-Rep sont traitées en séquence, on ajoute l'hyperparamètre K pour le nombre d'images dans la séquence, ainsi que la problématique de l'échantillonnage des images événementielles (décrite en Section 2.3.3.1).

Enfin, les résultats prometteurs de Bina-Rep démontrent la possibilité de créer des images événementielles qui encodent l'information temporelle tout en étant plus compacte que les séquences d'images binaires. Cependant, ces gains se limitent à l'étape de représentation d'événements, sans impacter l'algorithme de vision subséquent qui est souvent la partie la plus coûteuse en termes de calcul/mémoire. Ainsi, l'utilisation d'un algorithme de vision conventionnel avec des images événementielles ne permet

pas de tirer pleinement parti de l'efficacité énergétique des caméras événementielles ou des technologies neuromorphiques en générale. Une meilleure approche consisterait à développer directement un algorithme de vision adapté pour traiter le flux d'événements épars et asynchrones de la caméra. Les SNNs se révèlent prometteurs pour créer de tels algorithmes capables de gérer ces caractéristiques spécifiques des caméras événementielles.

Dans notre prochain chapitre, nous nous concentrerons sur la partie d'algorithme de vision, en développant et analysant des modèles basés sur les SNNs afin de créer des systèmes neuromorphiques hautement efficaces notamment en termes de consommation énergétique.

4

Développement et Analyses de Réseaux de Neurones Impulsionnels Profonds pour la Vision Artificielle

Sommaire

Les réseaux de neurones impulsionnels (SNNs) émergent comme une alternative prometteuse aux réseaux de neurones artificiels (ANNs) pour la vision artificielle à faible consommation énergétique. Les récentes avancées des règles d'apprentissage pour les SNNs, notamment l'apprentissage supervisé par substitut du gradient (ou "surrogate gradient learning" en anglais), ont permis le développement de réseaux profonds performants. Cependant, malgré ces progrès, il reste encore des questions en suspens quant à la capacité des SNNs convolutifs (ou CSNNs) profonds à traiter des données impulsionnelles complexes, telles que des flux d'événements ou des images converties via un schéma de codage neuronal.

Dans ce chapitre, nous nous intéressons à la conception de CSNNs profonds entraînés de manière supervisée en utilisant le substitut du gradient, afin d'explorer différentes tâches de vision, telles que la localisation d'objet et la reconnaissance d'expressions faciales événementielle. De plus, nous profitons de ces nouvelles applications pour analyser le comportement de ces CSNNs profonds supervisés, par exemple en examinant leur robustesse face aux corruptions des capteurs et l'influence du codage neuronal d'une image statique.

Ce chapitre est organisé de la manière suivante : premièrement, nous établissons une preuve de concept de l'entraînement d'un CSNN profond pour la localisation d'objet sur des images statiques, afin de vérifier l'applicabilité des technologies neuromorphiques considérées. Deuxièmement, sur base des observations faites lors de la preuve de concept, nous formalisons une architecture générique basée sur un CSNN, afin de traiter différentes tâches de vision basées sur des images ou des événements de manière polyvalente. En troisième lieu, nous employons ce modèle de référence pour étudier la capacité des CSNNs à traiter de l'information spatiale en résolvant une problématique de localisation d'objet sur des images ou des flux d'événements. Ce contexte particulier est mis à profit pour étudier divers aspects des CSNNs optimisés par substitut du gradient tels que la robustesse et le codage neuronal des images. Enfin, nous introduisons "*Spiking-Fer*", une adaptation du modèle générique pour réaliser la reconnaissance d'expressions faciales ("*Facial Expression Recognition*" ou FER, en anglais) à l'aide de capteurs événementiels.

4.1 Preuve de Concept de Réseaux de Neurones Impulsionnels Profonds Supervisés

4.1.1 Objectifs

Dans cette section, nous proposons une preuve de concept en concevant une architecture CSNN profonde pour la localisation d'objet sur des images naturelles complexes, sous la forme d'images en niveaux de gris. Cette preuve de concept valide la possibilité de nos études ultérieures dans ce chapitre, qui visent à étendre puis analyser ce paradigme d'apprentissage vis-à-vis de nouvelles tâches, y compris d'autres types de prédictions et/ou des bases de données plus complexes.

4.1.2 Formulation de la Localisation d'Objet

Nous formulons la tâche de localisation d'objet par un CSNN de la manière suivante : soit un flux d'impulsions en entrée obtenu à partir d'une caméra événementielle ou d'une image statique encodée via un schéma de codage neuronal, l'objectif est de prédire les coordonnées de la boîte englobante $\mathbf{B} = \{x_{min}, y_{min}, x_{max}, y_{max}\}$, où (x_{min}, y_{min}) et (x_{max}, y_{max}) représentent respectivement les coins supérieur gauche et inférieur droit de la boîte englobante.

Pour ce faire, nous concevons une architecture basée sur un CSNN $Localizer(\cdot)$ tel que :

$$Localizer(\mathbf{X}_T) = \{x_{min}, y_{min}, x_{max}, y_{max}\} = \mathbf{B} \tag{4.1}$$

où $\mathbf{X}_T \in \mathbb{B}^{T \times C \times H \times W} = \{X_t\}_{t=1}^{T}$ est le tenseur impulsionnel représentant les impulsions d'entrée discrétisées en T étapes temporelles (voir Section 2.3.3.1), obtenues à partir d'un capteur basé sur des images ou des événements avec une résolution $(H \times W)$ et C canaux (par exemple, $C = 3$ pour une image RGB, $C = 2$ pour un flux d'événements, ...).

Pour évaluer les performances d'un algorithme visant cette tâche de localisation d'objet, on emploie l'intersection sur l'union moyenne (*"mean intersection over union"* ou *"mIoU"*), décrite en Annexe A.2.

4.1.3 Méthode : Modèle pour la Preuve de Concept

Cette section est dédiée à la conception de la fonction CSNN désignée par $Localizer(\cdot)$ pour localiser strictement un objet dans le tenseur d'impulsions $\mathbf{X}_T$.

4.1.3.1 Codage Neuronal Fréquentiel d'Images Statiques

Dans le cadre de cette preuve de concept, le tenseur impulsionnel $\mathbf{X}_T$ en entrée du CSNN est obtenu à partir d'une image statique en niveaux de gris de dimension $(1 \times H \times W)$ (donc, $C = 1$ dans le tenseur impulsionnel $\mathbf{X}_T$). Comme décrit dans la Section 2.2.3, un CSNN traite l'information sous forme de trains d'impulsions binaires et n'est donc pas adapté pour des valeurs réelles d'intensité de pixels. Par conséquent, une fonction de pré-traitement, nommée "**codage neuronal**", est nécessaire.

Dans le cadre de ce travail, nous utilisons le **codage neuronal fréquentiel** (*"rate coding"*, en anglais) [Bre15] pour convertir la valeur d'intensité d'un pixel donné en une séquence de T impulsions binaires qui représentent une fréquence. La formulation de ce schéma de codage est donnée en Section 2.2.3.1.

4.1.3.2 Règle d'Apprentissage Supervisé

La règle d'apprentissage utilisée est présentée dans [KMN20], et est nommée "**De**ep **Co**ntinuous **L**ocal **Le**arning" (**DECOLLE**), ce qui se traduit en français par "Apprentissage local continu profond". Il s'agit d'un cadre d'apprentissage dans lequel un SNN est optimisé via une fonction d'erreur par couche de neurones pour un apprentissage supervisé par calcul de gradient local. Dans cette section, nous passons en revue les informations importantes à connaître par rapport à cette méthode (notamment la structure du réseau nécessaire), mais ne détaillons pas de manière exhaustive son fonctionnement. Pour plus de détails, veuillez consulter la publication originale [KMN20].

La Figure 4.1 illustre un schéma explicatif de la règle d'apprentissage. DECOLLE utilise le substitut du gradient (expliqué en Section 2.2.5.2) afin de réaliser un apprentissage supervisé basé sur une erreur calculée par une fonction de coût.

Contrairement à l'apprentissage (plus populaire) décrit en Section 2.2.5 basé sur la rétropropagation à travers le temps (BPTT), DECOLLE est une règle *locale*, c'est-à-dire que les informations employées pour la mise à jour d'un poids synaptique sont locales à celui-ci [BS16] (voir Section 2.2.4). Pour ce faire, chaque couche du SNN est connectée à une couche locale de **"lecture"** (*"readout"*, en anglais) via des connections

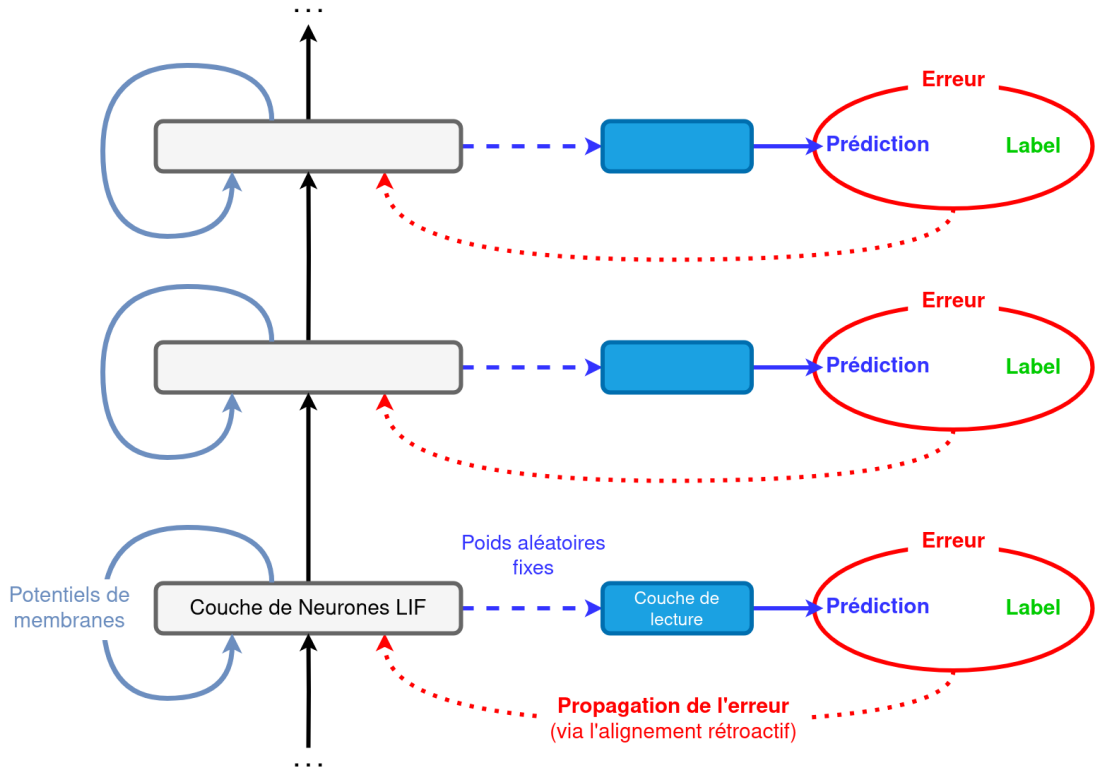


Figure 4.1: Schéma simplifié du fonctionnement de la règle d'apprentissage DECOLLE sur un SNN à plusieurs couches. Chaque couche de neurones LIF est connectée à une couche de lecture via des poids aléatoires fixes. La couche de lecture génère une prédiction, permettant de propager l'erreur de la fonction de coût à la couche de neurones. Les poids synaptiques de la couche de neurones LIF sont alors optimisés grâce à l'alignement rétroactif [Nøk16].

à poids aléatoires fixes [Nøk16; Lil+16]. L'erreur entre la prédiction fournie par cette couche de lecture est propagée à travers les connexions aléatoires pour permettre l'entraînement de la couche de neurones impulsionnels via le substitut du gradient. Cette propagation de l'erreur est limitée à la couche de neurones impulsionnels liée localement grâce à l'emploi de **l'alignement rétroactif** (*"feedback alignement"*, en anglais) [Nøk16]. En optimisant la même fonction de coût pour toutes les couches du réseau, DECOLLE suppose que le SNN apprend indirectement à extraire des caractéristiques hiérarchiques utiles qui permettent aux couches les plus profondes de minimiser l'erreur locale.

D'autre part, DECOLLE est une règle d'apprentissage **en ligne** (*"online"*, en anglais), ce qui signifie que le processus de propagation de l'erreur et l'apprentissage est réalisé à chaque étape temporelle. De cette manière, DECOLLE ne requiert pas de mémoire supplémentaire pour retenir les états précédents du SNN sur toute la durée de l'inférence, comme il est nécessaire de le faire avec la BPTT.

Grâce à ces deux caractéristiques (*locale* et *en ligne*), DECOLLE est une règle d'apprentissage dont l'implémentation sur matériel neuromorphique est possible [Chi+14; KMN20]. Néanmoins, il faut préciser que cela force le SNN à faire une prédiction à chaque étape temporelle, et pas seulement à la fin du traitement du tenseur impulsionnel en entrée. Cela est susceptible de poser des difficultés, notamment pour définir la prédiction finale adéquate parmi celles émises par le SNN durant toute l'inférence.

4.1.3.3 Architecture du Réseau de Neurones Impulsionnels

Le CSNN de cette preuve de concept est composé de neurones LIF (formulés en Section 2.2.2) optimisés par DECOLLE. Nous employons une architecture profonde de type "**encodeur-décodeur**" [YS19]. Cette architecture est populaire pour la conception de ANNs et est utilisée dans de nombreux contextes en vision artificielle (segmentation sémantique [RFB15; BKC17], génération d'images [SSK20], détection d'objets [Lin+17], ...). Elle est composée de deux éléments principaux, l'encodeur et le décodeur, connectés par des connexions résiduelles :

- **L'encodeur** est un réseau de neurones qui prend une image en entrée (ou dans notre cas, un tenseur impulsionnel) et la traite à travers une série de couches de convolutions pour extraire des caractéristiques de haut niveau tout en réduisant les dimensions spatiales. Ce processus s'appelle "encodage" car il transforme l'entrée en une représentation compacte dans un espace de caractéristiques de dimension inférieure [Mas+11]. Cependant, en extrayant ces caractéristiques de haut niveau, la représentation obtenue a tendance à perdre l'information spatiale de l'entrée d'originale.

- **Le décodeur** prend la représentation encodée provenant de l'encodeur et la remet à l'échelle pour obtenir la résolution originale de l'entrée tout en affinant progressivement les caractéristiques. Il peut réaliser cela grâce à une série de couches diverses ("upsampling" [BKC17], convolutions transposées [LSD15], ...). Le décodeur reconstitue efficacement la résolution de l'image originale à partir de la représentation compacte, et ce processus est appelé "décodage".

- Pour retrouver les informations spatiales fines perdues lors de l'encodage, des **connexions résiduelles** [He+16; BKC17; Hua+17] relient l'encodeur et le décodeur en fournissant un chemin direct pour les informations spatiales de bas niveau provenant de l'encodeur vers les couches du décodeur. Cela permet au décodeur d'exploiter les caractéristiques des couches basses de l'encodeur pour reconstruire

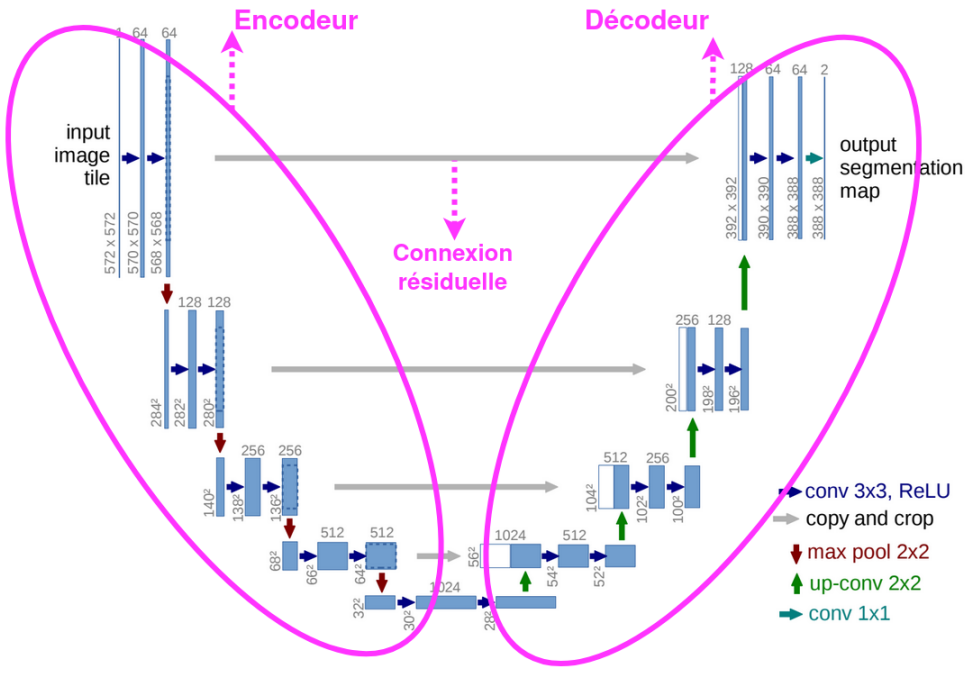


Figure 4.2: Schéma annoté de UNet [RFB15], une architecture encodeur-décodeur typique. Nos annotations explicatives de l'architecture encodeur-décodeur sont en magenta.

des cartes de caractéristiques (*"feature maps"*, en anglais) de haute résolution tout en conservant la haute valeur sémantique obtenue par les couches profondes de l'encodeur.

La Figure 4.2 illustre un exemple typique d'architecture encodeur-décodeur, nommé "UNet" [RFB15].

Concrètement, notre application du paradigme d'architecture encodeur-décodeur est réalisé en un CSNN à 6 couches de convolution (illustré en Figure 4.3). L'encodeur est composé d'une suite de 3 couches de convolution entrecoupées par des couches de "Max pooling" [KSH12] 2×2 où chacune divise la résolution des cartes de caractéristiques par 2. Le décodeur est composé de 3 couches de convolutions entrecoupées par des couches de "Upsampling" [Lin+17] 2×2 qui multiplient la résolution des caractéristiques par 2. Par conséquent, la carte de caractéristiques en sortie du modèle possède la même résolution spatiale que l'entrée.

Comme montré en bas à droite de la Figure 4.3, les connexions résiduelles employées dans notre modèle consistent en des opérations d'addition entre les caractéristiques

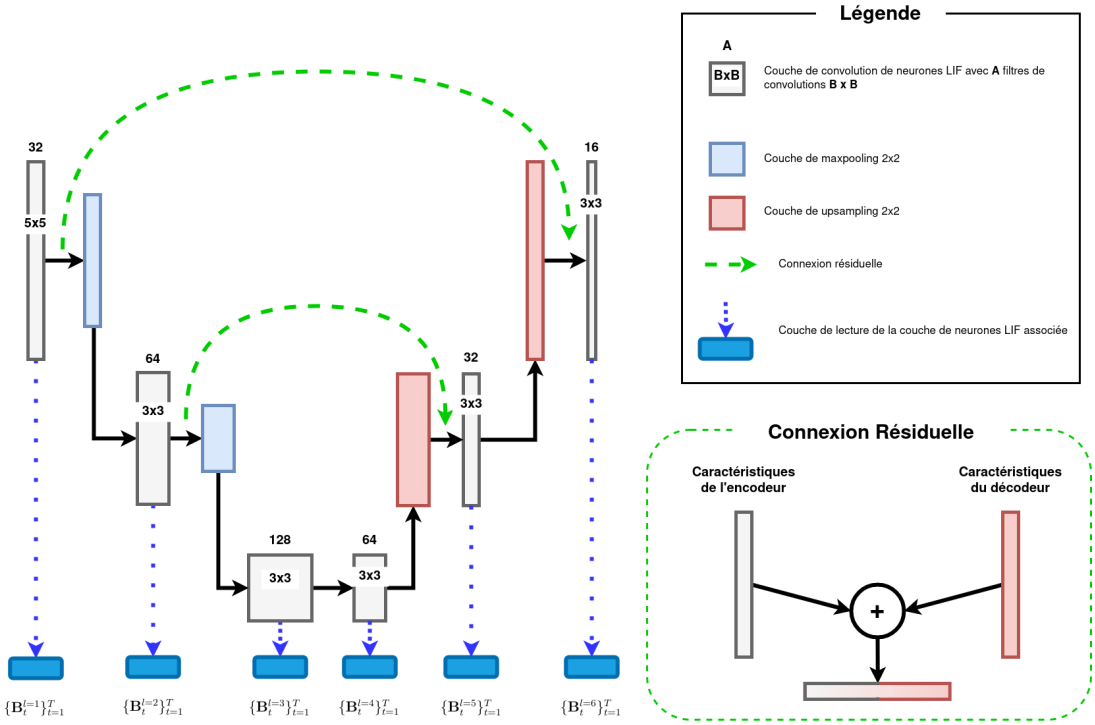


Figure 4.3: Architecture du CSNN conçu pour notre preuve de concept. L'architecture est de type "encodeur-décodeur", et chaque couche de neurones impulsionnels est connectée à une couche de lecture. Les couches de lecture génèrent des prédictions de boîte englobante à chacune des T étapes temporelles ($\{\mathbf{B}_t^l\}_{t=1}^T$).

de l'encodeur (obtenues en sortie d'une couche de convolution) et les caractéristiques du décodeur (obtenues en sortie d'une couche de Upsampling). Comme la règle d'apprentissage utilisée est DECOLLE, une couche de lecture à connexions aléatoires est attachée à chaque couche de convolution à neurones LIF.

Dans le cadre de cette preuve de concept, l'utilisation du paradigme de l'encodeur-décodeur est supposée présenter deux avantages. Premièrement, cela permet un meilleur traitement de l'information spatiale, utile pour la localisation d'objet, par rapport à un réseau CNN classique [Lin+17]. Deuxièmement, cela permet de tester la faisabilité de l'emploi d'une architecture de réseau de neurones différente de l'encodeur classique [Fan+21a] avec des neurones impulsionnels.

Pour finir, nous désignons un nombre d'étapes temporelles important, à savoir $T = 1000$, afin de garantir une simulation précise de la dynamique des neurones LIF.

4.1.3.4 Calcul de la Prédiction de Localisation

Problématique liée à DECOLLE. Comme expliqué en Section 4.1.2, la tâche de localisation d'objet visée consiste à obtenir une prédiction de boîte englobante pour l'ensemble du flux d'impulsions en entrée. Cependant, avec DECOLLE, chaque couche de lecture associée à une couche de neurones impulsionnels produit une prédiction à chacune des T étapes temporelles. Si l'on considère un CSNN optimisé par DECOLLE composé de L couches, une inférence générera alors $L \times T$ prédictions distinctes (dans l'architecture décrite en Section 4.1.3.3, $L = 6$ et $T = 1000$). Par conséquent, la problématique liée à ce fonctionnement consiste à identifier la meilleure prédiction parmi les $L \times T = 6000$ générées.

Solution Naïve. La méthode choisie pour calculer la prédiction finale du modèle proposé consiste à prendre la **prédiction de la dernière couche de lecture** (la couche la plus profonde) **émise à la dernière étape temporelle**. Ce choix est justifié par le fait que **(1)** la dernière couche (faisant partie du décodeur) est supposée avoir une haute valeur sémantique et conserver les informations spatiales fines de l'encodeur grâce aux connexions résiduelles [Lin+17]; et **(2)** on assure que la totalité des impulsions en entrée a été traitée par le CSNN à la dernière étape temporelle. Il faut cependant mentionner que d'autres stratégies peuvent être appliquées, telles que la moyenne ou la médiane de l'ensemble des prédictions, etc.

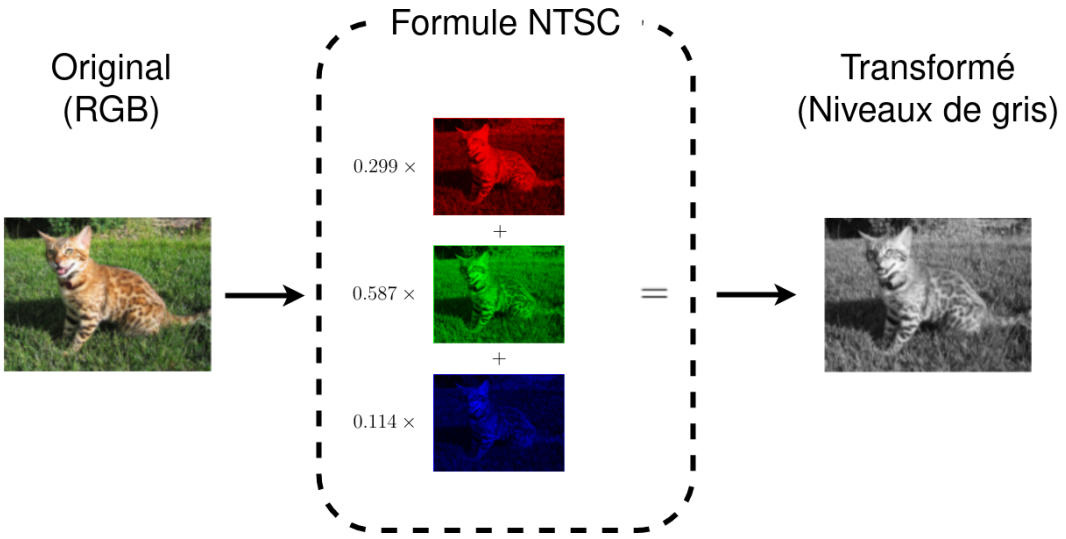
Formulation de la Solution. Pour un CSNN de L couches de neurones LIF entrainés par DECOLLE et fonctionnant sur T étapes temporelles, on désigne par $\mathbf{B}_t^l$ la prédiction de boite englobante à l'étape temporelle $1 \leq t \leq T$ de la couche de lecture associée à la couche de neurones LIF $1 \leq l \leq L$. Cette notation est illustrée en bas à gauche de la Figure 4.3. La prédiction retenue dans notre modèle est donc désignée par $\mathbf{B}_T^L$.

4.1.4 Expérimentations : Validation de la Preuve de Concept

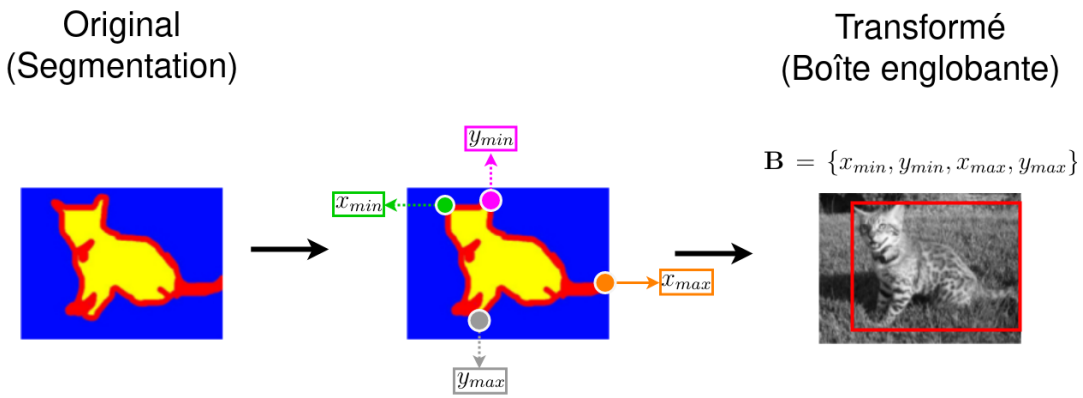
Dans cette section, nous présentons la validation expérimentale de notre preuve de concept basée sur un CSNN de type encodeur-décodeur.

4.1.4.1 Base de Données d'Images Statiques

La base de données employée est Oxford-IIIT-Pet [Par+12]. Elle est constituée d'images RGB contenant strictement un chien ou un chat. Pour chaque image, les annotations



(a) Conversion de l'image couleur RGB originale en niveaux de gris via la formule NTSC [Bro54] utilisée par défaut dans la librairie OpenCV [Braoo].



(b) Adaptation du label (un masque de segmentation) **pour récupérer les coordonnées de la boîte englobante** de l'objet d'intérêt.

Figure 4.4: Adaptation de la base de données Oxford-IIIT-Pet [Par+12] pour la tâche de localisation visée.

fournies sont **(1)** la classe de l'animal représenté (chien ou chat) ; et **(2)** le masque de segmentation délimitant l'animal par rapport au reste de l'image. Oxford-IIIT-Pet est composée de 7349 images RGB. Dans ce travail, nous définissons un ensemble d'apprentissage de 6000 échantillons sélectionnés aléatoirement, et le reste (1349 images) constitue l'ensemble de validation.

Pour adapter les échantillons de la base de données Oxford-IIIT-Pet [Par+12] à la tâche de localisation d'objet de cette preuve de concept, deux étapes de conversion de la base de données sont nécessaires, comme résumé dans la Figure 4.4 :

- **Modification de l'entrée :** les images sont converties du format couleur RGB au format en niveaux de gris en utilisant la formule NTSC [Bro54] (méthode par défaut utilisée sur OpenCV [Braoo]). La valeur en niveaux de gris d'un pixel est donnée par : Luminance $= 0.299 \times$ Rouge $+ 0.587 \times$ Vert $+ 0.114 \times$ Bleu.

- **Modification de la prédiction :** le masque de segmentation délimitant l'animal représenté permet de définir les coordonnées de la boîte englobante **B**. Les coordonnées (x_{min}, y_{min}) (resp. (x_{max}, y_{max})) correspondent aux coordonnées en x et y minimales (resp. maximales) du masque.

Dans le cadre de notre preuve de concept, l'utilisation d'Oxford-IIIT-Pet [Par+12] est justifiée par deux raisons. Premièrement, les images de cette base de données ont été capturées dans des scènes naturelles, ce qui la rend plus difficile et représentative de cas d'utilisation réels par rapport aux bases de données simulées [Orc+15; Li+17] couramment utilisées dans les travaux précédents portant sur les SNNs. L'utilisation d'images naturelles est cruciale dans notre preuve de concept car elle reflète mieux la complexité et la variabilité rencontrées dans les applications du monde réel. Deuxièmement, la problématique visée (c'est-à-dire, la localisation d'objet) reste relativement simple par rapport aux tâches populaires de l'état de l'art (comme la détection d'objets [Zou+23b], etc.), ce qui est approprié pour vérifier la possibilité de traiter de l'information spatiale avec des SNNs profonds supervisés.

4.1.4.2 Détails d'Implémentation

Nous développons notre architecture CSNN en adaptant l'implémentation originale de DECOLLE [KMN20][1] réalisée avec PyTorch [Pas+19]. Notre preuve de concept est réalisée sur 100 époques sur un ordinateur équipé d'un GPU NVIDIA 2080Ti. La taille du lot (*"batch size"*, en anglais) est fixée à 16. Nous employons l'optimiseur AdaMax [KB14] avec $\beta_1 = 0$, $\beta_2 = 0.95$, et une fonction de coût L1 lissée (`"SmoothL1Loss"`). Le taux d'apprentissage est initialisé à 10^{-9}. Les images sont redimensionnées en $(H, W) = (176, 240)$ pixels. Durant la phase d'apprentissage, nous appliquons aléatoirement des augmentations de données, comprenant des retournements horizontaux aléatoires et des variations aléatoires de luminosité.

(a) Prédictions.

(b) Vérité terrain.

Figure 4.5: Exemples des résultats obtenus par la preuve de concept proposée.

4.1.4.3 Résultats de la Preuve de Concept

Notre expérience montre un résultat prometteur de **63.2 % de mIoU** sur l'ensemble de validation. De plus, nous vérifions visuellement la pertinence des boîtes englobantes prédites par notre CSNN. La Figure 4.5 présente des exemples représentatifs des prédictions faites sur l'ensemble de validation. Les boîtes englobantes sont généralement prédites avec une bonne précision (entre 55% et 70% de mIoU). Cependant, des exemples plus rares entraînent des performances bien moindres (inférieures à 15% de mIoU). Nous constatons que ce problème est dû à des images où l'objet d'intérêt est petit et/ou situé en dehors de l'avant-plan de la scène. Pour expliquer ce phénomène, nous émettons deux hypothèses :

1. **Déséquilibre des données :** notre méthode est sensible au déséquilibre des données présentes dans Oxford-IIIT-Pet concernant les boîtes englobantes. En effet, la majorité des images présentent l'objet d'intérêt en gros plan. Par conséquent, notre méthode généraliserait sur ces exemples et ne prendrait pas en compte les exemples rares des animaux en arrière-plan.

2. **Calcul de la prédiction :** La solution naïve de calcul de la prédiction (définie en Section 4.1.3.4) n'est pas considérée comme optimale et pourrait entraîner une perte d'information cruciale.

[1]Implémentation originale : https://github.com/nmi-lab/decolle-public

4.1.5 Limitations du Modèle

Bien que la preuve de concept montre des résultats prometteurs qui valident l'utilisation de CSNNs profonds supervisés pour la suite de ce chapitre, nous estimons que certains choix importants dans la conception du modèle doivent être modifiés pour améliorer les performances et l'efficacité attendues.

Tout d'abord, l'utilisation de neurones LIF plutôt que de neurones IF n'est pas particulièrement justifiée dans le cas que nous considérons, à savoir la localisation d'objet sur des images statiques. En effet, une étude récente [Bou+22] montre que la dynamique de fuite du potentiel de membrane du neurone LIF n'apporte pas nécessairement un meilleur traitement de l'information par rapport au neurone IF, notamment lorsque la topologie du réseau de neurones n'est pas récurrente. Par conséquent, l'utilisation de neurones IF permettrait d'éviter les calculs supplémentaires associés au mécanisme de fuite, tout en maintenant de bonnes performances.

Deuxièmement, la règle d'apprentissage DECOLLE n'est pas justifiée pour notre tâche de localisation d'objet. La première raison découle de la problématique consistant à choisir la prédiction optimale parmi les $T \times L = 6000$ prédictions des couches de lecture (voir Section 4.1.3.4). De plus, la propriété d'apprentissage "en ligne" de DECOLLE est contre-intuitive dans la problématique visée, car chaque étape temporelle implique une modification des poids du CSNN. Bien que l'apprentissage en ligne soit une fonctionnalité excellente pour d'autres tâches de vision (par exemple, l'apprentissage continu [Mai+22]), cela n'est pas nécessaire dans le cas étudié et pourrait probablement entraîner un phénomène d'oubli catastrophique [Kir+17].

Malgré les hypothèses et les améliorations suggérées pour notre preuve de concept (notamment en Section 4.1.4.3), les inconvénients découlant des choix de conception sous-optimaux pour le CSNN nous incitent à effectuer des modifications substantielles au cœur de la méthode plutôt que de résoudre les problèmes identifiés. Nous avons l'intention d'adopter une règle d'apprentissage plus directe, en particulier la BPTT avec le substitut du gradient [NMZ19], plutôt que d'utiliser une variante d'apprentissage en ligne telle que DECOLLE. De plus, nous puisons dans des recherches antérieures [Bou+22] et choisissons d'utiliser des neurones IF au lieu de LIF afin de réduire la charge de calcul et l'impact de la fuite du potentiel de membrane.

En conclusion finale, cette preuve de concept a validé l'applicabilité des CSNNs profonds supervisés sur des problèmes de vision au-delà des bases de données tradi-

tionnelles (par exemple, N-MNIST [Orc+15]). De plus, elle nous a permis d'acquérir des enseignements précieux sur certains choix de conception de ces CSNNs profonds. Dans la suite de ce chapitre, nous appliquons ces enseignements pour élaborer un modèle CSNN générique, c'est-à-dire adaptable à diverses tâches de vision et divers types d'entrées (images statiques ou flux d'événements).

4.2 Modèle Générique de Réseau de Neurones Impulsionnels Convolutif

Dans cette section, nous décrivons le modèle générique basé sur un CSNN qui sera utilisé pour le reste de ce chapitre. Ce modèle tient compte des observations faites lors de la preuve de concept réalisée en Section 4.1.

4.2.1 Objectifs du Modèle

La conception du modèle CSNN proposé doit répondre à plusieurs critères, d'une part pour garantir son utilisation dans les problématiques étudiées ultérieurement dans ce chapitre, et d'autre part pour assurer des analyses fiables et équitables du fonctionnement d'un CSNN pour des tâches de vision. Nous examinerons en détail ces critères dans la suite de cette section.

Polyvalence de l'entrée. Le modèle CSNN doit être capable d'être entraîné efficacement sur diverses modalités, notamment des flux d'événements de caméras événementielles ou différents codages neuronaux pour des images statiques. Pour atteindre cet objectif, nous adoptons l'apprentissage par BPTT et substitut du gradient (détaillé en Section 2.2.5), car cette approche s'est avérée polyvalente pour entraîner des SNNs avec des types d'entrées variés (par exemple, des flux d'événements [Fan+21a; Zou+23a; Zho+22], des images par codage fréquentiel [KCP21], etc.).

Polyvalence de prédiction. Le modèle proposé doit être adaptable à diverses tâches de vision, notamment des problèmes de classification (reconnaissance d'expressions faciales) et de régression (localisation d'objet). Cependant, obtenir des prédictions précises avec des neurones impulsionnels peut être difficile, comme illustré dans la Section 4.1.3.4 de notre preuve de concept. Pour résoudre ce problème, nous suggérons d'utiliser un CSNN en tant qu'encodeur convolutif pour extraire un vecteur de caractéristiques à partir des données d'entrée. Ce vecteur de caractéristiques est ensuite transmis à une couche dense linéaire chargée de réaliser la prédiction finale. En adoptant cette approche, nous pouvons évaluer directement la capacité du SNN à extraire des caractéristiques significatives à partir des données visuelles, ce qui correspond à notre objectif d'étudier le comportement des SNNs supervisés. De plus, en confiant le calcul des prédictions à une partie distincte du modèle, nous pouvons nous concentrer sur l'analyse directe des capacités d'extraction de caractéristiques

du SNN, sans introduire d'étape intermédiaire.

Simplicité de l'architecture. Récemment, de nombreux travaux [Zha+22b; Yao+21; Zha+23; Yao+23] portés sur les SNNs exploitent l'efficacité de l'apprentissage supervisé par substitut du gradient pour concevoir des modules sophistiqués visant à améliorer les performances (comme des modules d'attention spatio-temporelle [Yao+21; Zha+22b], des hyperparamètres additionnels des neurones impulsionnels [Fan+21b; Tan+22c], de la distillation de savoir [Xu+23], etc.). Bien que l'efficacité de ces modules soit démontrée, leur utilisation pourrait entraver la réalisation d'une étude rigoureuse et équitable sur le rôle des neurones impulsionnels dans les performances obtenues. Cela serait problématique pour l'objectif d'analyser en profondeur les CSNNs dans ce chapitre, car l'incorporation de tels modules limiterait la portée de nos conclusions. Par conséquent, le CSNN adopté dans notre modèle générique est composé de neurones IF et suit la structure d'un encodeur convolutif [Fan+21a]. De cette manière, nous pouvons réutiliser l'un des nombreux CSNNs profonds déjà développés pour la classification [Fan+21a; Kim+22b; Xia+22; Fan+21b].

En résumé, en tenant compte de ces trois critères grâce à des choix de conception spécifiques, nous obtenons un modèle simple et générique, constitué d'un encodeur convolutif à neurones impulsionnels IF pour l'extraction de caractéristiques, ainsi que d'une couche dense linéaire pour la prédiction. Le tout forme un modèle entraîné de bout en bout via la BPTT et le substitut du gradient [NMZ19]. La Figure 4.6 offre une vue d'ensemble du modèle proposé.

Le reste de cette section détaille le fonctionnement interne des composants qui forment le modèle générique proposé.

4.2.2 Méthode : Encodeur Convolutif Impulsionnel

Comme montré dans la partie centrale de la Figure 4.6, l'encodeur convolutif est principalement un CSNN profond, composé de neurones IF (définis dans la Section 2.2.2.3). Il prend un tenseur d'impulsions $\mathbf{X}_T \in \mathbb{B}^{T\times C\times H\times W} = \{X\}_{t=1}^{T}$ en entrée, et produit un vecteur de caractéristiques $\mathcal{F} \in \mathbb{R}^K$, où K est le nombre de canaux en sortie. Le tenseur d'impulsions $\mathbf{X}_T$ peut être obtenu soit via un flux d'événements provenant d'une caméra événementielle qui a été discrétisé (comme expliqué dans la Section 2.3.3), soit par le codage neuronal d'une image statique (comme expliqué dans la Section 2.2.3). Comme mentionné par la formulation décrite en Section 2.1.2,

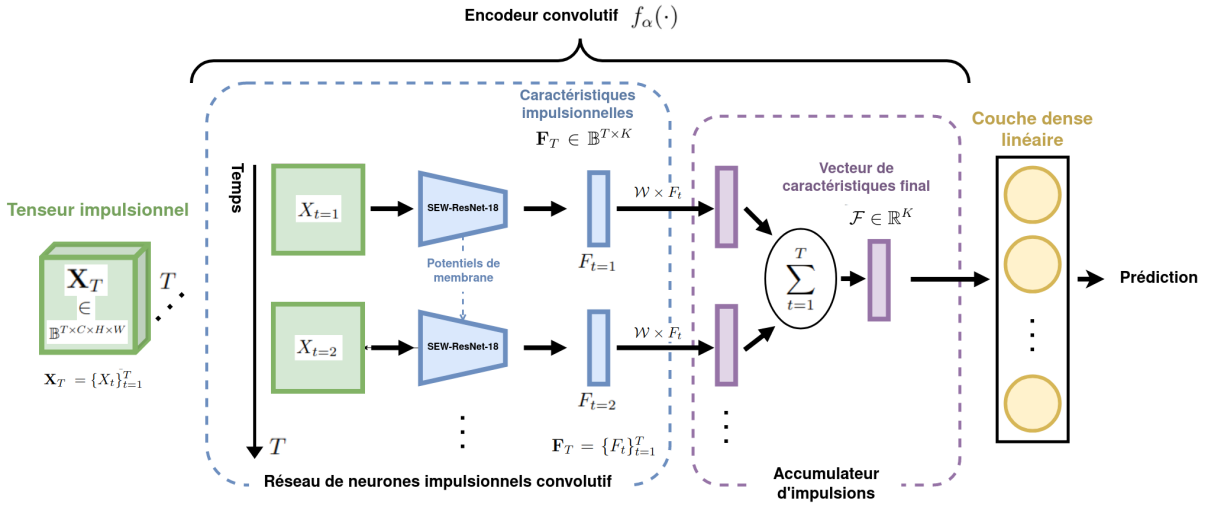


Figure 4.6: Modèle générique basé sur un CSNN proposé. Un encodeur convolutif impulsionnel $f_\alpha(\cdot)$ prend en entrée le tenseur impulsionnel $\mathbf{X}_T$ afin d'extraire un vecteur de caractéristiques $\mathcal{F} \in \mathbb{R}^K$. Pour ce faire, l'encodeur est composé de **(1)** une architecture CSNN profonde (SEW-ResNet-18 [Fan+21a]) ; et **(2)** un accumulateur d'impulsions permettant d'agréger les T caractéristiques impulsionnelles du CSNN en un vecteur unique de valeurs réelles. Ce vecteur de caractéristiques $\mathcal{F}$ peut alors être utilisé par une couche dense linéaire dédiée à générer la prédiction pour la tâche de vision visée. L'ensemble du modèle est entraîné par apprentissage supervisé via la BPTT et le substitut du gradient [NMZ19].

on désigne cet encodeur convolutif comme une fonction $f_\alpha(\cdot)$ avec α représentant l'ensemble des paramètres entraînables.

Étant composé de neurones impulsionnels, le CSNN profond de l'encodeur traite chaque image impulsionnelle $X_t \in \mathbf{X}_T$ à chaque étape temporelle $1 \leq t \leq T$. De plus, un CSNN produit un vecteur de caractéristiques binaire à chaque étape temporelle : on note $\mathbf{F}_T \in \mathbb{B}^{T\times K} = \{F_t\}_{t=1}^T$, l'ensemble des vecteurs binaires en sortie produits par le CSNN pour les T étapes temporelles de l'inférence. Par conséquent, ce fonctionnement binaire des neurones impulsionnels ne permet pas d'obtenir un vecteur de caractéristiques $\mathcal{F}$ unique pour l'ensemble des T étapes temporelles et composé de *valeurs réelles*, ce qui permettrait une meilleure expressivité. Pour remédier à cette problématique, nous adoptons un module nommé "**accumulateur d'impulsions**" [KCP21; Ran+21], qui permet de calculer le vecteur $\mathcal{F}$ à partir de $\mathbf{F}_T$.

En principe, l'architecture du CSNN importe peu, tant que celle-ci est conforme au format de l'entrée (le tenseur d'impulsions $\mathbf{X}_T$) et de la sortie (l'ensemble des caractéristiques impulsionnelles $\mathbf{F}_T$). De plus, il est possible d'employer des CSNNs profonds qui ont déjà montré de bonnes performances en classification, tels que les Spiking-ResNets [Zhe+21], SEW-ResNets [Fan+21a], etc. Dans ce chapitre, **nous**

employons une architecture SEW-ResNet-18 [Fan+21a], un CSNN à 18 couches qui a montré de très bonnes performances sur des problèmes de vision complexes, notamment sur ImageNet [Den+09] (c'est-à-dire, la classification d'images en 1000 classes). Par conséquent, le nombre de canaux des vecteurs de caractéristiques produits par l'encodeur convolutif est $K = 512$.

4.2.2.1 Accumulateur d'Impulsions

L'accumulateur d'impulsions est un module récemment introduit dans plusieurs travaux portés sur des SNNs profonds supervisés, notamment pour l'estimation de profondeur [Ran+21] (c'est-à-dire, un problème de régression) ainsi que pour la segmentation sémantique [KCP21] (un problème de classification). Son rôle est d'accumuler les impulsions provenant d'une ou plusieurs couches du CSNN pour produire, à la fin des T étapes temporelles, les caractéristiques finales $\mathcal{F}$.

Dans notre modèle générique, cet accumulateur d'impulsions est placé à la suite de la dernière couche du CSNN profond, donc les impulsions en entrées sont celles de l'ensemble $\mathbf{F}_T$. L'accumulateur est composé d'une couche dense dont les poids entraînables sont regroupés dans l'ensemble $\mathcal{W} \in \mathbb{R}^{K \times K}$. Pour calculer le vecteur $\mathcal{F}$, il accumule les caractéristiques impulsionnelles $\{F_t\}_{t=1}^{T}$ tel que :

$$\mathcal{F} = \sum_{t=1}^{T} \mathcal{W} \times F_t \tag{4.2}$$

Concrètement, l'utilisation d'un accumulateur d'impulsions demande l'ajout d'une couche dense supplémentaire, et donc l'ajout des paramètres $\mathcal{W}$ dans l'ensemble des paramètres entraînables de l'encodeur $f_\alpha(\cdot)$ ($\mathcal{W} \subseteq \alpha$).

Implémentation Neuromorphique. La formulation proposée de l'accumulateur d'impulsions peut sembler non triviale pour une implémentation sur du matériel neuromorphique. Cependant, il est possible d'envisager une solution à cette problématique en utilisant les mécanismes de neurones impulsionnels. Comme mentionné dans [Ran+21; KCP21], l'accumulateur d'impulsions peut être réalisé en employant une couche de neurones IF avec un seuil d'activation infini (c'est-à-dire, $\theta = +\infty$). Dans cette configuration, le potentiel de membrane de ces neurones représente le vecteur de caractéristiques $\mathcal{F}$. Cependant, il est important de noter qu'une telle implémentation neuromorphique suppose un accès facile aux potentiels de membrane de ces neurones IF, ainsi que la possibilité de fixer un seuil d'activation non-atteignable

(voire infini). Cette exigence peut poser des défis dans certaines architectures de matériel neuromorphique [Bas+22].

4.2.3 Méthode : Calcul de la Prédiction

L'encodeur convolutif $f_\alpha(\cdot)$ permet d'obtenir un vecteur de caractéristiques $\mathcal{F}$ à partir d'un tenseur d'impulsions $\mathbf{X}_T$ en entrée, en utilisant uniquement des mécanismes issus de neurones impulsionnels IF. Pour obtenir une prédiction à partir de cet extracteur de caractéristiques neuromorphique, nous ajoutons un dernier composant au modèle générique proposé : une couche dense linéaire (illustrée dans la partie droite de la Figure 4.6).

Cette couche dense linéaire peut être adaptée à tout type de problème, qu'il s'agisse de classification ou de régression, du moment que la sortie attendue correspond au format d'un vecteur. Dans les cas étudiés dans ce chapitre, tels que la localisation d'objet et la reconnaissance d'expressions faciales, la couche dense linéaire conçue est composée d'autant de neurones que nécessaires en fonction du format de la prédiction. Par exemple, pour la localisation d'objet, il y aura 4 neurones pour représenter les coordonnées de la boîte englobante $\mathbf{B}$, et pour la reconnaissance d'expressions faciales, il y aura autant de neurones que de types d'expressions différentes.

Ainsi conçu, le modèle générique peut être entraîné de bout en bout grâce à la BPTT et au substitut du gradient [NMZ19], via une fonction de coût spécifique pour la tâche de vision ciblée.

4.2.4 Avantages et Inconvénients du Modèle

Par ses choix de conception, le modèle générique proposé répond aux critères mentionnés en Section 4.2.1. Il permet donc une certaine polyvalence dans le type d'entrée et de sortie visés et a une architecture simple car la principale composante du modèle est un encodeur convolutif. De plus, l'emploi de neurones impulsionnels uniquement dans le processus d'extraction de caractéristiques facilite les études menées dans ce chapitre.

Au-delà de l'étude, un tel modèle est susceptible de présenter des difficultés pour une utilisation pratique. En effet, son déploiement potentiel sur du matériel neuromorphique soulève une série d'enjeux. Premièrement, l'architecture du CSNN en 18 couches (SEW-ResNet-18 [Fan+21a]) est très lourde (environ 11 millions de paramètres) pour être déployée sur les puces neuromorphiques actuelles [Bas+22],

à moins d'utiliser des architectures à plusieurs cœurs [Fur+14] et/ou de grande envergure [WTV18]. Deuxièmement, comme expliqué en Section 4.2.2.1, l'emploi de l'accumulateur d'impulsions pour obtenir le vecteur de caractéristiques utilisé lors du calcul de la prédiction implique la nécessité de pouvoir récupérer efficacement le potentiel de membrane d'une couche de neurones impulsionnels, ce qui n'est pas toujours possible selon le matériel neuromorphique utilisé [Dav+18]. Pour finir, le mode de fonctionnement du modèle implique que le CSNN utilisé doit être réinitialisé à chaque inférence, c'est-à-dire que les potentiels de membrane des neurones IF doivent être remis à leur potentiel de repos ($U_t^l = 0$, dans nos travaux) après chaque prédiction. Le modèle ne peut donc pas fonctionner en continu lors du déploiement, et le matériel neuromorphique utilisé doit pouvoir réinitialiser les valeurs des neurones IF efficacement. Il faut toutefois mentionner que ce désavantage est observé dans de nombreux travaux de l'état de l'art [KCP21; Ran+21; Fan+21a; Fan+21b].

4.3 Étude du Modèle Générique pour Traiter l'Information Spatiale via la Localisation d'Objet

Dans cette section, nous entreprenons une étude pour évaluer la capacité d'un CSNN profond supervisé (via BPTT et substitut du gradient [NMZ19]) à extraire des caractéristiques utiles pour le traitement des informations spatiales. Pour ce faire, nous utilisons le modèle générique défini à la Section 4.2 pour la même tâche de vision que celle abordée dans la preuve de concept (présentée à la Section 4.1), à savoir la localisation d'objet. Notre étude vise à analyser plusieurs aspects liés à des choix de conception fondamentaux des SNNs : le codage neuronal d'une image statique, la latence de l'inférence (c'est-à-dire, le nombre d'étapes temporelles T), etc.

4.3.1 Contexte de l'Étude

De manière similaire à la Section 4.1, la tâche de vision ciblée est la localisation d'objet, et celle-ci est formulée de la même manière qu'exposée dans la Section 4.1.2. Cependant, les objectifs de cette section diffèrent de la preuve de concept, car ils visent à explorer le comportement d'un CSNN selon différents aspects, contrairement à la preuve de concept où l'objectif était de vérifier l'applicabilité des SNNs profonds supervisés pour la localisation d'objet.

Dans cette étude, nous analysons les performances de localisation d'objet en variant les choix de conception du modèle générique. Nous examinons également deux types d'entrées différents : les codages neuronaux d'images statiques et les flux d'événements provenant de capteurs événementiels. La Figure 4.7 illustre les processus pour obtenir un tenseur impulsionnel $\mathbf{X}_T$ à partir de ces deux modalités.

Dans le reste de cette section, nous détaillons les aspects analysés dans notre étude.

Analyse de la Latence. L'hyperparamètre de latence, représenté par le nombre d'étapes temporelles T, joue un rôle crucial dans la conception d'un SNN. Il a une double signification : d'une part, il influence la précision de la simulation du SNN [PB12], et d'autre part, il détermine le temps nécessaire au modèle pour générer une prédiction [CRR22]. En d'autres termes, un T plus élevé devrait conduire à une prédiction plus précise, mais cela entraînerait également une plus grande durée de fonctionnement du SNN et une consommation d'énergie accrue [Lem+22]. Alors que les premières études sur les CSNNs profonds utilisaient des valeurs de T allant jusqu'à

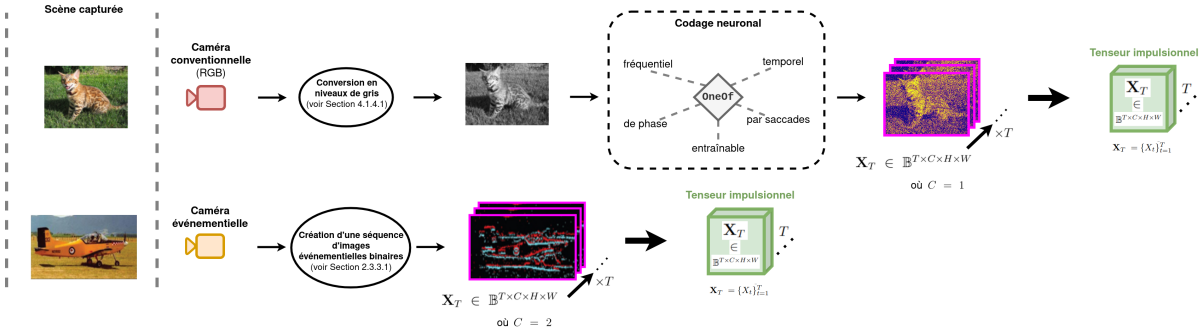


Figure 4.7: Vue d'ensemble du procédé de création du tenseur impulsionnel $\mathbf{X}_T$ à partir d'une image statique ou d'un flux d'événements. Concernant les images statiques, nous étudions divers schémas de codages neuronaux (*fréquentiel*, *temporel*, *par phases*, *par saccades* ou *entraînable*). Pour les flux d'événements, nous appliquons la génération d'une séquence d'images événementielles binaires, comme expliqué en Section 2.3.3.1.

1000 [NMZ19; KMN20], des travaux récents ont montré de meilleures performances en réduisant considérablement le nombre d'étapes temporelles nécessaires [Fan+21a; Fan+21b; CRR22; CRR21]. Par exemple, dans [CRR21], un modèle CSNN atteint des performances comparables à l'état de l'art avec un $T = 1$. Cependant, ces études utilisent des modèles SNN différents, ce qui limite notre compréhension de ces observations. Dans notre étude, nous entrainons le même modèle en variant la valeur de T afin d'évaluer son impact sur les performances.

Analyse de Robustesse aux Corruptions. En plus d'évaluer les performances d'un algorithme de vision sur des données non altérées, il est essentiel de comprendre sa robustesse face à des données corrompues, notamment en raison de capteurs défectueux. De manière similaire à notre analyse réalisée pour l'approche Bina-Rep en Section 3.3.4, nous évaluons la robustesse de notre modèle générique face à des corruptions courantes affectant les images statiques ou les flux d'événements provenant des caméras événementielles (en fonction du type d'entrée étudié). Cette évaluation de la robustesse est menée en introduisant des corruptions de données avec différents niveaux de sévérité, allant de 1 à 5, de manière croissante. La métrique utilisée pour évaluer la dégradation des performances est le score de baisse de précision relative ("*Relative Accuracy Drop*", en anglais) RAD^{corr}_{sev}, comme décrit dans l'Annexe A.1. De plus, les scores de baisse de précision relative obtenus pour chaque niveau de sévérité sont agrégés pour calculer un score moyen de baisse de précision relative globale ($mRAD^{corr}$).

Estimation du Coût Énergétique. Malgré les défis associés à l'accès au matériel neuromorphique, il existe plusieurs approches pour estimer la consommation énergé-

tique d'un SNN [Lem+22; KCP21; Dav+18]. Ces estimations se basent souvent sur le nombre d'impulsions émises par chaque couche de neurones et peuvent fournir des indications utiles, bien qu'elles ne prennent généralement pas en compte tous les aspects liés à la consommation d'énergie, tels que la gestion de la mémoire ou l'utilisation de circuits périphériques. Dans notre étude, nous adoptons une méthode d'estimation de la consommation d'énergie proposée par [KCP21], basée sur une puce CMOS de 45nm. Les détails de cette méthode sont fournis dans l'Annexe A.3. Il est important de noter que nos estimations du coût énergétique sont calculées pour une configuration à 8 étapes temporelles ($T = 8$).

Analyse des Codages Neuronaux pour les Images Statiques. Comme expliqué en Section 2.2.3, le traitement d'images statiques avec des SNNs requiert l'utilisation de codages neuronaux (fréquentiel, temporel, etc.) pour convertir les valeurs réelles en trains d'impulsions. Le choix du codage neuronal revêt une grande importance, car il détermine la manière dont les informations sont transmises au SNN. Bien que dans le domaine des CSNNs entraînés par apprentissage bio-inspiré (typiquement, STDP [BP98]), des études aient montré la supériorité du codage temporel par rapport au codage fréquentiel [Fal19] ou à d'autres types de codage (par phase, etc.) [Guo+21], des études similaires n'ont pas encore été menées dans le contexte de l'apprentissage supervisé pour CSNNs profonds. Par conséquent, dans notre étude sur les images statiques, nous comparons plusieurs codages neuronaux populaires de l'état de l'art [TDV01; Bon+22; Par+20; Kim+18; Bre15] en tenant compte des autres aspects présentés dans cette section (la latence, la robustesse aux corruptions et la consommation énergétique).

4.3.2 Adaptation du Modèle Générique CSNN

Pour traiter la tâche de localisation, nous adaptons le modèle générique CSNN défini en Section 4.2. La Figure 4.8a reprend la Figure 4.6, et intègre les modifications nécessaires pour la localisation d'objet.

Concrètement, la seule modification de la structure du réseau réside dans l'adaptation de la couche dense linéaire destinée à générer la prédiction de la boîte englobante $\mathbf{B}$ à partir du vecteur de caractéristiques $\mathcal{F} \in \mathbb{R}^K$ de l'encodeur convolutif $f_\alpha(\cdot)$. Cette couche dense est composée de 4 sorties normalisées entre 0 et 1. Le modèle est entraîné de bout en bout en utilisant la fonction de coût "Distance-Intersection over Union" (DIoU) [Zhe+20]. Cette fonction de coût est décrite en Annexe A.2.

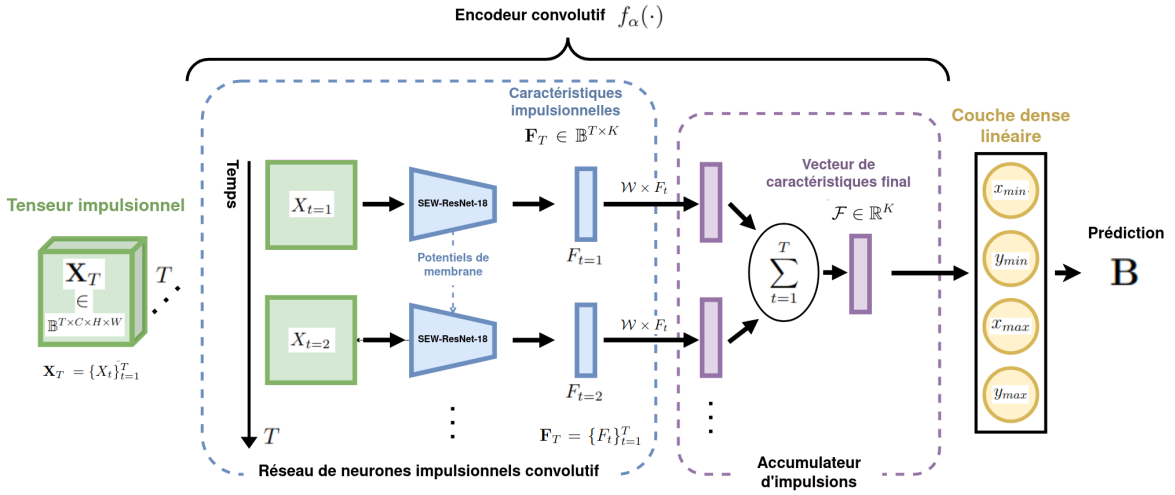
4.3.3 Comparaison avec un Réseau de Neurones Artificiels Similaire

En plus d'analyser différents aspects pour un CSNN profond supervisé, notre étude réalise une comparaison entre un CSNN (via le modèle générique proposé en Section 4.2) et un ANN de référence avec une architecture similaire selon les aspects étudiés (voir Section 4.3.1). Cette comparaison vise à identifier les différences entre les ANNs et les SNNs dans le traitement des données visuelles (images statiques et flux d'événements) dans le contexte de la localisation d'objet. La Figure 4.8 présente une vue d'ensemble des modèles utilisés dans notre analyse, y compris l'adaptation du modèle générique pour la localisation d'objet (Figure 4.8a) et le modèle basé sur l'ANN (Figure 4.8b).

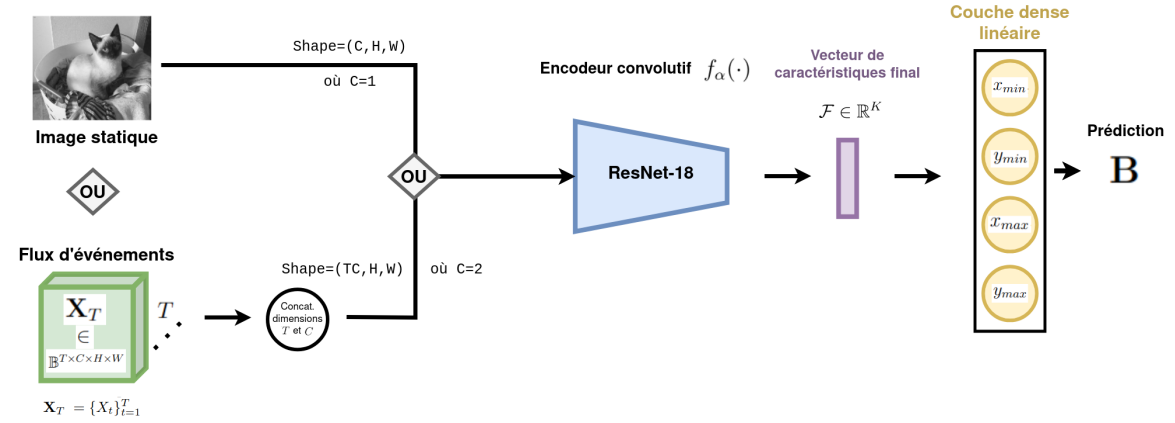
Le modèle basé sur un ANN est conçu dans la même philosophie que le modèle générique présenté en Section 4.2. Cependant, contrairement au modèle générique où l'encodeur convolutif $f_\alpha(\cdot)$ est un CSNN (plus précisément, une architecture SEW-ResNet-18 [Fan+21a]) suivi d'un accumulateur d'impulsions, le modèle ANN utilise directement une architecture CNN profonde pour extraire le vecteur de caractéristiques $\mathcal{F}$. Afin d'assurer une comparaison équitable, l'architecture ANN utilisée est un ResNet-18 [He+16], car celle-ci est similaire au CSNN SEW-ResNet-18 [Fan+21a] en termes de nombre de couches (18), de structure du réseau, et de paramètres ($\approx$ 11M).

Le modèle ANN étant basé sur une architecture 2D-CNN, le format d'entrée doit être de dimension $(C \times H \times W)$. Par conséquent, le format des données en entrée, que ce soit pour les images statiques ou les flux d'événements, diffère de celui utilisé pour le modèle CSNN. Pour les images statiques, l'entrée utilisée est directement l'image statique, sans nécessiter de codage neuronal préalable comme pour le CSNN, étant donné qu'un ANN n'est pas limité au traitement de trains d'impulsions. En ce qui concerne les flux d'événements, la problématique est semblable à celle décrite en Section 3.2, où le tenseur d'impulsions en 4 dimensions $(T \times C \times H \times W)$ n'est pas adapté au traitement par une architecture 2D-CNN. Par conséquent, nous procédons à la concaténation des dimensions T et C du tenseur d'impulsions pour obtenir une représentation sous forme de tenseur $(TC \times H \times W)$.

Tout comme le modèle CSNN, le modèle ANN de référence est optimisé de bout en bout en utilisant la fonction de coût DIoU [Zhe+20] (voir Annexe A.2).



(a) Adaptation du modèle générique présenté en Section 4.2 pour la localisation d'objet.



(b) Modèle de référence basé sur un ANN. Tout comme le modèle générique, un encodeur convolutif est utilisé pour extraire un vecteur de caractéristiques permettant de générer une prédiction. Dans le but de réaliser une comparaison équitable, l'ANN utilisé (un ResNet-18 [He+16]) est similaire au CSNN SEW-ResNet-18 [Fan+21a] du modèle générique, dans le sens où la structure et la complexité des réseaux sont semblables. Pour les flux d'événements, l'entrée est le tenseur impulsionnel $\mathbf{X}_T$ dont les deux premières dimensions ont été concaténées, tandis que les images statiques sont directement utilisées comme entrée sans passer par un schéma de codage neuronal.

Figure 4.8: Vue d'ensemble des modèles utilisés dans notre étude sur la localisation d'objet.

4.3.4 Détails d'Implémentation

Dans cette section, nous détaillons les aspects d'implémentation qui sont communs à l'ensemble des expérimentations menées dans notre étude. La simulation des neurones impulsionnels a été réalisée sur un GPU NVIDIA 2080Ti en utilisant Spikingjelly [Fan+20], un simulateur de réseaux de neurones impulsionnels basé sur PyTorch [Pas+19]. Les modèles ont été entraînés sur 150 époques en utilisant l'optimiseur Adam [KB14]. Le taux d'apprentissage de chaque modèle a été déterminé en utilisant une méthode de recherche de taux d'apprentissage (*learning rate finder*) [Smi17]. La

résolution de la caméra pour chaque échantillon (flux d'événements et images statiques) a été redimensionnée à $(H \times W) = (224 \times 224)$.

4.3.5 Configuration pour les Images Statiques

Dans cette section, nous détaillons les paramètres et les conditions de notre étude concernant les images statiques.

4.3.5.1 Codages Neuronaux Étudiés

Comme expliqué en Section 2.2.3, un codage neuronal est une fonction $u(\cdot)$ qui prend une image statique $\mathbf{I} \in \mathbb{R}^{C \times H \times W}$ en entrée, et retourne un ensemble de trains d'impulsions binaires que nous discrétisons en un tenseur d'impulsions $\mathbf{X}^{T \times C \times H \times W}$. Dans notre étude sur les images statiques, les codages neuronaux comparés sont : **codage fréquentiel** [Bre15], **codage temporel** [TDV01; Bon+22; Par+20], **codage par phases** [Kim+18], **codage par saccades**, et **codage entraînable** [Fan+21a]. Les codages fréquentiel, temporel et par phases ont été respectivement formulés en Sections 2.2.3.1, 2.2.3.2, et 2.2.3.3. Les codages entraînable et par saccades sont détaillés ici, car ils sont soit conçus pour notre approche (codage par saccades) ou développés spécifiquement pour un CSNN entraîné par BPTT (codage entraînable). La Figure 4.9 montre une visualisation des codages neuronaux évalués.

Codage par Saccades (Saccades Coding)**.** Dans ce travail, nous introduisons une nouvelle méthode de codage neuronal appelée "codage par saccades", qui vise à simuler numériquement le processus de capture de flux d'événements proposé par [Orc+15]. Dans [Orc+15], une caméra événementielle est positionnée devant un écran diffusant des images statiques. Cette caméra est montée sur une plateforme motorisée qui effectue trois mouvements de saccades successifs [LFoo]. Ces saccades produisent un mouvement qui permet à la caméra événementielle de générer des événements en réponse à l'image affichée sur l'écran. Pour simuler cette procédure, nous créons tout d'abord une séquence de T images qui représentent trois translations successives, simulant ainsi les "saccades". Ces translations sont utilisées pour déplacer progressivement l'image en fonction de deux valeurs de distance (en pixels) $r_x = 0.09 \times W$ et $r_y = 0.09 \times H$. La Figure 4.10 illustre ces saccades simulées, qui permettent de générer la séquence d'images translatées. Ensuite, nous appliquons un processus de modulation delta (issu de SnnTorch [Esh+21]) à la séquence d'images pour obtenir le tenseur d'impulsions final $\mathbf{X}_T$. La modulation delta génère une impulsion sur un

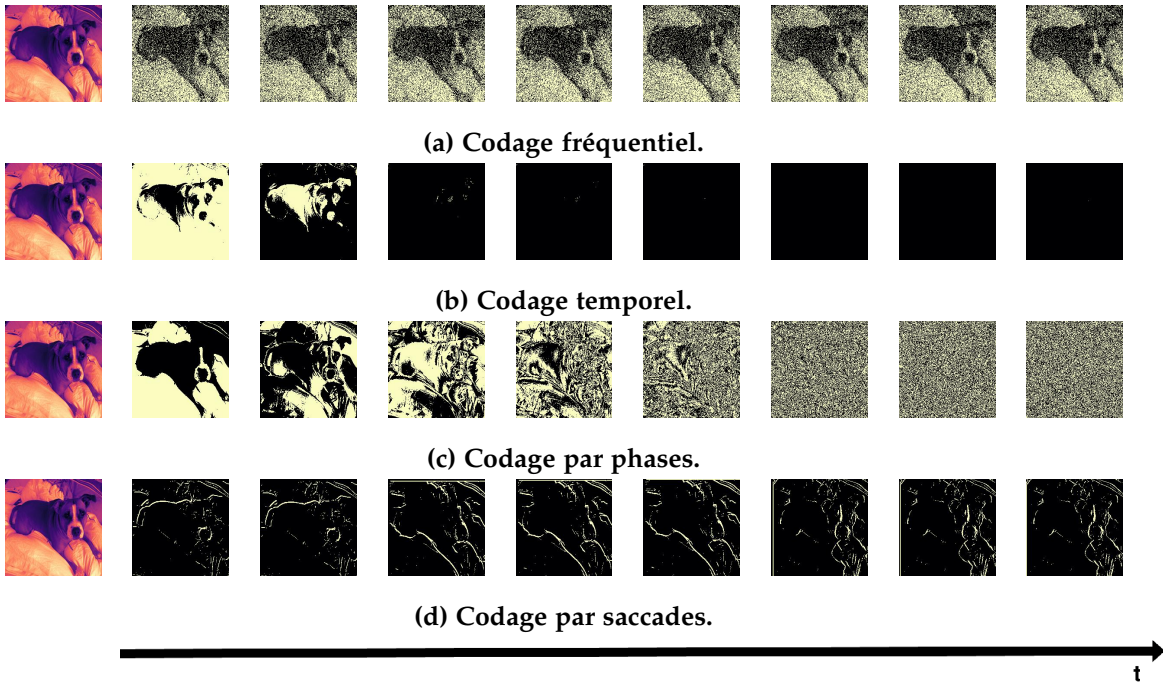


Figure 4.9: Illustration des schémas de codages neuronaux étudiés sur un exemple de Oxford-IIIT-Pet [Par+12].

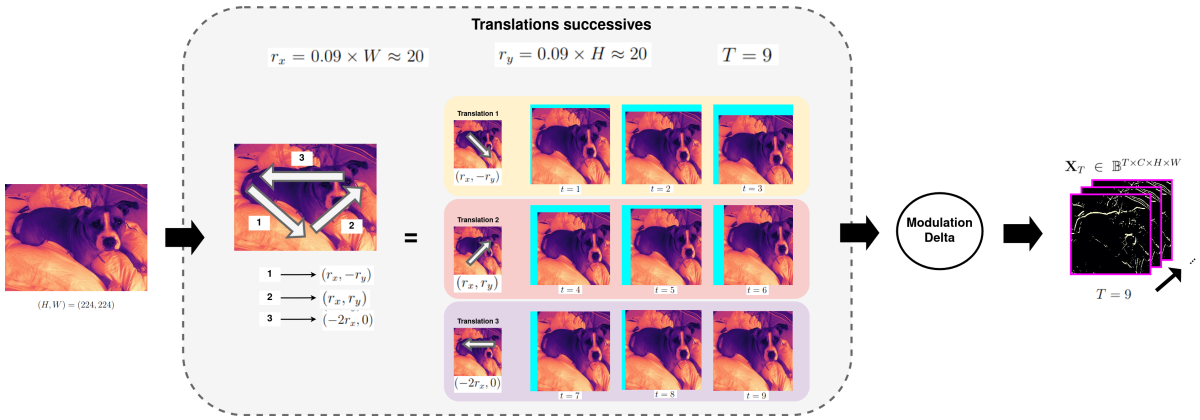


Figure 4.10: Schéma explicatif du codage par saccades. À partir d'une image statique, trois translations successives qui simulent les mouvements de saccades présentés dans [Orc+15] permettent de créer une séquence de T images. Ensuite, on applique une modulation delta [Esh+21] afin de créer le tenseur impulsionnel $\mathbf{X}_T$.

pixel donné lorsque le changement d'intensité entre deux images successives de la séquence dépasse un seuil défini (ici, le seuil est de 0.1). La Figure 4.10 détaille le processus de codage par saccades, mettant en évidence les trois translations successives. Une illustration du résultat du codage par saccades est présentée dans la Figure 4.9d, mettant en évidence des similitudes avec les échantillons de NCaltech-101 [Orc+15], présentés dans la Figure 4.16.

Codage Entraînable (Trainable Coding)**.** Dans le schéma de codage entraînable [Fan+21a], la première couche de convolution de notre modèle CSNN SEW-ResNet-18 [Fan+21a] est utilisée comme un mécanisme de codage neuronal. Concrètement, l'image $\mathbf{I}$ est introduite directement en entrée de cette couche. La sortie a une résolution de $32 \times \frac{H}{2} \times \frac{W}{2}$, résultant d'une opération de convolution. Ensuite, l'activation des neurones IF est appliquée à cette sortie, générant ainsi des impulsions. Ces impulsions sont ensuite répétées sur les T étapes temporelles, formant ainsi le tenseur d'impulsions $\mathbf{X}_T$. En conséquence, ce schéma de codage permet de construire un tenseur d'impulsions $\mathbf{X}_T$ sous la forme de cartes de caractéristiques de bas niveau. De plus, ces caractéristiques sont optimisées de bout-en-bout grâce à la rétropropagation lors de l'apprentissage.

4.3.5.2 Corruptions Étudiées

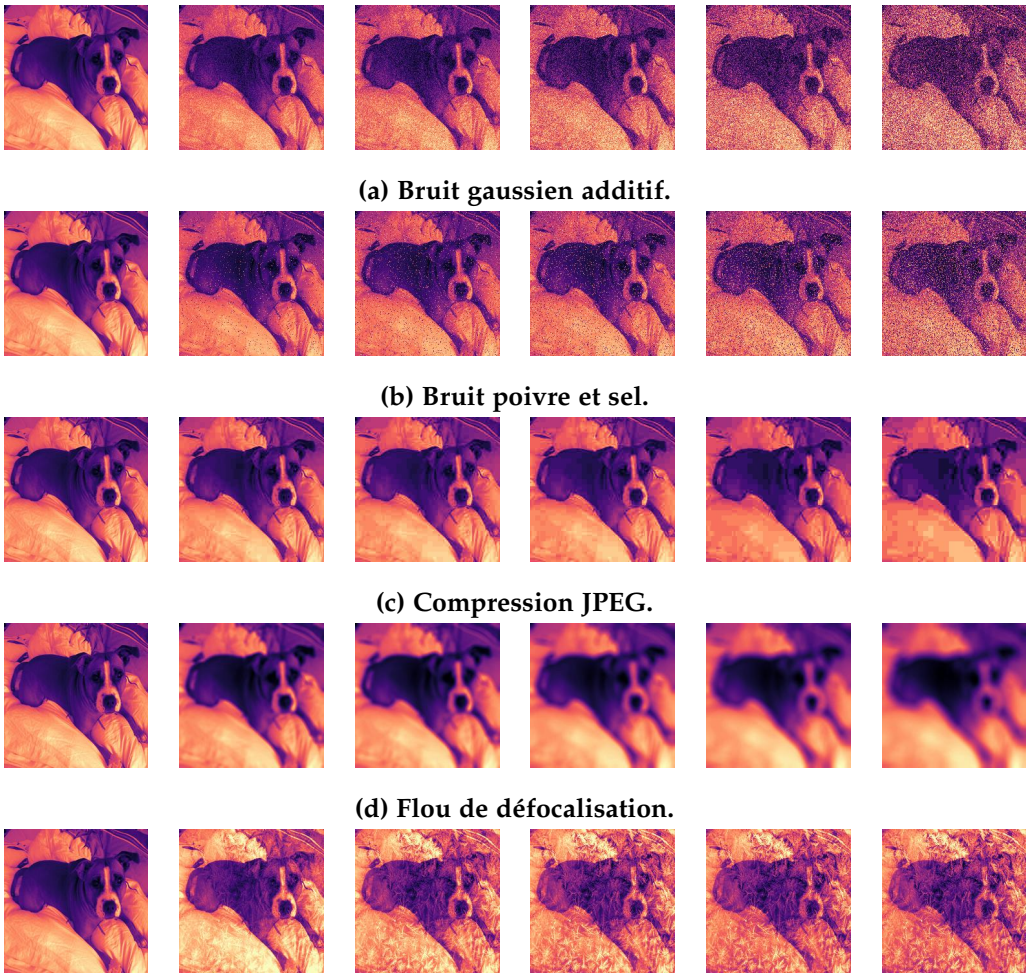
Dans notre étude sur les images statiques, nous appliquons des corruptions spécifiques à l'image statique $\mathbf{I}$ avant d'appliquer le codage neuronal. Cela nous permet d'évaluer la robustesse du modèle aux corruptions similaires à celles que pourraient rencontrer les capteurs conventionnels. Les corruptions que nous utilisons sont issues d'une étude antérieure sur la robustesse en vision artificielle [HD19], et nous utilisons les implémentations originales[2]. La Table 4.1 présente les différents paramètres de chaque corruption pour différentes valeurs de sévérité. La Figure 4.11 illustre des exemples de l'application de ces corruptions, avec différents niveaux de sévérité.

Bruit Gaussien Additif. Le bruit gaussien additif correspond à un type de perturbation dans lequel des variations aléatoires sont ajoutées aux valeurs des pixels de l'image. Ici, ces variations suivent une distribution normale avec une moyenne de zéro et un écart-type σ, notés $(0, \sigma)$. C'est un type de bruit fréquemment rencontré, pouvant être causé par divers facteurs tels que les fluctuations électriques lors de la capture de l'image. Le niveau de sévérité de la perturbation ajuste la valeur de l'écart-type σ de la distribution normale, influençant ainsi l'intensité du bruit ajouté aux pixels.

[2] https://github.com/hendrycks/robustness

Corruption	Hyperparamètre(s)	Sévérité 1	2	3	4	5
Bruit Gaussien Additif	σ	0.08	0.12	0.18	0.26	0.38
Bruit Poivre et Sel	sp	0.03	0.06	0.09	0.17	0.27
Compression JPEG	Qualité %	25	18	15	10	7
Flou de Défocalisation	r_{defocus}	3	4	6	8	10
Perturbation du Givre	(κ, λ)	(1, 0.4)	(0.8, 0.6)	(0.7, 0.7)	(0.65, 0.7)	(0.6, 0.75)

Table 4.1: Définition des valeurs des hyperparamètres des corruptions d'images statiques étudiées, pour chacun des 5 niveaux de sévérité.



(a) Bruit gaussien additif.

(b) Bruit poivre et sel.

(c) Compression JPEG.

(d) Flou de défocalisation.

(e) Perturbations du givre.

Figure 4.11: Illustrations des corruptions d'images statiques étudiées. La première colonne montre l'exemple sans corruption, et les colonnes suivantes illustrent les corruptions avec un niveau croissant de sévérité (de 1 à 5).

Bruit Poivre et Sel. Le bruit poivre et sel est une forme de corruption qui se traduit par la conversion aléatoire de certains pixels de l'image en valeurs extrêmes, soit la valeur minimale (représentant le "poivre") ou la valeur maximale (représentant le "sel"). Ce type de bruit peut survenir en raison de divers facteurs tels que des dysfonctionnements de la caméra, des erreurs de transmission des données captées, ... [AZA18]. Un certain pourcentage, noté $sp\%$, des pixels de l'image est choisi au hasard et modifié pour avoir soit la valeur minimale, soit la valeur maximale. Le niveau de sévérité est défini par le pourcentage sp des pixels affectés par cette corruption.

Compression JPEG. La corruption par compression JPEG provient de la perte d'informations et, par conséquent, de la détérioration de la qualité de l'image résultant de l'utilisation de l'algorithme de compression JPEG [PM92]. Dans cette étude, nous appliquons l'algorithme de compression JPEG en définissant un pourcentage de qualité. Le niveau de sévérité est déterminé par le pourcentage de qualité de l'image conservée.

Flou de Défocalisation. Le flou de défocalisation correspond à l'effet de flou généré lorsque la scène capturée se trouve en dehors de la plage de mise au point de la caméra. Une méthode courante pour simuler cette corruption [OFB13] consiste à appliquer un filtre en forme de disque flou avec un rayon r_{defocus}, qui est un paramètre défini par le niveau de sévérité. En d'autres termes, un rayon plus grand entraîne un flou de défocalisation plus intense.

Perturbation du Givre. La perturbation du givre fait référence aux occultations causées par des cristaux de glace se formant sur l'objectif de la caméra. Pour simuler efficacement ces perturbations, des échantillons de cristaux de glace sont choisis au hasard parmi des images préalablement générées, puis ajoutés à l'image originale d'entrée $\mathbf{I}$. Un exemple d'image de cristaux préalablement générée est présenté dans la Figure 4.12. Nous notons $\mathbf{I}_{frost} \in \mathbb{R}^{C\times H\times W}$ l'échantillon de cristaux de glace sélectionné. Le niveau de sévérité détermine dans quelle mesure les cristaux de glace occultent l'image originale. Pour ce faire, nous introduisons deux paramètres κ et $\lambda \in [0, 1]$, qui représentent respectivement l'intensité de $\mathbf{I}$ et de $\mathbf{I}_{frost}$. Ainsi, l'image corrompue résultante est obtenue en combinant $\kappa\mathbf{I}$ et $\lambda\mathbf{I}_{frost}$.

4.3.5.3 Base de Données Utilisée

Pour établir une comparaison entre les performances de notre modèle générique et celles du modèle présenté dans la preuve de concept de la Section 4.1, nous pour-

Figure 4.12: Exemple d'une image pré-générée de cristaux de glace pour la perturbation du givre.

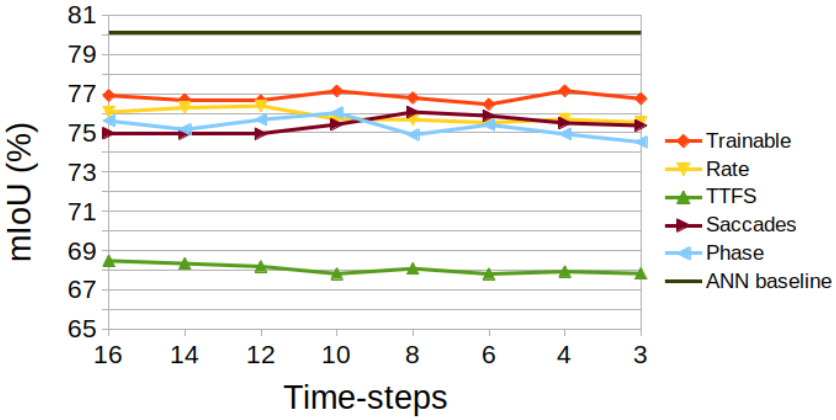


Figure 4.13: Performances de notre approche pour les images statiques en fonction du nombre total d'étapes temporelles T.

suivons l'évaluation de notre modèle sur le jeu de données Oxford-IIIT-Pet [Par+12]. Nous utilisons des images en niveaux de gris, ce qui se traduit par $C = 1$ dans le tenseur impulsionnel $\mathbf{X}_T$. Vous pouvez retrouver les détails de cet ensemble de données dans la Section 4.1.4.1.

4.3.6 Résultats pour les Images Statiques

Dans cette section, nous présentons les résultats de nos analyses sur les images statiques.

4.3.6.1 Analyse de la Latence

La Figure 4.13 illustre les performances de notre approche concernant les images statiques, en fonction du nombre total d'étapes temporelles T alloué durant l'inférence. Nous pouvons constater que notre modèle générique basé sur un CSNN obtient

des résultats cohérents pour toutes les méthodes de codages neuronaux examinées, ce qui met en évidence l'adaptabilité de l'accumulateur d'impulsions à différents types d'entrées. De plus, il est à noter que notre modèle présente généralement des performances légèrement inférieures à celles de l'ANN dans la plupart des cas.

Nos résultats ne révèlent aucune corrélation significative entre le nombre d'étapes temporelles (T) et les performances finales des CSNNs entraînés par rétropropagation. Ces conclusions diffèrent de résultats antérieurs [Wu+19], qui laissaient entendre la nécessité d'un grand nombre d'étapes temporelles. En outre, notre étude présente des conclusions divergentes par rapport à des travaux précédents sur des schémas de codage utilisant des règles d'apprentissage non supervisées d'inspiration biologique telles que STDP [Guo+21]. Bien que STDP favorise le codage temporel pour améliorer la précision (par rapport au codage fréquentiel), nos résultats montrent que le codage temporel est moins performant que d'autres méthodes. Cette différence suggère que les règles d'apprentissage STDP pourraient ne pas être adaptées aux entrées à forte intensité d'impulsions provenant d'images encodées en fréquence, contrairement à la BPTT. Dans l'ensemble, les performances des différents schémas de codage sont comparables, et les variations du score de précision en fonction du nombre d'étapes temporelles suivent une tendance similaire, avec une légère avance observée pour le codage entraînable.

La Table 4.2 présente les résultats quantitatifs illustrés dans la Figure 4.13 et les compare aux résultats de notre preuve de concept exposée dans la Section 4.1. Toutes les variantes de notre modèle surpassent la preuve de concept en termes de $mIoU$, tout en ayant une latence temporelle considérablement réduite. En outre, la meilleure version de notre modèle CSNN (à savoir, le codage entraînable avec $T = 4$) présente des performances inférieures de seulement 2.97% par rapport aux performances de l'ANN.

4.3.6.2 Analyse de Robustesse

Nous débutons en analysant la robustesse globale de chaque schéma de codage neuronal, ce qui se traduit par le score moyen de baisse de précision relative ($mRAD^{corr}$), comme indiqué dans la Table 4.3. En outre, la progression de cette robustesse face à une augmentation de la sévérité des corruptions est illustrée dans la Figure 4.14. En général, le modèle CSNN se montre plus résistant que l'ANN dans certains cas où un codage neuronal spécifique est employé pour traiter un type particulier de corruption. Par exemple, le codage entraînable se révèle plus robuste face à la com-

	$T =$ 16	14	12	10	8	6	4	3
Codage Entraînable	76.90	76.69	76.63	77.13	76.78	76.45	**77.14**	76.74
Codage Fréquentiel	76.06	76.28	**76.37**	75.70	75.67	75.52	75.69	75.53
Codage Temporel	**68.48**	68.34	68.19	67.82	68.08	67.81	67.93	67.83
Codage par Saccades	74.98	74.96	74.95	75.43	**76.06**	75.87	75.50	75.37
Codage par Phases	75.61	75.17	75.67	**76.02**	74.90	75.41	74.94	74.53
Modèle ANN (Section 4.3.3)	80.11							
Preuve de Concept (Section 4.1) $T = 1000$	63.2							

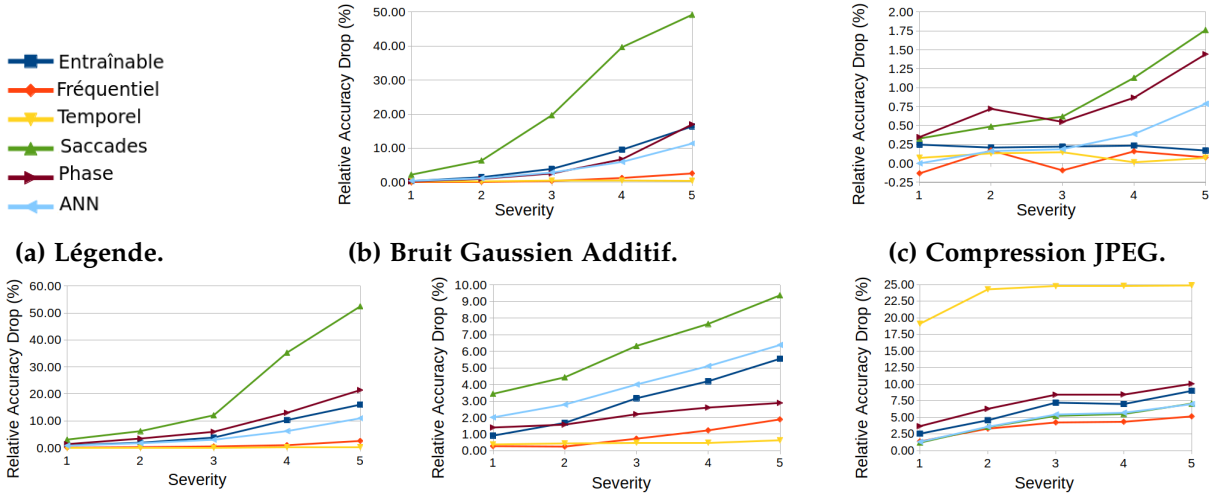
Table 4.2: Performances (en % de *mIoU*) obtenues sur Oxford-IIIT-Pet [Par+12] par le modèle générique selon chaque schéma de codage neuronal en fonction du nombre d'étapes temporelles. Pour chaque ligne, les meilleurs scores sont affichés **en gras**. De plus, les performances du modèle ANN de référence et de la preuve de concept sont données en-dessous.

	Bruit Gaussien Additif	Compression JPEG	Bruit Poivre & Sel	Flou de Défocalisation	Perturbation du Givre
Modèle ANN	4.35	0.3	4.56	4.05	4.61
Entraînable	6.34	0.22	6.62	3.1	6.08
Fréquentiel	0.87	**0.04**	0.92	0.87	3.39
Temporel	**0.38**	0.09	**0.15**	**0.48**	23.57
par Saccades	23.41	0.87	21.83	6.24	4.52
par Phases	5.51	0.79	9.06	2.13	7.39

Table 4.3: Scores moyens de baisse de précision relative pour chaque schéma de codage neuronal étudié. Les scores plus bas que ceux du modèle ANN (c'est-à-dire, indiquant une meilleur robustesse) sont affichés en vert. Les scores **en gras** indiquent le codage neuronal ayant atteint la meilleure robustesse.

pression JPEG et au flou de défocalisation, mais demeure plus vulnérable aux autres types de corruptions étudiées.

Contrairement à certaines études antérieures qui ont utilisé STDP [Fal19; Guo+21], notre constat montre que le codage temporel affiche une robustesse significativement plus élevée par rapport aux autres schémas de codages examinés. Il est généralement admis que le codage fréquentiel est plus résistant que le codage temporel, car un grand nombre d'impulsions offre la possibilité d'atténuer les erreurs dans les fréquences générées [Esh+21; Guo+21]. Cependant, nos résultats sur le codage temporel présentent une perspective alternative. Ils démontrent que le score de baisse de précision relative observé reste globalement stable, même en présence de corruptions de forte sévérité.



(a) Légende. **(b) Bruit Gaussien Additif.** **(c) Compression JPEG.**

(d) Bruit Poivre & Sel. **(e) Flou de Défocalisation.** **(f) Perturbation du Givre.**

Figure 4.14: Évolutions des scores de baisse de précision relative en fonction du niveau de sévérité pour chaque schéma de codage neuronal.

Cette observation peut s'expliquer par le fait que les bruits n'ont qu'un impact marginal sur la dépendance logarithmique entre l'intensité d'entrée et les temps de décharge des neurones. En contraste, le codage fréquentiel, avec ses impulsions fréquentes et nombreuses, offre davantage d'occasions pour que le bruit prenne effet. Comme le met en évidence la Figure 4.14f, une sensibilité marquée du codage temporel à la perturbation du givre peut être remarquée même avec une faible sévérité, ce qui suggère que ce schéma de codage repose particulièrement sur les premières impulsions d'entrée. Celles-ci sont fortement affectées par les pixels lumineux (c'est-à-dire, dont la valeur est proche de 1) constituant les perturbations induites par le givre.

D'autre part, le codage par saccades démontre une robustesse significativement plus faible par rapport aux autres techniques, notamment face à des entrées fortement corrompues. Bien qu'il puisse présenter une robustesse similaire à d'autres méthodes en cas de faible sévérité, il ne semble pas être bien adapté aux conditions où le bruit est très présent, affichant une croissance plus rapide du score de baisse de précision relative. Néanmoins, nous observons une robustesse intéressante vis-à-vis de la perturbation du givre, presque équivalente à celle de l'ANN, que ce soit en termes de score moyen ou d'évolution en fonction du niveau de sévérité. Cette observation peut être expliquée par le fait que le processus de modulation delta [Esh+21] ne génère des impulsions que lorsque des différences d'intensité entre étapes temporelles successives sont observées sur le même pixel. Étant donné que les cristaux de givre restent immobiles sur le capteur, la modulation delta ne génère pas d'impulsions supplémentaires en raison

	E_{ANN} (mJ)	E_{SNN} (mJ)	E_{ANN}/E_{SNN}
Entraînable	11129.44	248.34	44.82
Fréquentiel		208.6	53.35
Temporel		87.94	**126.6**
par Saccades		114.69	97.04
par Phases		141.79	78.49
Flux d'Événements	13 399.34	**294.63**	**45.48**

Table 4.4: Comparaison de la consommation énergétique entre le modèle générique CSNN et le modèle de référence ANN, pour les images statiques et les flux d'événements.

d'un quelconque mouvement. Ainsi, seule l'occultation causée par la perturbation du givre subsiste, ce qui réduit considérablement le potentiel de bruit.

En dernier lieu, le codage fréquentiel se révèle être le plus robuste parmi les différentes techniques de codages neuronaux testées. Il montre une résilience solide face à divers types de corruptions d'images et surpasse systématiquement le modèle ANN. Par conséquent, le codage fréquentiel apparaît comme le choix le plus approprié pour gérer une large gamme de scénarios de corruptions d'images, en parvenant à équilibrer précision et robustesse.

4.3.6.3 Estimation du Coût Énergétique

Comme détaillé dans l'Annexe A.3, la méthodologie employée [KCP21] pour évaluer le coût énergétique du CSNN et de l'ANN repose sur le calcul du nombre total d'opérations à virgule flottante (FLOPS) effectuées lors de l'inférence. Ces informations sont ensuite utilisées pour estimer la consommation énergétique sur une puce CMOS de 45 nm [Hor14]. La consommation énergétique obtenue et le ratio de consommation entre l'ANN et le modèle générique CSNN sont présentés en détail dans la Table 4.4. Le modèle générique CSNN démontre des avantages significatifs en termes de consommation énergétique par rapport au modèle ANN, avec une économie d'énergie allant de 44.82× à 126.6×.

La méthodologie d'estimation des coûts pour les CSNNs utilisée [KCP21] repose en partie sur l'activité des couches de neurones impulsionnels, c'est-à-dire la fréquence à laquelle ces couches émettent des impulsions. Étant donné que les encodeurs CSNN adoptés sont des architectures de type ResNet [He+16], le CSNN peut être subdivisé en 5 blocs de traitement. Chaque bloc traite des cartes de caractéristiques avec une

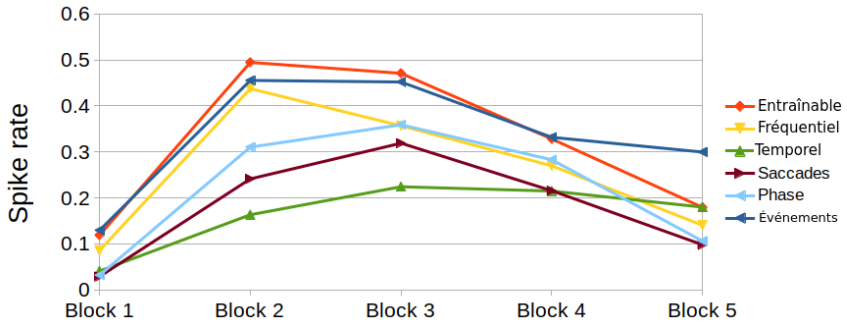


Figure 4.15: Fréquence d'émission des impulsions pour chaque bloc résiduel de l'architecture SEW-ResNet [Fan+21a], en fonction du codage neuronal employé (ou pour les flux d'événements).

résolution réduite de moitié par rapport au bloc précédent. Cette division en blocs permet une analyse plus aisée de l'activité des 18 couches composant notre architecture SEW-ResNet-18 [Fan+21a]. La Figure 4.15 illustre la fréquence des impulsions pour chaque bloc du CSNN, qui correspond à la moyenne des fréquences d'impulsions de toutes les couches au sein d'un même bloc. De manière intéressante, un schéma similaire est observé pour tous les types de codages neuronaux étudiés. En particulier, les blocs 2, 3 et 4 affichent une activité plus élevée en termes de fréquence d'impulsions par rapport au bloc 1 (qui reçoit les impulsions en entrée) et au bloc 5 (qui précède l'accumulateur d'impulsions). En d'autres termes, ces résultats suggèrent que les couches intermédiaires de l'architecture génèrent une activité d'impulsions plus prononcée que celle observée à l'entrée du réseau. Cette observation soulève des préoccupations potentielles quant à la parcimonie du CSNN et ouvre des perspectives d'amélioration de l'efficacité énergétique.

En ce qui concerne la comparaison des schémas de codages neuronaux, nos résultats sont en accord avec des recherches antérieures [Guo+21] ainsi qu'avec les observations faites dans la Figure 4.15. Les schémas de codages intensifs en impulsions (c'est-à-dire, les codages fréquentiel et par phases) présentent des niveaux plus élevés d'activité d'impulsions, ce qui se traduit par des coûts énergétiques plus élevés. En revanche, les configurations à faible activité d'impulsions (codage temporel et par saccades) sont plus économes en énergie. Il est important de noter que le schéma de codage entraînable se démarque comme un cas unique, car sa représentation finale dépend fortement de la phase d'entraînement. Introduire un terme de pénalisation des impulsions [Ran+21] dans la fonction de coût pourrait permettre d'obtenir des résultats plus optimisés pour ce schéma de codage en particulier.

4.3.7 Configuration pour les Flux d'Événements

Dans cette section, nous détaillons les paramètres et les conditions de notre étude concernant les flux d'événements.

4.3.7.1 Corruptions Étudiées

Nous concentrons notre étude sur des corruptions courantes qui peuvent se produire sur des caméras événementielles, en les simulant directement sur le tenseur d'impulsions en entrée. Deux types de corruptions sont analysés : le **bruit d'activité de fond** [Fen+20] et le **bruit "hot pixels"** [HLD21]. Il convient de mentionner que des techniques de pré-traitement existent pour atténuer ces corruptions [GD22]. Cependant, notre analyse exclut l'utilisation de tels pré-traitements afin de mesurer la robustesse des modèles face aux entrées "brutes".

Bruit d'Activité de Fond (Background Activity Noise). Cette corruption correspond à l'apparition d'événements parasites qui ne sont pas causés par des changements d'intensité dans la scène [Gal+20]. Une explication du bruit d'activité de fond et de sa simulation en fonction du niveau de sévérité est déjà fournie dans la Section 3.3.4.

Bruit "Hot Pixels". Cette corruption correspond à l'apparition de pixels "défectueux" dans le capteur, ce qui signifie que ces pixels sont constamment actifs et génèrent donc des événements à une fréquence élevée [HLD21]. Dans notre étude, nous simulons ce bruit en sélectionnant aléatoirement un pourcentage des pixels de la caméra pour les marquer comme des "pixels chauds" ou "hot pixels". Ces pixels "chauds" émettent des événements à chaque étape temporelle dans le tenseur d'impulsions $\mathbf{X}_T$. Le niveau de sévérité définit le pourcentage de pixels du capteur qui seront traités comme des "pixels chauds".

4.3.7.2 Base de Données Utilisée

Nous utilisons un sous-ensemble de la base de données N-Caltech101 [Orc+15] pour nos analyses sur les flux d'événements. N-Caltech101 [Orc+15] est une version adaptée de l'ensemble de données d'images statiques Caltech101 [FFP04]. Dans cette version, des flux d'événements ont été capturés en suivant la méthodologie de N-MNIST [Orc+15]. Cette méthodologie implique l'utilisation d'une caméra événementielle en mouvement devant un écran affichant des images. En se référant aux catégories décrites dans la Section 2.3.4, N-Caltech101 peut être considérée comme une base de

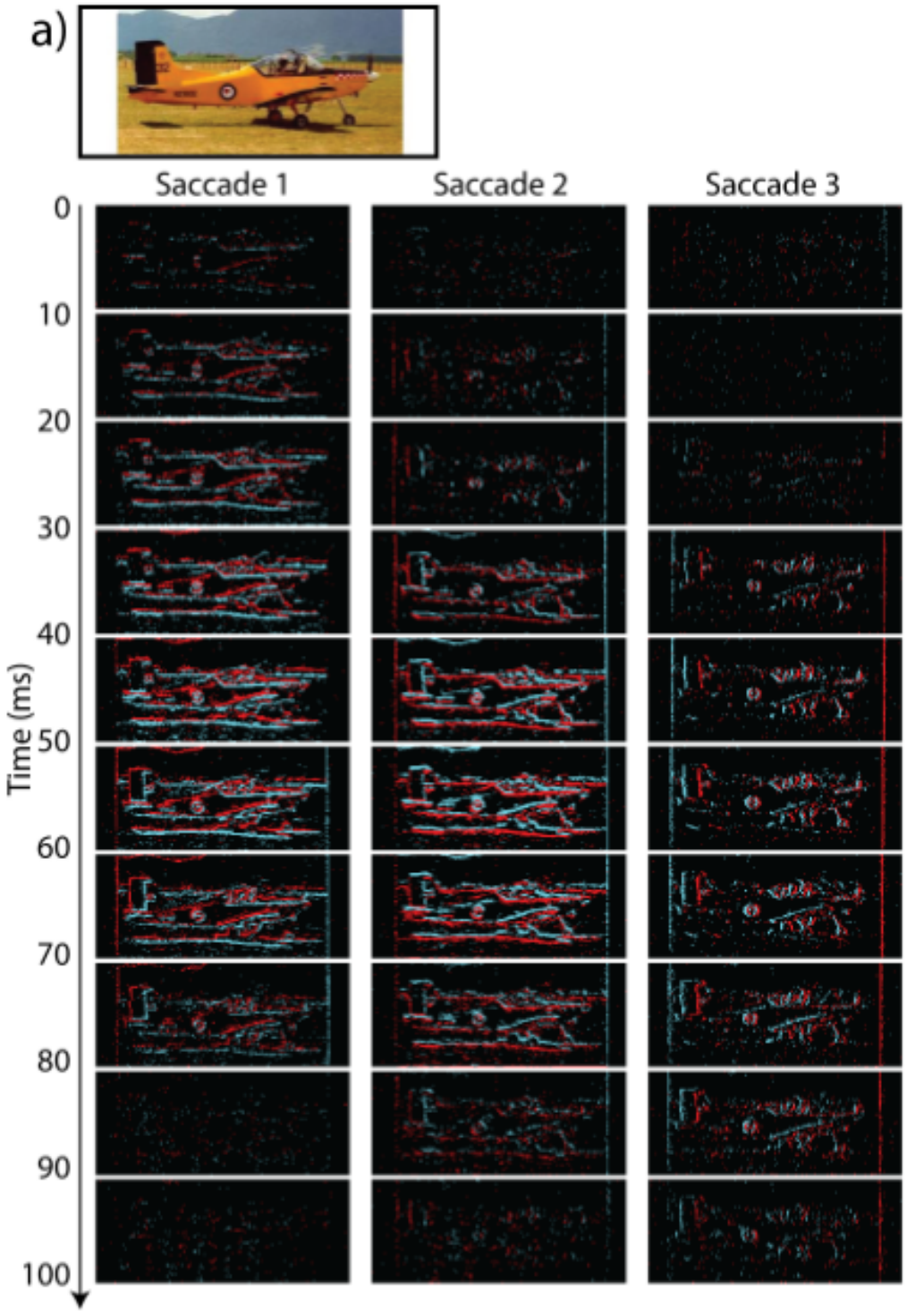


Figure 4.16: Échantillon de N-Caltechc101. Illustration provenant de [Orc+15].

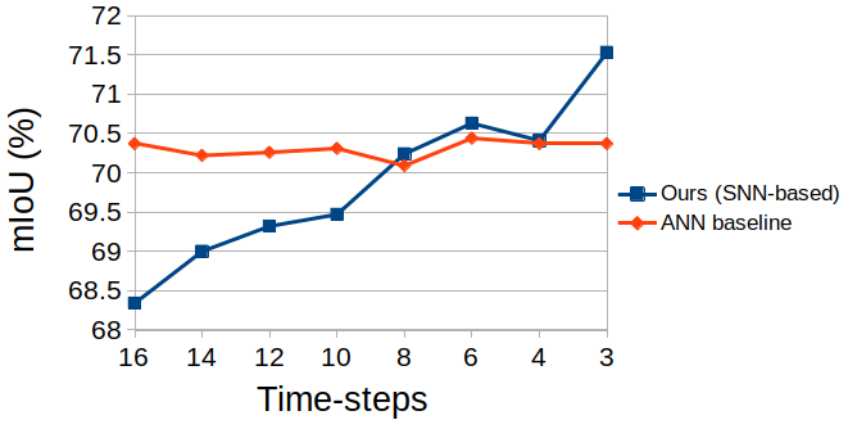


Figure 4.17: Performances de notre approche pour les flux d'événements en fonction du nombre total d'étapes temporelles T.

données caractérisée par **un comportement statique** mais obtenue **par une acquisition simulée**. Un exemple d'échantillon de N-Caltech101 est illustré dans la Figure 4.16.

Plus précisément, nous utilisons un sous-ensemble de N-Caltech101 [Orc+15] composé d'échantillons appartenant à des catégories d'objets ayant au moins 100 représentants. Cette sélection a été effectuée pour atténuer l'impact potentiel du déséquilibre des classes sur l'apprentissage. En conséquence, le sous-ensemble choisi comprend 2035 échantillons pour l'ensemble d'apprentissage et 609 échantillons pour l'ensemble de validation, totalisant ainsi 2644 échantillons.

4.3.8 Résultats pour les Flux d'Événements

Dans cette section, nous présentons les résultats de nos analyses sur les flux d'événements.

4.3.8.1 Analyse de la Latence

La Figure 4.17 présente les performances en fonction du nombre d'étapes temporelles T pour les flux d'événements, et les résultats quantitatifs précis sont répertoriés dans la Table 4.5. De manière intéressante, nous observons de meilleures performances avec des valeurs plus petites de T pour le modèle CSNN, contrairement à l'ANN de référence qui montre des résultats similaires pour toutes les valeurs de T. Plus spécifiquement, notre modèle générique CSNN surpasse l'ANN pour $T \leq 8$. De plus, nous constatons la meilleure performance de 71.53% de mIoU pour $T = 3$, ce qui peut être considéré comme une faible latence (voire "ultra-faible" [Men+22; CRR22]). En réalité, le procédé de capture pour construire N-Caltech101 [Orc+15]

	$T =$ 16	14	12	10	8	6	4	3
Modèle Générique (CSNN)	68.34	69	69.32	69.47	70.24	70.63	70.41	**71.53**
Modèle de référence (ANN)	70.37	70.22	70.26	70.31	70.09	**70.44**	70.37	70.37

Table 4.5: Performances (en % de *mIoU*) obtenues sur N-Caltech101 [Orc+15] par le modèle générique et le modèle de référence pour les flux d'événements.

	Bruit "Hot pixels"	**Bruit d'Activité de Fond**
Modèle de référence ANN	4.92	6.13
Modèle générique CSNN	17.78	8.85

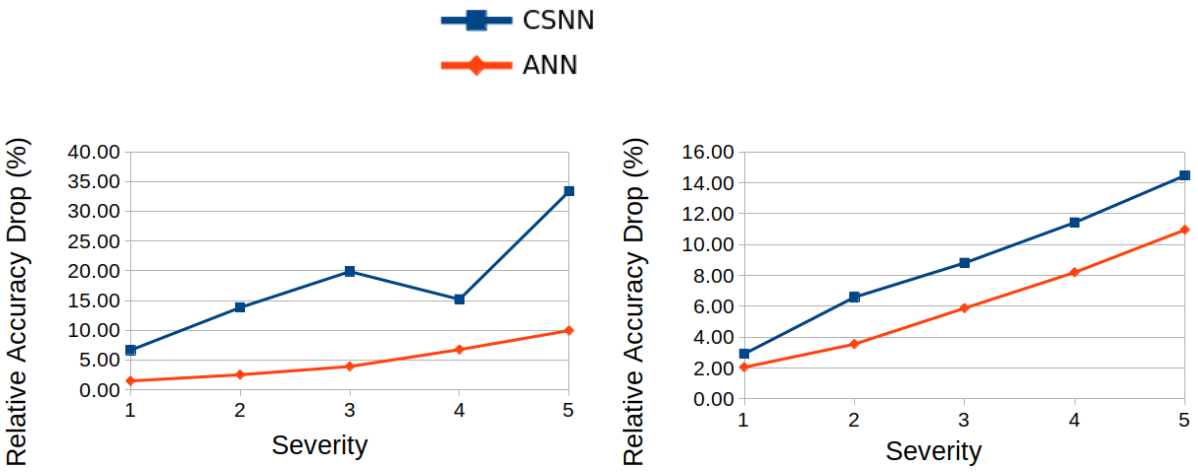
Table 4.6: Scores moyens de baisse de précision relative pour les modèles CSNN et ANN pour le traitement des flux d'événements.

est largement reconnu comme biologiquement plausible. Cependant, lorsqu'il est appliqué à des images statiques dans le cadre d'une "acquisition simulée", peu voire aucune information temporelle pertinente n'est présente dans le flux d'événements résultant [ICL21]. Par conséquent, lorsque la scène est statique, aucune information utile n'est transmise dans le temps. Intégrer les impulsions sur une durée plus courte permet d'éviter de traiter de multiples images événementielles redondantes.

4.3.8.2 Analyse de Robustesse

La Table 4.6 présente en détail les résultats du score moyen de baisse de précision relative pour les modèles ANN et CSNN dans le contexte des flux d'événements. De plus, les évolutions de ce score de baisse de précision relative en fonction du niveau de sévérité sont illustrées dans la Figure 4.18.

Contrairement à nos résultats dans l'analyse de robustesse pour les images statiques, on remarque que le modèle CSNN est plus sensible aux corruptions que ne l'est l'ANN dans le contexte des flux d'événements. En ce qui concerne le bruit d'activité de fond, on observe une légère diminution de la robustesse du CSNN (une différence de 2.72 dans le score moyen de baisse de précision relative), et le score de baisse de précision relative en fonction de la sévérité (voir Figure 4.18) suit une tendance similaire pour les deux modèles. En revanche, la sensibilité du CSNN par rapport au bruit "hot pixels" est nettement plus élevée que celle de l'ANN (une différence de 12.86 dans le score



(a) Bruit "Hot Pixels". **(b) Bruit d'Activité de Fond.**

Figure 4.18: Évolutions des scores de baisse de précision relative en fonction du niveau de sévérité pour les flux d'événements.

moyen de baisse de précision relative). De plus, l'évolution du score en fonction de la sévérité montre que le modèle générique CSNN a une sensibilité aux "hot pixels" qui augmente plus rapidement que celle de l'ANN.

Une telle sensibilité du modèle CSNN aux "hot pixels" peut s'expliquer par la dynamique des neurones à impulsions IF. Ces neurones sont constamment stimulés par les événements constants produits par les pixels "chauds", ce qui entraîne une forte augmentation de l'activité d'impulsions des neurones concernés. Cela interfère avec la capacité initiale du CSNN à extraire des caractéristiques pertinentes, car l'activité accrue due aux "hot pixels" peut dominer les réponses neuronales. En comparaison, les "hot pixels" perturberaient potentiellement le modèle ANN de manière similaire au bruit poivre et sel dans les multiples images d'événements de $\mathbf{X}_T$, mais cette perturbation est moins préjudiciable car les neurones artificiels ne tiennent pas compte des impulsions des étapes temporelles précédentes comme le font les neurones à impulsions.

4.3.8.3 Estimation de Coût Énergétique

En parallèle de l'estimation des coûts énergétiques pour les images statiques, la Table 4.4 présente également le coût énergétique estimé pour les flux d'événements. De même, la Figure 4.15 illustre l'activité d'impulsions des blocs de l'architecture SEW-ResNet-18 [Fan+21a] pour les événements, en parallèle de l'activité d'impulsions pour les images statiques.

En ce qui concerne la consommation énergétique, nous observons que notre modèle générique CSNN atteint une efficacité énergétique similaire avec les flux d'événements qu'avec le codage neuronal le plus énergivore, à savoir le codage entraînable. Cela suggère que les événements capturés au cours d'une séquence sont très nombreux, créant ainsi un tenseur d'impulsions peu épars et entraînant une forte consommation énergétique. Cependant, il est important de noter que N-Caltech101 [Orc+15] est une base de données à comportement statique, où les événements sont générés à partir d'images RGB conventionnelles. Ce type de base de données ne tire pas pleinement parti de la nature éparse des caméras événementielles, ce qui pourrait expliquer pourquoi l'efficacité énergétique du modèle CSNN n'est pas plus élevée par rapport à l'ANN. Néanmoins, notre estimation montre tout de même que le CSNN est 45.48× plus efficace que l'ANN en termes de consommation énergétique. Il est également important de rappeler que, en termes de performances de localisation, le modèle générique CSNN surpasse l'ANN, en particulier à faible latence. De plus, une latence réduite entraîne une consommation moindre, renforçant l'avantage du CSNN pour le traitement des flux d'événements.

De manière intéressante, nous observons la même tendance dans les activités d'impulsions des blocs de l'architecture SEW-ResNet du modèle CSNN que pour les schémas de codage neuronaux. Cela renforce nos observations faites en Section 4.3.6.3, où nous avons constaté que les couches intermédiaires avaient tendance à être plus actives dans notre architecture.

4.3.9 Conclusion de l'Étude

Dans cette étude, nous avons exploré les impacts relatifs de choix de conception fondamentaux sur les CSNNs profonds entraînés avec la BPTT et le substitut du gradient [NMZ19], en utilisant la localisation d'objet comme contexte sensible pour le traitement de l'information spatiale. Plus précisément, nous avons examiné l'effet de la latence temporelle (c'est-à-dire le nombre d'étapes temporelles T), des schémas de codage neuronaux pour le traitement d'images statiques, ainsi que de la robustesse contre les corruptions courantes des capteurs. Pour conduire cette étude, nous avons comparé le modèle générique basé sur un CSNN (décrit en Section 4.1) à une architecture ANN similaire (décrite en Section 4.3.3). Notre évaluation a été menée à la fois sur les flux d'événements et sur des images statiques converties à l'aide de schémas de codages neuronaux.

Supériorité des Faibles Latences. Comme discuté dans la Section 4.3.1, la question de savoir s'il est avantageux d'utiliser un grand nombre d'étapes temporelles pour simuler un SNN était en suspens, à la lumière des résultats de travaux antérieurs [CRR21; Wu+19]. Notre étude soutient la corrélation entre les CSNNs à faible latence et des performances améliorées. Par conséquent, notre étude renforce les intuitions des travaux de l'état de l'art se concentrant sur les CSNNs à faible latence [Men+22; CRR22; CRR21], car ils sont également en mesure de fournir une meilleure efficacité énergétique.

Étude des Codages Neuronaux. En ce qui concerne la comparaison des schémas de codages neuronaux pour les images statiques, nos expériences révèlent que les conclusions tirées des études précédentes sur l'apprentissage hebbien [Guo+21] *ne peuvent pas* être généralisées aux CSNNs entraînés par rétropropagation. Cependant, nous pouvons identifier des orientations pour faire des choix de codages neuronaux appropriés pour les CSNNs optimisés via le substitut du gradient. Premièrement, le **codage entraînable** atteint une précision élevée, même à faible latence, et présente une robustesse notable. Cela est en partie dû à sa capacité d'intégration relativement facile dans un CSNN entraîné par rétropropagation. Cependant, pour garantir une faible consommation d'énergie, il pourrait être nécessaire d'appliquer une régularisation supplémentaire lors de l'entraînement [PZM21]. De plus, ce schéma de codage particulier nécessite un entraînement spécifique sur un ensemble de données particulier, ce qui peut rendre sa fiabilité moins assurée si les données à traiter changent après le déploiement [Gam+14]. Les conclusions relatives au **codage fréquentiel** et au **codage temporel** diffèrent considérablement des études précédentes basées sur la règle STDP [Fal19; Guo+21]. Le codage fréquentiel présente une haute précision, une robustesse significative et une efficacité notable, surtout avec un nombre limité d'étapes temporelles pour l'inférence. D'un autre côté, le codage temporel affiche généralement des performances moins bonnes, mais il se démarque par une robustesse remarquable. Ces différences suggèrent que les conclusions basées sur les règles d'apprentissage bio-plausibles ne peuvent pas être directement généralisées à d'autres stratégies d'entraînement. Le **codage par saccades** se montre le plus sensible aux corruptions en général, mais il démontre également une bonne efficacité énergétique ainsi qu'une précision de localisation adéquate, surtout avec de faibles valeurs de T. Enfin, le **codage par phases** présente des caractéristiques similaires au codage fréquentiel, mais il obtient généralement des résultats légèrement moins performants. Dans l'ensemble, ces observations soulignent l'intérêt du codage fréquentiel en tant que compromis optimal selon les trois aspects étudiés dans nos expérimentations.

Un Réseau de Neurones Impulsionnels Compétitif. Concernant la comparaison avec une architecture ANN similaire, le modèle générique CSNN révèle des performances de précision en localisation supérieures avec les caméras événementielles. De plus, il démontre une possible supériorité en termes de robustesse face aux corruptions des images statiques, en fonction de la technique de codage neuronal utilisée. Enfin, le modèle CSNN présente une nette amélioration en termes d'efficacité énergétique dans les deux contextes, avec des économies d'énergie allant de 44.82× à 126.6×. Ces observations confirment ainsi l'intérêt d'intégrer des mécanismes neuromorphiques pour développer des systèmes de vision basés sur l'apprentissage profond. De plus, ces résultats soulignent l'importance d'utiliser des capteurs visuels neuromorphiques tels que les caméras événementielles en combinaison avec les SNNs, ce qui permet à ces derniers de surpasser partiellement les performances des ANNs.

En conclusion de cette étude, nous croyons que nos résultats peuvent servir de guide pour les futures recherches visant à concevoir des CSNNs entraînés par rétropropagation de manière plus efficace. Les conclusions que nous avons tirées des choix de conception fondamentaux, étayées par nos expérimentations, pourraient être utiles pour orienter les travaux à venir dans ce domaine.

4.4 Spiking-Fer : Reconnaissance d'Expressions Faciales par Approche Neuromorphique

Dans cette section, nous développons une approche nommée "*Spiking-Fer*" pour exécuter la tâche de "**Reconnaissance d'Expressions Faciales**" (*"Facial Expression Recognition"* ou **FER**, en anglais) [Saj+23]. Nous capitalisons sur les résultats de notre étude concernant la localisation d'objets, présentée dans la Section 4.3, en combinant un CSNN avec une caméra événementielle. Par conséquent, nous désignons la tâche de FER, dans ce contexte, sous le nom de "**Reconnaissance d'Expressions Faciales Événementielles**" (ou FER événementielle). Notre approche repose sur le modèle générique CSNN présenté dans la Section 4.2, lequel est utilisé pour classifier les expressions faciales capturées par une caméra événementielle sous la forme d'un flux d'événements.

4.4.1 Contexte

La FER a récemment captivé l'attention en raison de ses multiples applications pratiques dans le domaine de la vision par ordinateur, notamment dans des secteurs tels que la sécurité [Li+21b], la santé [Yun+17] et l'éducation [TO20]. Jusqu'à présent, des modèles d'apprentissage performants basés sur les ANNs ont été élaborés pour la FER [LD20b]. Cependant, ces approches ont souvent négligé les contraintes de consommation d'énergie [Gar+19], mettant l'accent sur la précision. L'alignement entre la nécessité d'accroître la complexité des modèles d'apprentissage [Hu+21] et le besoin d'une infrastructure matérielle à la pointe de la technologie engendre des implications significatives en termes de consommation énergétique.

Dans ce contexte, l'adoption de technologies neuromorphiques comme les SNNs et les caméras événementielles se profile comme une solution adéquate pour atténuer la consommation énergétique inhérente à la FER. De surcroît, cette problématique de vision constitue un cadre expérimental propice à l'exploration de l'utilisation de CSNNs profonds supervisés conjointement avec des caméras événementielles.

Le reste de cette section décrit les travaux entrepris dans le cadre de "Spiking-Fer".

Méthode pour la FER Événementielle. Notre méthode, Spiking-Fer, constitue une adaptation du modèle générique CSNN évoqué dans la Section 4.2, spécifiquement conçue pour accomplir une tâche de classification telle que la FER. Tel qu'exposé dans la Section 4.2.4, l'utilisation d'un modèle élémentaire tel que le modèle générique

offre l'opportunité d'explorer les capacités d'extraction de caractéristiques inhérentes à un CSNN. Dans ce contexte, nous explorons une tâche visuelle (la FER) exigeant l'apprentissage de motifs de mouvement précis, notamment les mouvements des composantes du visage [Li+22a]. Cette particularité diffère grandement du cadre de la localisation d'objet décrit dans la Section 4.3. À l'époque de la conception de notre méthode, Spiking-Fer a marqué une première en tant que modèle dédié à la FER événementielle.

Bases de Données de Référence en FER Événementielle. Les caméras événementielles ont été récemment incorporées dans le domaine de la FER [Ber+23], ce qui limite l'étendue des cas d'utilisation et la disponibilité de bases de données événementielles spécifiques. Toutefois, un ensemble considérable de bases de données pour la FER avec des caméras conventionnelles est disponible [Van+11; Luc+10; Zha+11; Pan+05; All+18], dont certaines se penchent sur des problématiques particulières comme les micro-mouvements du visage [Li+22a] ou les situations d'occlusion [Pou+18]. Afin de satisfaire le besoin en ensembles de données pour l'évaluation de la FER événementielle, nous présentons une série de bases de données spécialement conçues pour cette tâche. Ces bases de données sont obtenues par la conversion de bases de données vidéo classiques [Van+11; Zha+11; Luc+10; Pan+05].

Étude sur les Augmentations de Données. Outre l'analyse du fonctionnement des CSNNs pour la FER événementielle, il s'avère tout aussi essentiel de considérer d'autres éléments intrinsèques à l'entraînement de tout modèle d'apprentissage profond, notamment les stratégies d'augmentations de données événementielles ("*Event Data Augmentations*" [Li+22b] ou EDAs, en anglais). Dans notre étude sur Spiking-Fer, nous approfondissons l'exploration des capacités d'un CSNN à tirer profit de diverses méthodes d'EDAs au cours d'un processus d'apprentissage supervisé pour la FER événementielle.

Estimation du Coût Énergétique. De manière analogue aux Sections 4.3.6.3 et 4.3.8.3, nous entreprenons une évaluation comparative de l'utilisation des SNNs par rapport aux ANNs en quantifiant les gains potentiels en termes de consommation énergétique.

Comparaison avec un Réseau de Neurones Artificiels. De la même manière que dans notre analyse de la localisation d'objets décrite dans la Section 4.3, nous réalisons une comparaison entre le modèle CSNN que nous avons développé et un ANN possédant une architecture et une complexité similaires, en prenant en compte les

aspects examinés (à savoir, la précision de la reconnaissance, l'impact des EDAs sur l'apprentissage, ainsi que la consommation énergétique).

4.4.2 Reconnaissance d'Expressions Faciales avec des Caméras Conventionnelles

Les approches de FER basées sur des caméras conventionnelles peuvent être classées en deux catégories : les méthodes statiques (reposant sur des images individuelles) et les méthodes dynamiques (s'appuyant sur des séquences vidéo). Les méthodes statiques extraient des caractéristiques spatiales à partir d'images fixes. Elles se fondent sur des caractéristiques artisanales [TXC20] ou apprises [LD20a], principalement à l'aide de CNNs [LD20a]. Récemment, les ViTs ont également fait leur apparition [Aou+21]. Les méthodes axées sur des séquences sont mises en œuvre de deux manières distinctes : soit en agrégeant des images, par exemple onset-apex-offset [Yao+18] ou onset-apex [Pou+21], soit en utilisant des images successives pour capturer l'information temporelle. Ces approches tirent principalement parti d'architectures profondes capables de traiter l'information spatio-temporelle, telles que les 3D-CNNs [Pan+19], les réseaux de neurones récurrents (RNNs) [Zha+18b] et les ViTs [Liu+21b].

Récemment, l'élément de mouvement a pris de l'importance et s'est avéré efficace dans le contexte de la FER basée sur des séquences vidéos [LD20a]. De plus, des propositions ont été avancées pour relever divers défis spécifiques aux FER (tels que les occultations [Pou+21] et l'intensité du mouvement [KB22]), en exploitant l'idée que les variations individuelles de mouvement sont plus aptes à la FER que les caractéristiques d'apparence [CN19]. L'amélioration des performances s'obtient en accroissant la complexité des méthodes d'apprentissage, notamment en incorporant une dimension de codage spatio-temporel. Cependant, cette amélioration est généralement associée à une augmentation de la consommation énergétique due à l'utilisation d'architectures ANN de grande envergure.

En tirant parti des capacités des caméras événementielles, reconnues pour leur aptitude à détecter le mouvement à travers les réactions aux variations d'intensité des pixels [Gal+20], Spiking-Fer bénéficie d'un avantage inhérent pour exploiter de manière plus aisée l'importance des variations de mouvement dans le contexte de la FER. De plus, grâce à leur mode de fonctionnement, les neurones impulsionnels d'un SNN possèdent une capacité intrinsèque de traitement spatio-temporel [Sam+23]. Cette particularité renforce l'attrait de l'association de flux d'événements avec un CSNN en

tant que candidat prometteur pour la FER, avec une efficience énergétique accrue.

4.4.3 Reconnaissance d'Expressions Faciales Événementielle

Le domaine de la FER événementielle demeure relativement nouveau, ce qui explique la rareté des études explorant les potentialités des techniques neuromorphiques dans ce cadre. Par conséquent, notre travail portant sur Spiking-Fer se situe dans un domaine encore peu exploré en ce qui concerne les approches analogues et la disponibilité de bases de données événementielles appropriées.

Dans un domaine connexe à la FER, il convient de mentionner des travaux portant sur l'utilisation de caméras événementielles pour l'analyse des visages humains [BPD22; LIB20; Rya+21; SB20], étant donné qu'ils sont centrés sur le même type d'objet capturé que la FER (c'est-à-dire, le visage d'un individu). Face à la pénurie de bases de données événementielles disponibles, une approche proposée par [Rya+21] consiste à convertir une base de données standard à l'aide du simulateur de capteur événementiel ESIM [RGS18]. Cette méthode vise la détection de visages ainsi que le suivi oculaire. En plus des flux d'événements simulés, il existe trois autres ensembles de données capturées par de véritables caméras événementielles et dédiées à l'analyse de visages humains [BPD22; LIB20; SB20]. [SB20] se concentre sur le défi de l'alignement de la pose du visage [JT17] et fournit une base de données de 108 séquences. Dans [LIB20], une base de données événementielles composée de 48 séquences est introduite pour la détection de clignements d'yeux. Se rapprochant davantage de la FER, [BPD22] propose une méthode de reconnaissance de réactions basée sur des flux d'événements capturant les réactions des sujets face à des images de vêtements. La base de données événementielles mise à disposition comprend 455 séquences où les réactions des sujets sont classifiables en 3 catégories : négatif, positif et neutre.

En ce qui concerne spécifiquement la FER événementielle, un travail récent [Ber+23] présente NEFER, une vaste base de données événementielle comprenant 609 séquences, capturées par une caméra Prophesee Evaluation Kit HD affichant une résolution de $(H \times W) = (720 \times 1280)$. De plus, [Ber+23] développe un modèle d'apprentissage basé sur un 3D-CNN afin de proposer des performances de référence sur ce nouvel ensemble de données.

Cependant, au moment où Spiking-Fer a été conçu (travail soumis en avril 2023), la base de données NEFER n'était pas encore disponible, car cette recherche a été

publiée en juin 2023[3]. Bien que NEFER soit la première base de données capturée par une caméra événementielle réelle, elle n'a pas pu être exploitée pour la conception de Spiking-Fer. Par conséquent, afin d'évaluer notre approche sur des flux d'événements, nous avons adopté une démarche similaire à celle de [Rya+21], en utilisant un simulateur de capteur événementiel (en l'occurrence, [HLD21]) pour convertir des bases de données vidéos existantes [Van+11; Zha+11; Luc+10; Pan+05].

D'autre part, même si [Ber+23] présente une base de données événementielle obtenue par **acquisition réelle**, ce qui est préférable pour explorer le domaine de la FER événementielle par rapport aux bases de données converties, le modèle qu'ils utilisent est un ANN, laissant ainsi le domaine des SNNs inexploré. C'est la raison pour laquelle Spiking-Fer demeure la première méthode s'appuyant sur des neurones impulsionnels pour la FER événementielle.

4.4.4 Formulation de la Reconnaissance d'Expressions Faciales

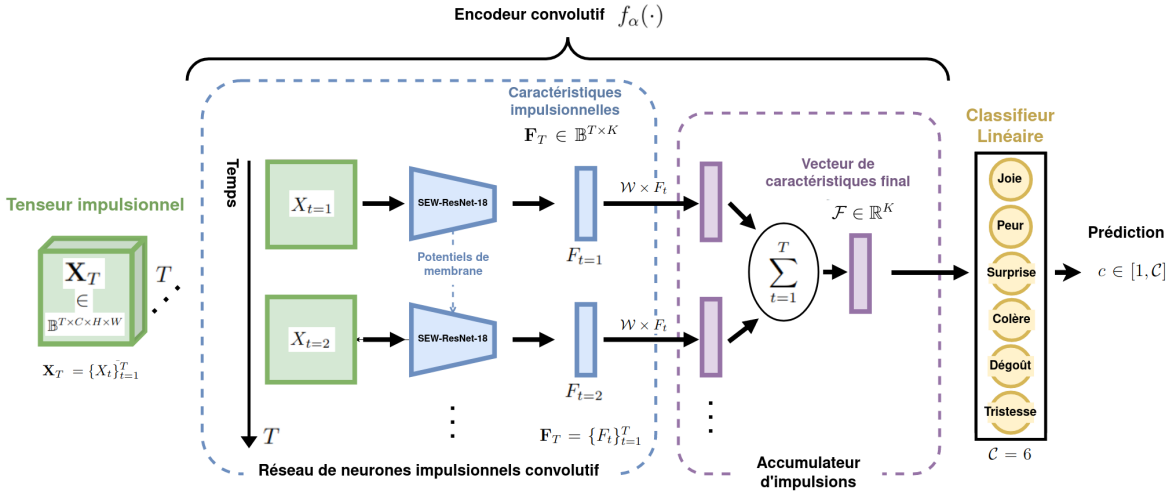
La tâche de FER événementielle visée est définie de la manière suivante : à partir d'un tenseur impulsionnel $\mathbf{X}_T \in \mathbb{B}^{T\times C\times H\times W}$ obtenu à partir de l'enregistrement d'un sujet exprimant une expression faciale à l'aide d'une caméra événementielle de résolution $(H \times W)$, l'objectif consiste à identifier cette expression en anticipant la classe appropriée parmi les $\mathcal{C}$ catégories prédéfinies. Ce tenseur impulsionnel $\mathbf{X}_T$ émane de la discrétisation d'un flux d'événements en images binaires événementielles (voir la Section 2.3.3.1), d'où $C = 2$. Pour aborder cette tâche de FER événementielle, nous mettons en place un modèle $FER(\cdot)$ de la manière suivante :

$$FER(\mathbf{X}_T) = c \tag{4.3}$$

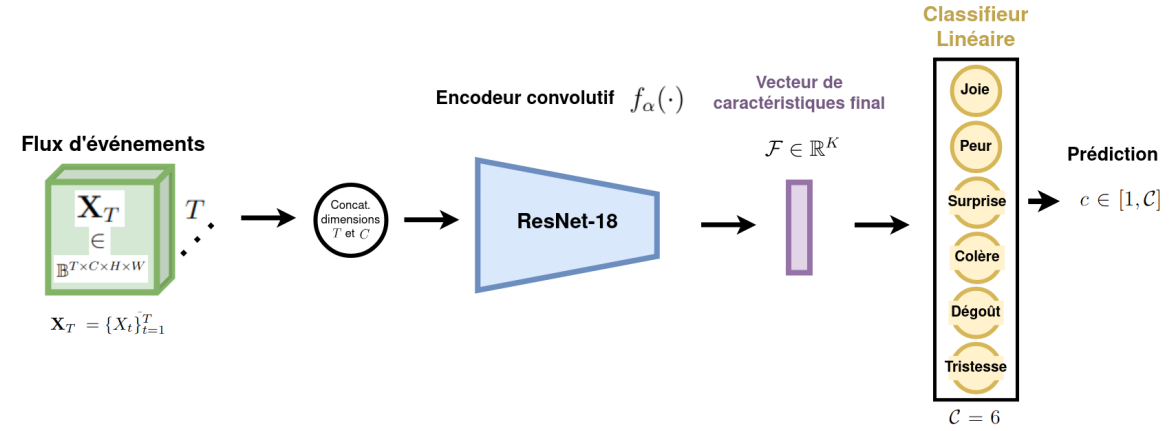
où $c \in [1, \mathcal{C}]$ est la classe attribuée à l'expression faciale capturée.

Dans le contexte de la tâche de FER événementielle que nous visons, nous spécifions un ensemble de $\mathcal{C} = 6$ classes distinctes, à savoir : *Joie*, *Peur*, *Surprise*, *Colère*, *Dégoût*, *Tristesse*. Étant donné que la FER constitue un problème de classification, la métrique d'évaluation employée est la "précision" (ou "taux de reconnaissance"), qui mesure le nombre d'échantillons correctement classifiés dans l'ensemble de validation par rapport au nombre total d'échantillons de cet ensemble.

[3] [Ber+23] a été publié à CVPR 2023.



(a) Modèle Spiking-FER : adaptation du modèle générique présenté en Section 4.2 pour la FER événementielle.



(b) Modèle de référence basé sur un ANN. Un ResNet-18 [He+16] (similaire au SEW-ResNet-18 [Fan+21a] de Spiking-FER) sert d'encodeur convolutif pour extraire les caractéristiques utilisées par le classifieur linéaire. Le tenseur impulsionnel $\mathbf{X}_T$ est utilisé en entrée de cet encodeur convolutif, avec les deux premières dimensions concaténées afin de le rendre compatible avec l'architecture 2D-CNN.

Figure 4.19: Vue d'ensemble des modèles utilisés dans notre travail sur la FER événementielle.

4.4.5 Méthode : Conception du Modèle Spiking-Fer

Spiking-Fer constitue une adaptation du modèle générique basé sur un CSNN, tel qu'expliqué dans la Section 4.2. La Figure 4.19a illustre le modèle Spiking-Fer en reprenant les concepts de la Figure 4.6.

En pratique, la seule modification apportée à Spiking-Fer réside dans l'ajustement de la couche dense linéaire qui sert à générer une prédiction à partir du vecteur de caractéristiques $\mathcal{F}\in\mathbb{R}^K$. Étant donné que la tâche de FER événementielle vise à classifier en $\mathcal{C}=6$ catégories, cette couche dense linéaire, également appelée "classifieur

linéaire", génère 6 valeurs en sortie correspondant aux scores de reconnaissance. Le modèle Spiking-Fer est soumis à un entraînement de bout en bout en utilisant la fonction de coût de cross-entropie.

4.4.6 Méthode : Comparaison avec un Réseau de Neurones Artificiels Similaire

Tout comme dans notre travail relatif à la localisation d'objet, exposé à la Section 4.3.3, nous définissons un modèle de référence ANN, semblable au modèle CSNN, afin de permettre des comparaisons entre les deux types de réseaux de neurones.

La Figure 4.19b présente le modèle ANN de référence. Ce modèle suit la même approche que le modèle ANN utilisé pour la localisation d'objet, décrit dans la Section 4.3.3. En pratique, le modèle ANN emploie directement une architecture 2D-CNN pour extraire le vecteur de caractéristiques $\mathcal{F}$. Afin de garantir une comparaison équitable, l'architecture ANN adoptée est un ResNet-18 [He+16], aligné en termes de nombre de couches (18), de structure et de paramètres ($\approx$ 11M) avec le CSNN SEW-ResNet-18 [Fan+21a] du modèle générique.

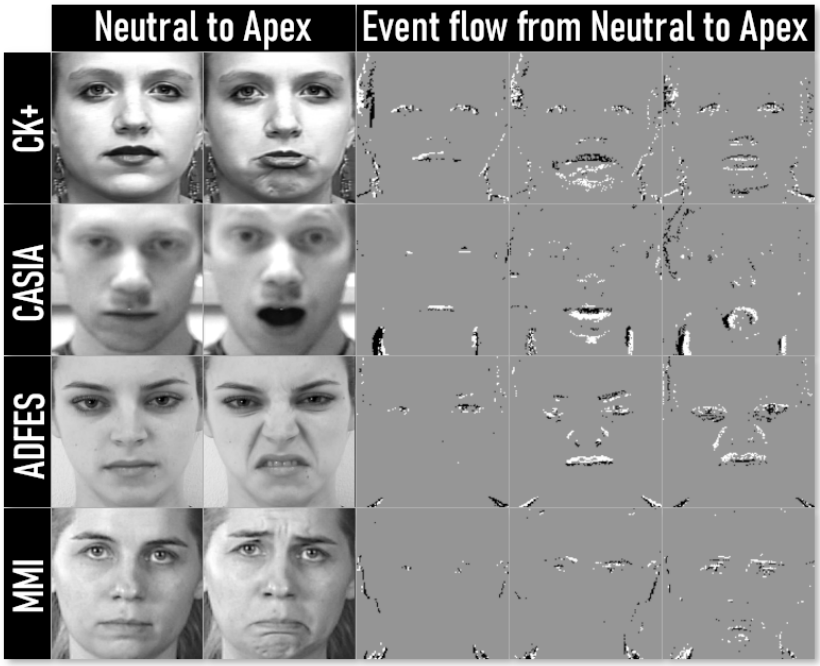
L'encodeur convolutif requiert une entrée au format $(C \times H \times W)$ en raison de son architecture 2D-CNN. Ainsi, le format d'entrée d'un flux d'événements diffère du modèle CSNN. Pour les flux d'événements, nous concaténons les dimensions T et C du tenseur d'impulsions pour obtenir $(TC \times H \times W)$, adapté à un 2D-CNN.

Tout comme Spiking-Fer, le modèle ANN est optimisé de bout en bout en utilisant la fonction de coût de cross-entropie.

4.4.7 Méthode : Création de Bases de Données Événementielles pour la Reconnaissance d'Expressions Faciales

Pour évaluer notre approche de FER événementielle, nous transformons des bases de données reconnues en FER classique : ADFES [Van+11], CASIA [Zha+11], CK+ [Luc+10], et MMI [Pan+05]. Un résumé de ces ensembles de données vidéo est présenté dans la partie inférieure de la Figure 4.20.

Pour accomplir l'étape de conversion, chaque séquence vidéo des bases de données considérées est traitée en deux étapes successives :



	Sequences	Seq. length	Emotion
CK+	374	4 to 71 img	Happiness, Sadness, Anger, Disgust, Fear, Surprise
Oulu -CASIA	480	9 to 72 img	
ADFES	198	180 to 180 img	
MMI	213	11 to 70 img	

Figure 4.20: Résumé des bases de données de FER événementielle obtenues par conversion "vidéo-vers-événements". *En haut :* exemples d'échantillons convertis pour chaque ensemble de données. *En bas :* résumé des bases de données originales [Van+11; Pan+05; Luc+10; Zha+11] qui ont été converties par V2E [HLD21].

1. **Standardisation de la vidéo** [All+22] : Chaque image composant la séquence vidéo subit une normalisation afin de filtrer les éventuelles anomalies susceptibles d'impacter la reconnaissance. Le visage du sujet capturé est extrait et aligné horizontalement par rapport à la caméra en utilisant 68 points de repère faciaux [WJ19]. Ensuite, l'image est transformée en niveaux de gris selon la méthode NTSC [Bro54] (consulter la Section 4.1.4.1) et redimensionnée à une résolution de $(H \times W) = (200 \times 200)$.

2. **Conversion "vidéo-vers-événements" :** on applique le simulateur V2E sur la vidéo standardisée pour obtenir un flux d'événements [HLD21].

Le haut de la Figure 4.20 illustre des exemples des flux d'événements obtenus pour

chaque base de données convertie.

4.4.8 Configuration des Expérimentations

Dans cette section, nous précisons les détails concernant l'organisation de nos expérimentations.

4.4.8.1 Évaluation par Validation Croisée

Les modèles évalués sur un ensemble de données spécifié sont soumis à une configuration de validation croisée à 10 plis : les échantillons sont aléatoirement répartis en 10 plis de taille équivalente. La métrique utilisée pour chaque itération est la précision de classification. En fin de compte, nous présentons la moyenne des scores de précision calculés sur les 10 plis.

4.4.8.2 Détails d'Implémentation

Les expérimentations ont été implémentées en utilisant les bibliothèques PyTorch, Tonic [Len+21] et SpikingJelly [Fan+20]. Elles ont été exécutées sur une carte graphique NVIDIA A40. Nos modèles (Spiking-Fer et le modèle ANN de référence) ont été formés sur 500 époques. Nous avons employé un optimiseur SGD avec un taux d'apprentissage de $0,01$, associé à un planificateur de réduction cosinus [LH16]. Les meilleures performances sur l'ensemble de validation ont été conservées. Nous avons adopté un régime de latence réduite avec $T = 6$. Le code source est accessible sur `https://github.com/Barchid/spiking-fer`.

4.4.9 Expérimentations : Étude sur les Augmentations de Données

Notre étude vise à évaluer l'influence de différentes techniques d'augmentations de données événementielles (EDAs) dans le but d'identifier les combinaisons qui améliorent l'entraînement de nos modèles proposés (Spiking-Fer et le modèle ANN). En outre, à travers nos expérimentations, nous cherchons à quantifier cet impact positif pour chaque ensemble de données événementielles.

Nos expérimentations se divisent en deux phases successives, en fonction de deux ensembles distincts d'EDAs: les EDAs communes et les EDAs spécifiques. Dans la première phase, nous examinons toutes les combinaisons possibles d'EDAs communes afin d'évaluer les impacts généraux de chaque technique. Dans la deuxième

phase, nous réutilisons les meilleures combinaisons d'EDAs communes découvertes précédemment et y ajoutons les EDAs spécifiques pour vérifier si ces transformations spécifiques se révèlent avantageuses.

4.4.9.1 Description des Augmentations de Données Événementielles

Dans ce travail, on définit une EDA comme une fonction $d(\cdot)$ prenant un tenseur impulsionnel $\mathbf{X}_T$ en entrée et retournant une version modifiée de ce tenseur $\mathbf{X}_T^d \in \mathbb{B}^{T\times C\times H\times W}$, tel que :

$$\mathbf{X}_T^d = d(\mathbf{X}_T) \tag{4.4}$$

Les EDAs sont "composables", ce qui signifie qu'une fonction $d(\cdot)$ peut représenter une seule augmentation (comme une rotation aléatoire, par exemple) ou être une composition de plusieurs augmentations successives $d = d_1 \circ d_2 \circ \dots$.

Les EDAs étudiées pour Spiking-Fer sont catégorisées en deux groupes : les EDAs "communes" et les EDAs "spécifiques". Des exemples d'EDAs évaluées sont présentés dans la Figure 4.21.

EDAs Communes. Les EDAs dites "communes" se réfèrent aux distorsions d'un flux d'événements largement utilisées en vision événementielle [Len+21], qui n'ont que peu ou pas de caractéristiques en commun. Les EDAs communes étudiées comprennent les suivantes :

- **Bruit d'activité de fond** (`Noise`) **:** cette augmentation approxime le bruit d'activité de fond généralement présent dans les capteurs événementiels (voir Section 3.3.4). Un pourcentage aléatoire $r_{\text{noise}} \in [0.5, 20]$ d'événements bruités selon une distribution uniforme est ajouté au flux d'événements en entrée.
- **Inversion de polarité** (`PolFlip`) **:** cette augmentation inverse la polarité de tous les événements du flux d'entrée, c'est-à-dire $p_i = -p_i$.
- **Recadrage** (`Crop`) : cette augmentation effectue un recadrage aléatoire avec redimensionnement à un rapport d'aspect aléatoire $r_{\text{crop}} \in [0.08, 1.0]$.
- **Inversion Horizontale** (`HFlip`) **:** cette augmentation réalise une inversion horizontale aléatoire de l'ensemble du tenseur d'impulsions.

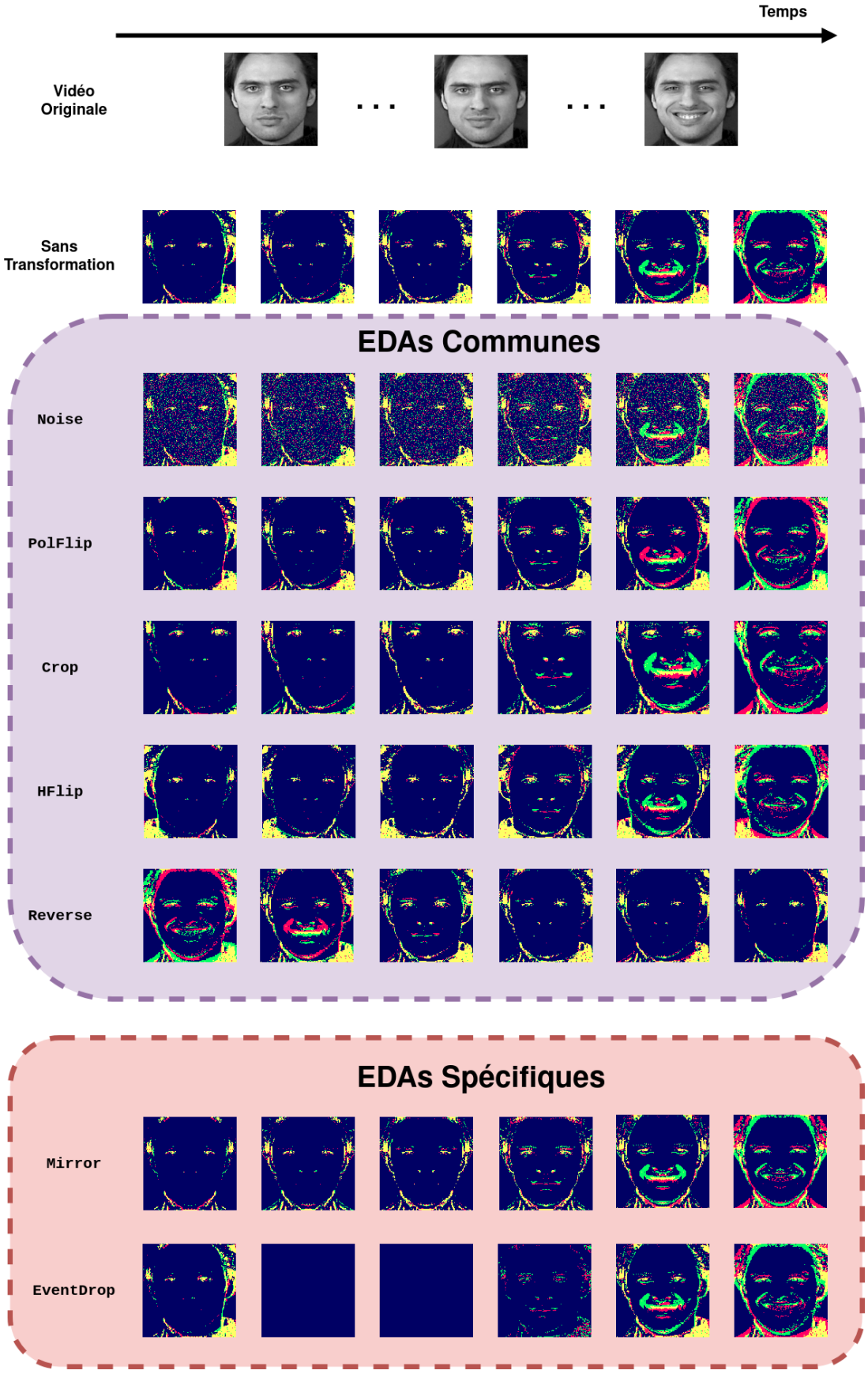


Figure 4.21: Illustrations des EDAs communes et spécifiques étudiées.

Type	Nom	Probabilité
EDA commune	`Noise`	0.5
	`PolFlip`	0.2
	`Crop`	1.0
	`HFlip`	0.5
	`Reverse`	0.2
EDA spécifique	`Mirror`	0.5
	`EventDrop` [Gu+21]	0.75

Table 4.7: Probabilités pour chaque EDA d'être échantillonnée de la distribution D lors d'une itération de la phase d'entraînement afin d'être employée dans la composition finale $d(\cdot)$.

- **Rebobinage** (`Reverse`) **:** cette augmentation inverse l'ordre temporel des événements, transformant ainsi le flux d'événements en une séquence "à l'envers".

EDAs Spécifiques. Les EDAs spécifiques se réfèrent à celles qui répondent à deux critères : **(1)** elles ont un effet de régularisation lors de l'entraînement d'un réseau de neurones lorsque les données d'entraînement sont limitées [DT17]; ou **(2)** elles sont spécifiques à des tâches de vision centrées sur le visage. Pour Spiking-Fer, nous examinons chaque type d'EDA spécifique :

- **Mirroir facial** (`Mirror`) : Cette augmentation consiste à copier la moitié droite (ou gauche) du tenseur impulsionnel sur l'autre moitié. Étant donné que les images de la vidéo d'origine dans une base de données ont été standardisées (voir Section 4.4.7), cette EDA effectue essentiellement un effet miroir de la moitié du visage capturé.
- **EventDrop** [Gu+21] : cette augmentation implique la suppression aléatoire d'événements dans le flux d'événements en fonction de diverses stratégies (spatiale, temporelle, aléatoire).

Utilisation des EDAs pendant l'Entraînement. Lors de l'entraînement d'un modèle sur une base de données, une distribution D d'EDAs à utiliser est spécifiée. À chaque itération de l'entraînement, une composition d'EDAs est créée à partir de celles sélectionnées en effectuant un échantillonnage à partir de D, où chaque EDA a une probabilité d'être choisie pour la composition finale $d(\cdot)$. Les probabilités associées à chaque EDA étudiée sont présentées dans la Table 4.7.

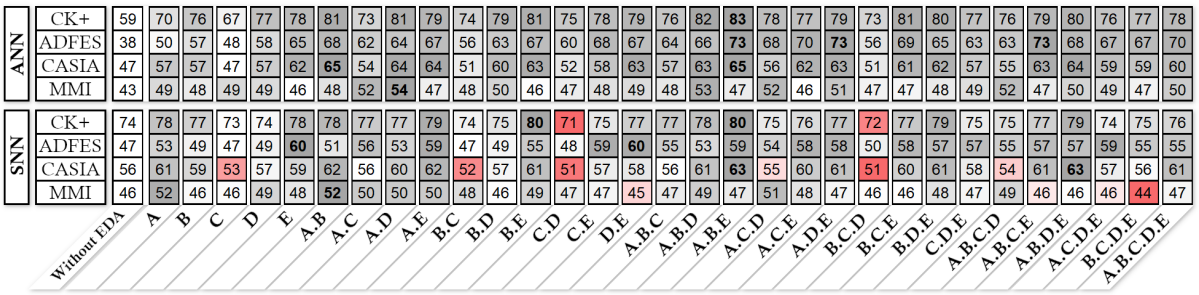


Figure 4.22: Précisions obtenues en fonction de la combinaison d'EDAs communes employée : (A) `HFlip` ; **(B)** `Noise` ; **(C)** `Reverse` ; **(D)** `PolFlip` ; et **(E)** `Crop`.

4.4.9.2 Partie 1 : Transformations Communes

L'évaluation de toutes les combinaisons possibles d'EDAs communes aboutit à un total de 32 expériences pour chaque base de données, et les résultats obtenus sont présentés dans la Figure 4.22.

Les résultats indiquent que le modèle Spiking-Fer présente de meilleures performances que le modèle de référence ANN lorsqu'aucune augmentation n'est appliquée, c'est-à-dire en utilisant uniquement les données d'origine du flux d'événements. Comme souvent observé dans l'entraînement de réseaux de neurones [TN18], l'augmentation de données tend à sensiblement améliorer les performances. Ceci est particulièrement vrai pour la FER, où les bases de données sont souvent restreintes en termes de nombre d'échantillons. Concernant le modèle ANN, nous constatons que toutes les combinaisons d'EDAs ont un impact positif ou neutre, en contraste avec le modèle CSNN, où certaines combinaisons d'EDAs semblent réduire les performances. Parmi les différentes méthodes d'EDAs, la combinaison `Crop, HFlip, Noise` améliore significativement les performances des deux types de réseaux de neurones, sauf pour l'ensemble de données MMI, où l'amélioration est moins prononcée. Cela peut s'expliquer par la plus grande complexité des données de MMI, qui présentent des variations plus importantes en termes de pose de la tête et de motifs de mouvement facial.

Ensuite, nous évaluons les scores de précision observé pour un modèle et une base de données, ce qui génère un total de 320 scores par modèle utilisé et par base de données. Nous menons une analyse de régression multivariée sur cet ensemble de 320 scores, en considérant les EDAs appliquées comme des variables catégorielles indépendantes. Pour chaque EDA, l'analyse de régression fournit une estimation de l'impact attendu sur les performances. Les résultats de l'analyse de régression sont illustrés dans la Figure 4.23 pour chaque jeu de données.

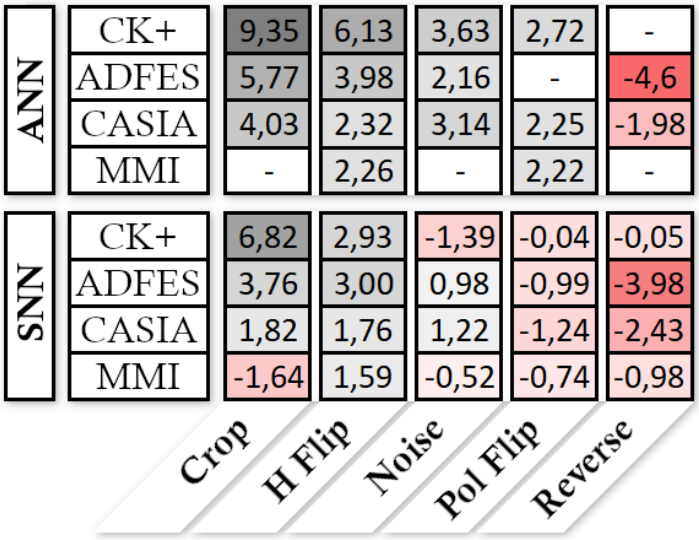


Figure 4.23: Coefficients de régression significatifs (p-value < 0.05) **calculés sur la population des 320 scores de précision**. Ces coefficients correspondent à l'impact de chaque EDA commune sur les différents plis des bases de données.

Selon les coefficients de régression obtenus, les EDAs `Crop` et `HFlip` présentent généralement un impact positif sur les performances, suggérant qu'ils sont bien adaptés à la FER événementielle. Ces EDAs semblent bien couvrir les variations mineures (comme les translations du visage ou les changements de résolution d'image) observées dans différentes bases de données. En revanche, pour `Reverse`, les résultats montrent soit des effets non significatifs, soit des impacts négatifs dans tous les cas. Cela peut s'expliquer par le fait que l'activation d'une expression faciale suit une séquence temporelle dictée par les mouvements musculaires du visage. Dans les bases de données étudiées, les séquences d'expressions faciales vont uniquement de l'état neutre à l'apex [Lio+15], ce qui rend le retournement temporel des événements peu cohérent.

En ce qui concerne l'EDA `PolFlip`, on observe des différences entre Spiking-Fer et le modèle ANN. Alors que Spiking-Fer montre des effets négatifs constants, le modèle ANN bénéficie de l'application de `PolFlip`. Cela suggère que les réseaux de neurones impulsionnels (SNN) ne tirent pas avantage de l'EDA `PolFlip` pour la FER événementielle. Une hypothèse plausible est que l'inversion de polarité des événements perturbe la dynamique des neurones impulsionnels dans un CSNN, limitant la capacité du modèle Spiking-Fer à extraire des caractéristiques pertinentes pour la reconnaissance.

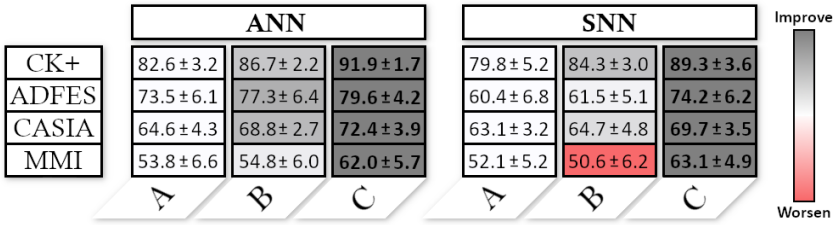


Figure 4.24: Résultats de l'étude sur les EDAs spécifiques. (A) la meilleure combinaison d'EDAs communes (voir la Section 4.4.9.2) ; **(B)** avec l'addition de `EventDrop` [Gu+21] ; et **(C)** avec l'addition de `{EventDrop, Mirror}`.

4.4.9.3 Partie 2 : Transformations Spécifiques

Dans cette partie, nous retenons la meilleure combinaison d'EDAs communes trouvée (c'est-à-dire celle qui atteint le score de précision le plus élevé) pour chaque paire base de données - modèle, et y ajoutons les EDAs spécifiques. Plus précisément, nous testons d'abord l'effet de l'ajout de l'EDA `EventDrop`, suivi de la composition `{EventDrop, Mirror}`. Les résultats obtenus sont présentés dans la Figure 4.24.

En ce qui concerne les performances, nous pouvons constater que la combinaison d'`EventDrop`, qui vise à régulariser l'entraînement des modèles sur des ensembles de données limités, et de `Mirror`, qui modifie l'apparence visuelle du visage d'un sujet, s'avère particulièrement efficace pour améliorer la reconnaissance des expressions faciales, à la fois pour les ANNs et les SNNs. De plus, nous notons que l'écart de performance entre les modèles ANN et CSNN se réduit significativement, en particulier pour la base de données ADFES. Les deux EDAs ont été conçues pour tenir compte des variations interindividuelles, notamment la symétrie du visage et les délais d'activation de l'expression.

En conclusion de cette partie de l'étude, il est possible d'affirmer que l'utilisation d'EDAs spécifiques apporte des bénéfices à l'entraînement, indépendamment de la composition préalable des EDAs communes.

4.4.10 Expérimentations : Estimation de la Consommation Énergétique

Nous menons une analyse de la consommation énergétique sur une puce CMOS de 45 nm [Hor14], similaire à celle présentée dans les Sections 4.3.6.3 et 4.3.8.3. La méthode d'estimation du coût énergétique est détaillée dans l'Annexe A.3.

	E_{ANN} (mJ)	E_{SNN} (mJ)	E_{ANN}/E_{SNN}
DVS-ADFES	1428.67	21.85	65.39
DVS-CASIA		26.22	54.49
DVS-CK+		27.15	52.62
DVS-MMI		30.13	47.42

Table 4.8: Estimation de la consommation d'énergie de Spiking-FER (E_{SNN}) et du modèle ANN de référence (E_{ANN}) selon la méthode présentée en Annexe A.3.

La Table 4.8 affiche les estimations moyennes de l'énergie d'inférence pour chaque ensemble de données. De manière analogue aux expériences sur la localisation d'objet dans les Sections 4.3.6.3 et 4.3.8.3, les résultats indiquent que Spiking-Fer présente une meilleure efficacité énergétique de plusieurs ordres de grandeur (de 47.42× à 65.39× plus efficace que le modèle ANN de référence). Ceci confirme les avantages des CSNNs pour des applications de FER à faible consommation d'énergie sur des dispositifs embarqués.

4.4.11 Conclusion de Spiking-Fer

Dans cette étude, nous avons évalué un modèle CSNN appelé "Spiking-Fer", une adaptation du modèle générique expliqué dans la Section 4.2, pour la tâche de FER événementielle. Cette tâche relativement peu explorée s'est avérée être un cadre approprié pour évaluer l'efficacité de la combinaison entre une caméra événementielle et un CSNN dans le contexte d'une classification où les motifs spatio-temporels jouent un rôle crucial.

Pour surmonter le manque de bases de données dédiées à la FER événementielle, nous avons procédé à la conversion de bases de données vidéos populaires [Van+11; Zha+11; Pan+05; Luc+10] en flux d'événements. Ensuite, nous avons mené des études comparatives en confrontant notre modèle Spiking-Fer à un modèle de référence ANN ayant une architecture similaire.

Tout d'abord, nos expérimentations relatives aux EDAs ont permis de mettre en évidence les techniques d'augmentations de données qui ont un effet positif sur l'entraînement des modèles de FER événementielle. De plus, les résultats obtenus ont mis en lumière les disparités d'impact entre les ANNs et les CSNNs.

Ensuite, notre estimation de la consommation énergétique de Spiking-Fer par rapport au modèle ANN de référence a révélé les avantages significatifs de l'utilisation d'une

caméra événementielle en combinaison avec un SNN, particulièrement lorsque des limitations de consommation d'énergie sont à prendre en compte.

En conclusion finale, nous pensons que notre travail sur Spiking-Fer est un point de départ pour la FER événementielle où nous avons pu d'une part, mettre en contexte l'étude des CSNNs profonds supervisés que nous étudions dans ce Chapitre ; et d'autre part, fournir des informations importantes sur la conception de méthodes d'entraînements pour les futures méthodes de FER événementielle (combinaisons des EDAs, conversions de bases de données, ...).

4.5 Conclusion

Dans ce chapitre, notre étude s'est concentrée sur l'exploration de l'utilisation de CSNNs dans le contexte de la vision artificielle et événementielle, en mettant en œuvre un apprentissage supervisé à l'aide du substitut du gradient [NMZ19]. Nos travaux ont été menés en deux étapes principales : tout d'abord, nous avons élaboré des modèles d'apprentissage profonds basés sur les CSNNs, puis nous avons entrepris des analyses approfondies pour mieux comprendre certains choix de conception fondamentaux et en tirer des conclusions.

4.5.1 Récapitulatif des Contributions

Tout d'abord, nous avons élaboré une preuve de concept (Section 4.1) pour démontrer l'applicabilité des CSNNs profonds dans le traitement de tâches de vision impliquant des scènes complexes et naturelles, en dehors des contextes d'expérimentations classiques (tels que MNIST [LeC+98], CIFAR10 [KH+09], etc.). Cette preuve de concept a validé la faisabilité de notre méthode en se concentrant sur la localisation d'objet dans des images en niveaux de gris, tout en fournissant des informations cruciales sur certains choix de conception nécessaires pour créer un modèle CSNN efficace.

Ensuite, en se basant sur les enseignements de la preuve de concept, nous avons proposé un modèle générique qui intègre un CSNN sous la forme d'un encodeur convolutif conçu pour extraire des caractéristiques à partir de données visuelles en entrée. Ce modèle générique a été développé pour faciliter l'étude des CSNNs entraînés via BPTT et le substitut du gradient. Il se distingue par sa simplicité architecturale et sa flexibilité pour aborder divers types de problèmes, qu'il s'agisse de régression ou de classification.

Nous avons examiné ce modèle générique dans des contextes similaires à celui de la preuve de concept : la localisation d'objet sur des images statiques encodées par des schémas de codages neuronaux, ainsi que sur des flux d'événements. Cette exploration nous a permis de disséquer certains aspects et choix fondamentaux liés aux CSNNs, comme les effets des différents codages neuronaux sur les images statiques, l'influence d'une latence temporelle réduite sur les performances, la robustesse des CSNNs face aux altérations courantes des données d'entrée et la consommation énergétique. En comparant le modèle générique à un modèle ANN de référence, nous avons pu dévoiler des avantages et limites de l'utilisation des CSNNs.

En élargissant notre exploration, nous avons également appliqué notre modèle générique à une nouvelle tâche de vision : la reconnaissance d'expressions faciales événementielle (FER événementielle). En fournissant des ensembles de données pour évaluer cette tâche émergente de vision événementielle [Ber+23], nous avons démontré la capacité des CSNNs à extraire des informations spatio-temporelles pertinentes pour la reconnaissance de micro-mouvements, tels que les expressions faciales. Dans ce contexte novateur, nous avons évalué l'impact des EDAs sur l'apprentissage des CSNNs en termes de performances, mettant en lumière leur rôle dans l'amélioration de l'apprentissage.

4.5.2 Limitations des Approches Proposées

Tout d'abord, nos expérimentations sur la preuve de concept (Section 4.1.4) ont révélé que l'utilisation d'une règle d'apprentissage locale et en ligne (DECOLLE [KMN20]), bien que adaptée au déploiement sur des architectures neuromorphiques [Dav+18], peut poser des défis pour obtenir des prédictions efficaces, car cela exige de résoudre le problème de trouver la prédiction optimale. De plus, les propriétés locales et en ligne ne conviennent pas à de nombreuses tâches de vision, ce qui peut limiter la précision.

En explorant la localisation d'objet et la FER événementielle, nous avons mis en évidence l'avantage (en efficacité énergétique et, parfois, en précision) d'utiliser un CSNN en tant qu'encodeur convolutif pour l'extraction de caractéristiques. Cependant, nous avons également identifié des difficultés dans le modèle générique CSNN, notamment en termes de déploiement sur des architectures neuromorphiques. Ces défis incluent la complexité architecturale, la mise en œuvre de l'accumulateur d'impulsions avec des neurones impulsionnels et la nécessité de réinitialiser les états internes de tous les neurones du réseau à chaque inférence.

En plus des obstacles mentionnés précédemment, une limitation intrinsèque à l'apprentissage supervisé réside dans le besoin de disposer de grandes quantités de données annotées de qualité pour obtenir de bonnes performances [Hes+17; WL20]. Cependant, l'annotation de données peut être extrêmement coûteuse [Kov+16], en particulier dans le domaine émergent de la vision événementielle, où les bases de données annotées sont encore rares. Cette contrainte a suscité un intérêt croissant pour la réduction de la dépendance aux données labellisées dans la recherche [Tan+18; Ran+23; PI23].

Dans notre prochain chapitre, nous concentrerons notre attention sur la vision événementielle, étant donné son potentiel en termes d'efficacité énergétique, en par-

ticulier lorsqu'elle est associée à des CSNNs. Nous chercherons à relever le défi de la faible disponibilité de données annotées en développant une méthode visant à réduire les besoins en données labellisées pour entraîner des encodeurs convolutifs. Cette approche exploitera notamment les avantages observés des EDAs (détaillés dans les Sections 4.4.9.2 et 4.4.9.3) pour créer une méthode d'apprentissage non supervisée. Pour élargir l'impact de cette méthode, nous étudierons également son applicabilité à d'autres types d'encodeurs convolutifs tels que les ANNs utilisant des architectures 2D-CNN ou 3D-CNN.

5

Pré-entraînement par Apprentissage Auto-supervisé pour Réduire le Besoin en Données Événementielles Annotées

Sommaire

Une grande variété de travaux en vision événementielle reposent sur l'emploi de réseaux de neurones profonds entraînés par apprentissage supervisé [Zhe+23]. Bien que cet apprentissage supervisé permet d'atteindre de très bonnes performances sur des tâches complexes (détection d'objets [GS23], segmentation sémantique [KCP21], ...), cela nécessite une grande quantité de données annotées, comme le montrent les bases de données de grande envergure en vision artificielle conventionnelle (par exemple, ImageNet [Den+09] à $\approx$ 1.1M annotations). Or, il existe très peu de bases de données événementielles de cette envergure [Kim+21; De +20], et leur création systématique pour chaque tâche de vision événementielle aurait un coût significatif lié à la captation et l'annotation des flux d'événements.

Dans ce chapitre, notre but est de réduire ce besoin en données annotées pour la vision événementielle. Pour ce faire, nous concevons une approche d'**apprentissage de représentations auto-supervisé** [Eri+22] ("*Self-Supervised Representation Learning*" ou **SSRL**, en anglais) afin de pré-entraîner un modèle sur des flux d'événements sans avoir besoin d'annotations. En conséquence, le réseau de neurones pré-entraîné peut être affiné sur une base de données moins volumineuse. L'approche proposée, basée sur une architecture d'encodage conjoint ("*joint embedding architecture*", en anglais) avec des encodeurs convolutifs, repose notamment sur l'utilisation d'**augmentations de données événementielles** ("*event data augmentations*" ou **EDAs**, en anglais). En plus des EDAs existantes dans l'état de l'art, nous proposons de nouvelles EDAs pour booster les performances de notre approche. Enfin, nous formulons des protocoles d'évaluation standardisés pour comparer les performances des travaux futurs en SSRL événementiel, et effectuons des analyses quantitatives des vecteurs de caractéristiques extraits par les encodeurs convolutifs pré-entraînés.

Notre chapitre est organisé de la manière suivante : premièrement, nous abordons les pratiques existantes en vision événementielle pour mitiger le problème du manque de données annotées afin de positionner notre travail en SSRL événementiel. Deuxièmement, notre approche est présentée, ainsi que les variantes possibles dépendant du type d'encodeur convolutif employé (2D-CNN, 3D-CNN ou CSNN). Troisièmement, nous présentons les EDAs étudiées avec notre méthode pour trouver une combinaison efficace de celles-ci dans le cadre du SSRL événementiel. Ensuite, nous formulons les protocoles d'évaluation destinés à comparer quantitativement notre approche et les travaux futurs similaires. Par la suite, nous présentons les résultats de nos diverses expérimentations basées sur les protocoles établis. Finalement, nous complétons notre étude expérimentale en analysant quantitativement la qualité des représentations ex-

traites par les encodeurs convolutifs pré-entraînés avec notre approche, notamment en employant les mesures d'Uniformité [WI20], de Tolérance [WL21] et d'Alignement de Noyau Centré Linéaire (*"Linear Centered Kernel Alignement"* ou CKA, en anglais).

5.1 Solutions Existantes au Manque de Données Événementielles Annotées

Étant donné la nouveauté relative de la vision événementielle, il existe peu de bases de données publiques pour une problématique de vision donnée, et la plupart de ces bases de données ont un nombre limité d'annotations. Les solutions existantes se basent principalement sur le pré-entraînement d'un réseau de neurones profond, soit en utilisant une grande base de données labélisées ou via l'apprentissage auto-supervisé de représentation.

Pré-entraînement Supervisé. Le pré-entraînement d'un modèle sur des ensembles de données plus importants est une technique courante [Rid+21] pour améliorer les performances des tâches ultérieures avec un nombre limité de données étiquetées. De nombreuses méthodes existent pour obtenir ces larges ensembles de données événementielles requis. Comme expliqué en Section 2.3.4.2, [Orc+15; Li+17; Kim+21] placent une caméra événementielle devant un écran pour capturer des bases de données basées sur des images statiques afin d'obtenir des flux d'événements simulés. De cette manière, il est possible de tirer parti des nombreuses annotations disponibles dans les ensembles de données d'images statiques, mais cela nécessite une configuration matérielle calibrée qui peut être difficile à reproduire en plus d'être différente d'une situation réelle où une caméra événementielle serait utilisée. Les convertisseurs "vidéo-vers-événements" [RGS18; HLD21] peuvent réaliser plus facilement la même simulation de flux d'événements et peuvent, dans une certaine mesure, simuler les conditions réelles bruitées des capteurs événementiels.

Apprentissage Auto-supervisé. Plutôt que de convertir des ensembles de données d'images statiques, on retrouve l'usage de l'apprentissage auto-supervisé dans le cadre de tâches de vision événementielle de bas niveau, telles que le flux optique pour les images + événements [Zhu+18a] ou la reconstruction d'image avec un modèle génératif [PC21]. Plus récemment, des méthodes de SSRL événementiel ont été conçues pour les réseaux de neurones profonds : [Li+22b] étudie l'applicabilité de SimCLR [Che+20] pour les SNNs avec une nouvelle stratégie d'augmentation des données événementielles. [Kle+22] propose une stratégie de modélisation masquée pour le pré-entraînement de ViTs à grande échelle sur des flux d'événements. De même, [YPL23] conçoit une méthode de SSRL événementiel pour les ViTs utilisant à la fois des événements et des images RGB associées. Bien que ces travaux pionniers montrent de

bons résultats en reconnaissance d'objets avec de courts flux d'événements (c'est-à-dire, des bases de données à comportement statique), les tâches de vision à comportement dynamique, telles que la reconnaissance d'actions [Ami+17; Liu+21a], ne sont toujours pas étudiées avec le SSRL événementiel.

Dans ce chapitre, nous optons pour une approche basée sur le SSRL plutôt que sur un pré-entraînement supervisé, car cela permet un pré-entraînement sur des échantillons captés dans le même domaine applicatif que celui de la tâche "cible" ultérieure. En comparaison aux autres applications du SSRL événementiel de l'état de l'art, notre approche est étudiée sur des encodeurs convolutifs légers (2D, 3D et impulsionnel) et montre de bonnes performances sur les bases de données statiques ET dynamiques (reconnaissance d'objets et reconnaissance d'actions, respectivement). De plus, notre approche prouve son efficacité, même en limitant la quantité de données non-étiquetées disponibles pour la phase de pré-entraînement.

5.2 Méthode : Apprentissage Auto-supervisé pour les Événements

Cette section détaille l'approche proposée pour le SSRL événementiel, qui exploite diverses techniques d'augmentation de données. Tout d'abord, nous introduisons des notions préliminaires sur la représentation des événements en entrée. Ensuite, nous détaillons l'architecture d'encodage conjoint qui est optimisée à l'aide de la fonction de coût "VICReg" [BPL22]. Deux variantes de l'architecture sont décrites : la première, appelée "Jumeaux", utilise une structure de réseau similaire à des réseaux de neurones siamois [KZS+15], tandis que la deuxième, appelée "Étudiant-Professeur", adopte une structure asymétrique. Enfin, la section explique les techniques d'augmentation de données étudiées avec notre approche. Bien qu'il existe de nombreuses méthodes d'augmentation de données dans les travaux précédents [Li+22b; Len+21; Gu+21; SZZ22; Nae+22], nous proposons des techniques nouvelles spécialement conçues pour la vision événementielle afin de booster les performances de notre approche.

5.2.1 Notions Préliminaires

Représentation du flux d'événements. Nous partons de la notation des flux d'événements formulée en Sections 2.3.2 et 2.3.3. Dans ce travail, nous employons des séquences d'images événementielles binaires. On définit une séquence de T sous-ensembles d'événements $\{\mathcal{E}_t\}_{t=1}^{T}$ à partir d'un flux d'événements donné $\mathcal{E}$ pour reconstruire une séquence d'images événementielles binaires $\mathbf{X}_T \in \mathbb{B}^{T\times C\times H\times W} = \{X_t\}_{t=1}^{T}$, aussi appelée "tenseur d'impulsions" ou "tenseur impulsionnel" (*"spike tensor"*, en anglais) avec $C = 2$ étant le nombre de canaux correspondant aux deux polarités de la caméra. Les sous-ensembles d'événements $\{\mathcal{E}_t\}_{t=1}^{T}$ sont obtenus en divisant le flux original $\mathcal{E}$ capturé sur un intervalle de temps $\Delta\mathcal{T}$ pour que chaque sous-ensemble ait une durée $\frac{\Delta\mathcal{T}}{T}$.

Formulation d'une EDA. Nous réutilisons la formulation des EDAs décrite en Section 4.4.9.1 : un processus d'EDA est désigné comme une fonction $d(\cdot)$ qui prend un tenseur d'impulsions $\mathbf{X}_T$ en entrée et retourne une version modifiée de ce tenseur $\mathbf{X}_T^d \in \mathbb{B}^{T\times C\times H\times W}$, tel que :

$$\mathbf{X}_T^d = d(\mathbf{X}_T) \tag{5.1}$$

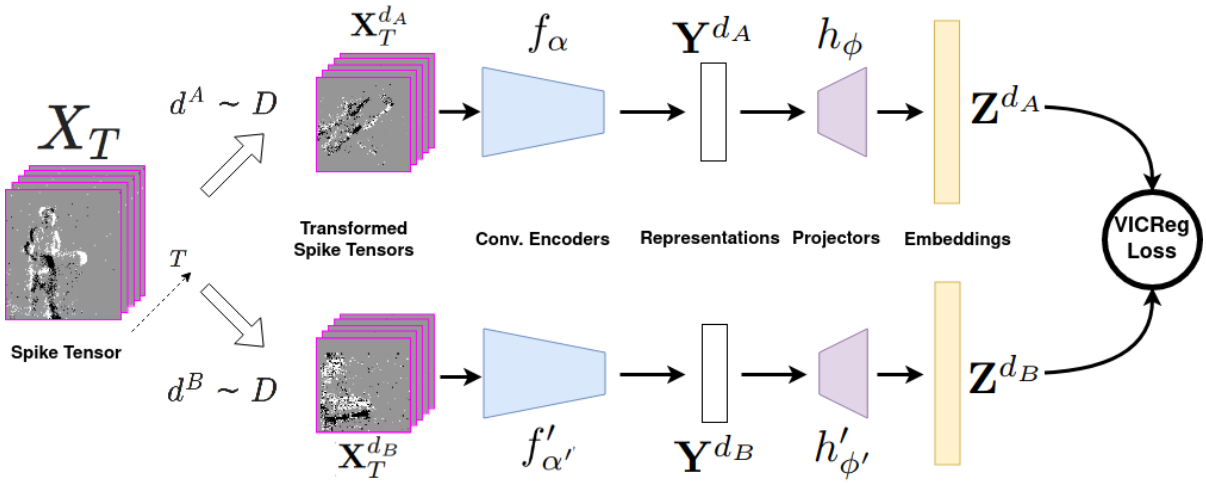


Figure 5.1: Vue d'ensemble de la méthode proposée de SSRL événementiel basé sur une architecture d'encodage coinjoint.

De plus, les fonctions EDA sont considérées comme "**composables**", c'est-à-dire qu'une fonction $d(\cdot)$ peut soit représenter une seule augmentation (par exemple, une translation aléatoire), soit une composition de plusieurs augmentations successives $d = d_1 \circ d_2 \circ \dots$.

5.2.2 Architecture d'Encodage Conjoint

Notre méthode est illustrée dans la Figure 5.1. De manière similaire aux tendances populaires de SSRL [Zbo+21; BPL22], elle est basée sur une architecture d'encodage conjoint, où deux réseaux de neurones (ici, deux encodeurs convolutifs) sont entraînés à inférer des encastrements similaires à partir de deux vues différentes de la même entrée.

Formellement, deux fonctions EDAs d_A et d_B sont échantillonnées à partir d'une distribution D. Soit un tenseur d'impulsion $\mathbf{X}_T$, chaque fonction EDA en crée une vue déformée tel que :

$$\mathbf{X}_T^{d_A} = d_A(\mathbf{X}_T) \tag{5.2}$$
$$\mathbf{X}_T^{d_B} = d_B(\mathbf{X}_T) \tag{5.3}$$

. Ces deux vues sont encodées par deux fonctions $f_\alpha(\cdot)$ et $f'_{\alpha'}(\cdot)$ pour obtenir deux

"**représentations**" $\mathbf{Y}^{d_A}, \mathbf{Y}^{d_B} \in \mathbb{R}^K$ données par :

$$\mathbf{Y}^{d_A} = f_\alpha(\mathbf{X}_T^{d_A}) \tag{5.4}$$

$$\mathbf{Y}^{d_B} = f'_{\alpha'}(\mathbf{X}_T^{d_B}) \tag{5.5}$$

Ces fonctions $f_\alpha(\cdot)$ et $f'_{\alpha'}(\cdot)$ sont des encodeurs de type réseau de neurones convolutifs (ou "**ConvEnc**") dont les paramètres à entraîner sont représentés respectivement par les ensembles α et α'. Il faut mentionner que les représentations sont des vecteurs de caractéristiques extraits sur toute la séquences d'images événementielles et perdent ainsi la dimension temporelle discrétisée en T étapes temporelles.

Enfin, les représentations sont étendues à l'aide de deux "**projecteurs**" ("*projectors*", en anglais) $h_\phi(\cdot)$ et $h'_{\phi'}(\cdot)$ pour produire les deux "**encastrements**" (ou "*embeddings*", en anglais) $\mathbf{Z}^{d_A}, \mathbf{Z}^{d_B} \in \mathbb{R}^{3K}$:

$$\mathbf{Z}^{d_A} = h_\phi(\mathbf{Y}^{d_A}) \tag{5.6}$$

$$\mathbf{Z}^{d_B} = h'_{\phi'}(\mathbf{Y}^{d_B}) \tag{5.7}$$

$$\tag{5.8}$$

De manière similaire aux travaux employant ce type d'architecture [Zbo+21; BPL22], les projecteurs sont des réseaux de neurones artificiels de paramètres ϕ et ϕ' composés de trois couches entièrement connectées (*"fully connected layers"*, en anglais), chacune avec $3K$ neurones. Les deux premières couches sont suivies par une "`BatchNorm`" [IS15] et une fonction d'activation `ReLU` [Aga18]. Comme expliqué dans [BPL22], le rôle d'un projecteur est double :

1. Éliminer l'information par laquelle les deux représentations $\mathbf{Y}^{d_A}$ et $\mathbf{Y}^{d_B}$ diffèrent.
2. Augmenter la dimension (de $\mathbb{R}^K$ à $\mathbb{R}^{3K}$) de manière non linéaire de sorte que la décorrélation des variables des encastrements réduise les dépendances (pas seulement les corrélations) entre les variables des vecteurs de représentation.

Durant la phase de pré-entraînement, les deux encastrements $\mathbf{Z}^{d_A}$ et $\mathbf{Z}^{d_B}$ sont utilisés en lots (*"batches"*, en anglais) pour calculer la fonction de coût VICReg [BPL22]. Cette fonction de coût permet de **(1)** minimiser la distance entre deux encastrements provenant de la même entrée ; **(2)** maintenir la variance de chaque variable d'encastrement sur un lot au-dessus d'un seuil spécifié ; et **(3)** faire tendre

vers zéro la covariance entre les paires de variables d'encastrements sur un lot pour décorréler les variables les unes des autres.

Après ce pré-entraînement, les projecteurs sont abandonnés pour uniquement garder un encodeur dont les poids ont été optimisés sans annotation.

5.2.2.1 Encodeurs Convolutifs Étudiés

Dans notre approche de SSRL événementiel, nous étudions trois types de ConvEncs couramment utilisés en vision événementielle : 2D-CNN, 3D-CNN et CSNN. Pour garantir une comparaison équitable, nous utilisons une architecture légère de type ResNet pour les trois encodeurs, à savoir ResNet-18 [He+16], MC3-ResNet-18 [Tra+17] et SEW-ResNet-18 [Fan+21a] respectivement pour 2D-CNN, 3D-CNN et CSNN. Spécifiquement, ces architectures de type ResNet de 18 couches créent des vecteurs de caractéristiques (c'est-à-dire, des représentations) à 512 canaux, donc $K = 512$. Pour prendre en compte ces différents types de ConvEncs, des clarifications sur le modèle ou des modifications mineures de la formulation originale fournies dans la Section 5.2.2 sont nécessaires.

2D-CNN. Étant donné que l'entrée de la méthode proposée est une séquence d'images événementielles, nous rencontrons la même problématique que celle expliquée dans les Sections 3.2 et 4.3.3 : un 2D-CNN ne peut pas traiter des séquences d'images telles quelles, car il ne peut pas prendre en compte la dimension temporelle. C'est pourquoi nous appliquons sur le tenseur impulsionnel la même modification que celle décrite en Sections 3.2 et 4.3.3, à savoir une concaténation des dimensions de temps et du nombre de canaux de l'image impulsionnelle. Par conséquent, le tenseur impulsionnel en entrée prend la forme $\mathbf{X}_T \in \mathbb{R}^{TC \times H \times W}$ lorsqu'il est traité par un encodeur convolutif de type 2D-CNN.

3D-CNN. Un encodeur convolutif de type 3D-CNN prend en entrée une séquence d'images, et retourne un vecteur de caractéristiques pour toute la séquence, abandonnant donc la dimension temporelle. Par conséquent, l'utilisation de ce type d'encodeur dans notre méthode est directe car ne nécessite pas de modification spécifique.

CSNN. Nous employons le même modèle CSNN générique que celui proposé en Section 4.2 en raison des qualités observées lors des expérimentations du Chapitre 4. Premièrement, l'emploi d'une architecture profonde de neurones impulsionnels IF

optimisés par le substitut du gradient permet d'obtenir un CSNN performant sur des flux d'événements complexes tout en étant comparable aux autres types d'encodeurs convolutifs étudiés. Deuxièmement, il a été montré que l'emploi d'un accumulateur d'impulsions pour obtenir une représentation $\mathbf{Y}^d$ (autrement dit, un vecteur de caractéristiques) composée de valeurs réelles ($\in \mathbb{R}$) est efficace au vu des performances observées sur la localisation d'objet (voir Section 4.3), une tâche qui nécessite de prédire précisément des valeurs réelles sous forme de coordonnées d'une boite englobante.

5.2.2.2 Variantes du Modèle

Nous définissons deux variantes de l'architecture d'encodage conjoint utilisée dans notre approche de SSRL événementiel.

1. **Jumeaux :** cette variante désigne la conception standard de l'architecture d'encodage conjoint, où les deux encodeurs convolutifs, $f_\alpha(\cdot)$ et $f'_{\alpha'}(\cdot)$, sont des modèles identiques dont les poids sont partagés, c'est-à-dire $\alpha = \alpha'$.

2. **Étudiant-Professeur :** les neurones impulsionnels dans un CSNN n'échangent que des valeurs binaires (c'est-à-dire, des impulsions), qui sont moins expressives que les valeurs réelles traitées par les 2D/3D-CNN. Pour résoudre ce problème, nous concevons une structure de réseaux conjoints asymétriques sans poids partagés (c'est-à-dire, $\alpha \neq \alpha'$). La première branche $f_\alpha(\cdot)$ de l'architecture est un CSNN (l'"Étudiant"). La deuxième branche $f'_{\alpha'}(\cdot)$ est un 2D/3D CNN (le "Professeur") dont l'objectif est d'améliorer l'apprentissage du CSNN.

5.2.3 Augmentations de Données Événementielles

La définition d'une distribution efficace D d'EDAs est essentielle pour un pré-entraînement efficace dans le cadre de notre approche SSRL. Dans cette section, nous décrivons les EDAs étudiées issues de travaux antérieurs avec leurs paramètres aléatoires associés. De plus, nous proposons de nouvelles transformations pour élargir le choix des EDAs et potentiellement améliorer le pré-entraînement de notre méthode. Les EDAs étudiées sont réparties en 3 groupes : les augmentations communes, les augmentations en découpage et les augmentations géométriques. Des visualisations des EDAs mentionnées sont disponibles dans les Figures 5.2, 5.3, et 5.4.

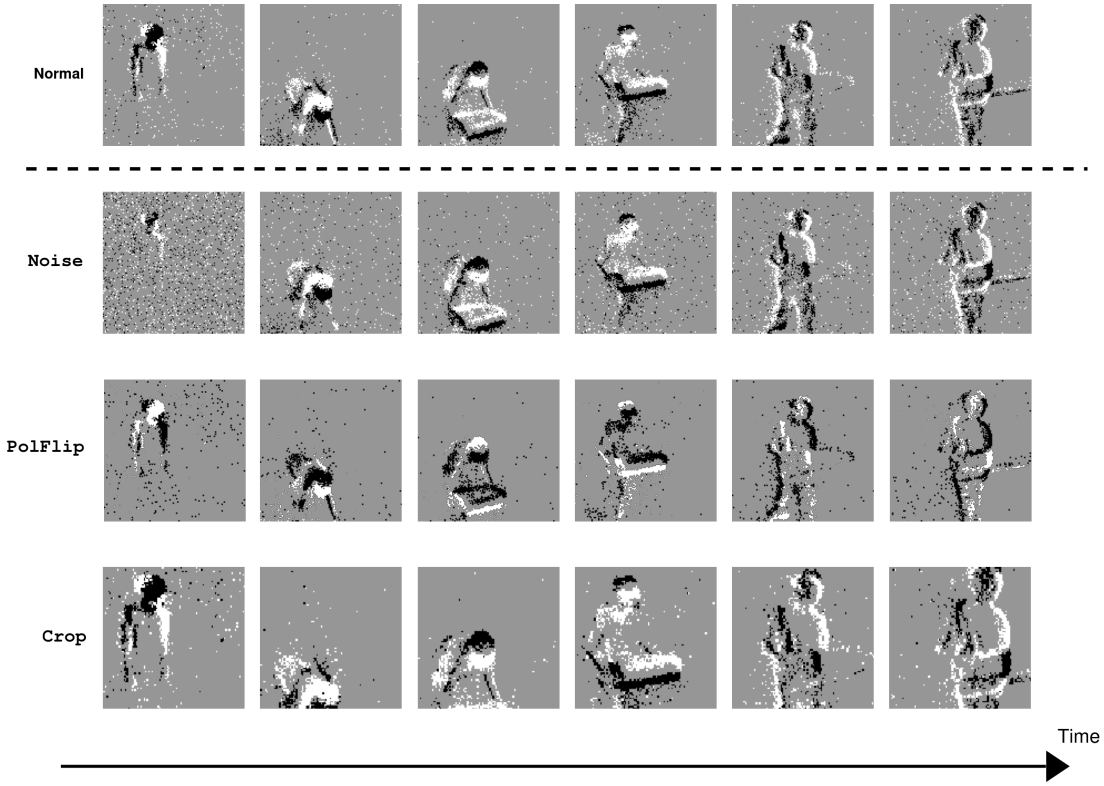


Figure 5.2: Visualisation des EDAs communes étudiées.

5.2.3.1 Augmentations Communes

Ce terme fait référence à un ensemble de distorsions couramment utilisées dans la vision événementielle qui ne partagent aucune caractéristique spécifique entre elles. Les EDAs communes décrites ici sont les mêmes que celles expliquées en Section 4.4.9.1. Toutes ces EDAs sont disponibles dans la bibliothèque Tonic [Len+21].

- **Bruit d'activité de fond** (`Noise`): approximation d'un bruit d'activité de fond présent dans les capteurs événementiels (voir Section 3.3.4). Un pourcentage aléatoire $r_{\text{noise}} \in [0.5, 20]$ d'événements bruités selon une distribution uniforme est ajouté au flux d'événements en entrée.

- **Inversion de polarité** (`PolFlip`): inverse la polarité de tous les événements du flux d'entrée, c'est-à-dire $p_i = -p_i$.

- **Recadrage** (`Crop`): recadrage aléatoire avec redimensionnement à un rapport d'aspect aléatoire $r_{\text{crop}} \in [0.08, 1.0]$.

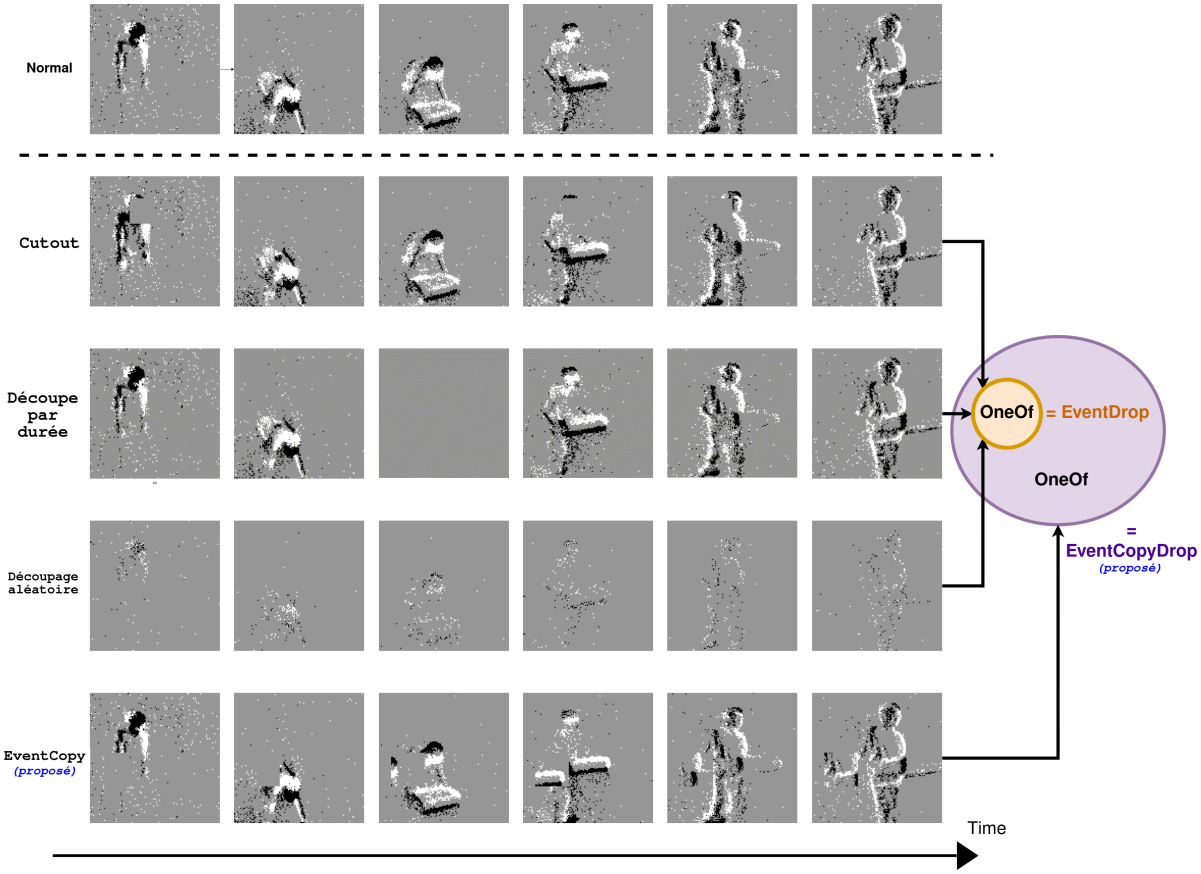


Figure 5.3: Illustration des EDAs en découpage étudiées.

5.2.3.2 Augmentations en Découpage

Les EDAs en découpage sont des augmentations de flux événementiels où des événements sont supprimés du tenseur d'impulsions en entrée. Plusieurs primitives de ce type d'EDA sont définies dans [Gu+21] :

- **Découpe par zone** (`Cutout`): supprime les événements se trouvant dans une boîte englobante de dimensions $(r_{\text{cut}}H \times r_{\text{cut}}W)$, avec le paramètre $r_{\text{cut}} \in [0.05, 0.3]$.
- **Découpe par durée**: supprime tous les événements du flux d'événements en entrée $\mathcal{E}$ situés dans un intervalle de temps aléatoire de durée $r_{\text{time}} \times \Delta_{\mathcal{T}}$ où $r_{\text{time}} \in [0.1, 0.9]$.
- **Découpage aléatoire**: chaque événement a une probabilité $r_{\text{drop}} \in [0.1, 0.9]$ d'être supprimé du flux d'événement original $\mathcal{E}$.

La transformation nommée "**EventDrop**" [Gu+21] utilise ces primitives via une relation `OneOf`, c'est-à-dire qu'une de ces trois primitives d'EDA en découpage est

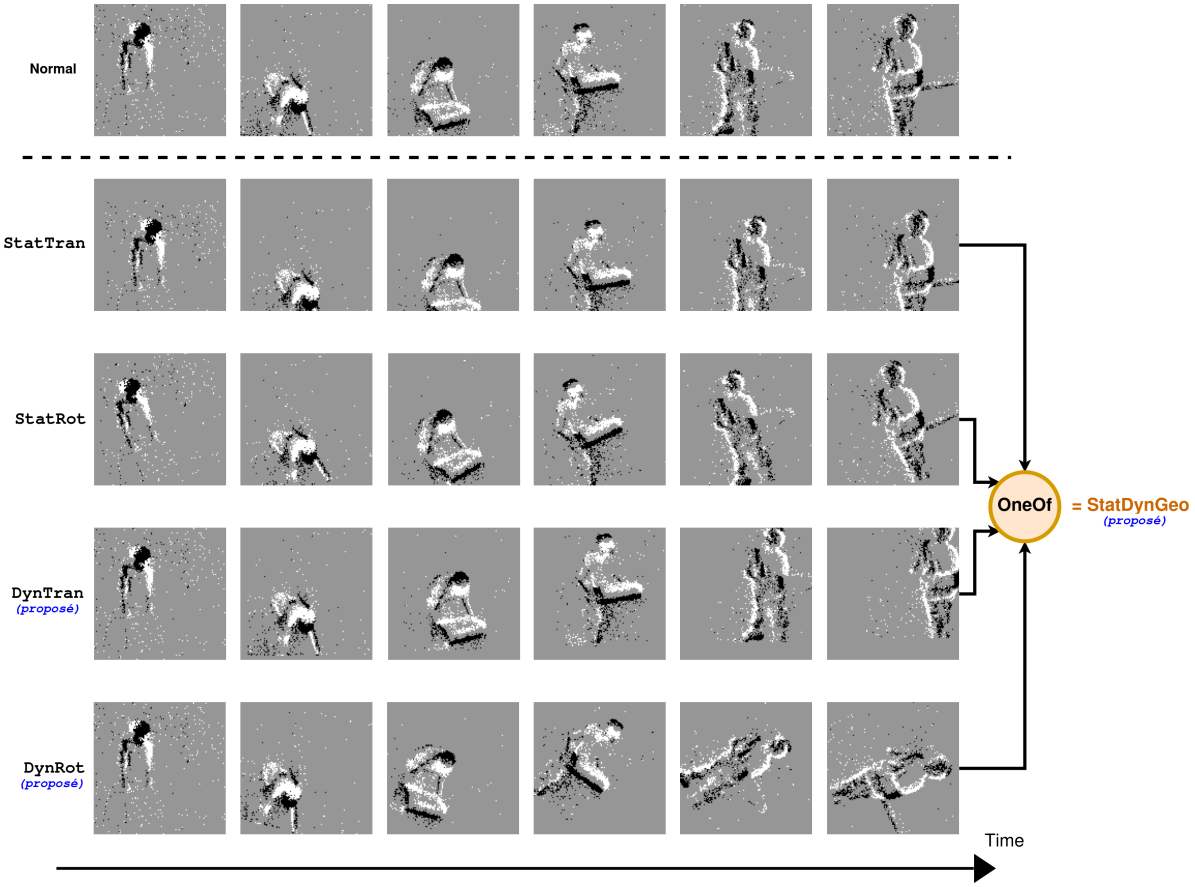


Figure 5.4: Illustration des EDAs géométriques étudiées.

sélectionnée de manière aléatoire (à probabilité égale) afin d'être appliquée sur le flux d'événement en entrée. De cette manière, il est possible d'employer plusieurs augmentations en découpage différentes pendant la phase d'apprentissage, sans que les primitives ne se superposent sur le même tenseur d'impulsion en entrée.

En plus de ces trois primitives, nous en proposons une quatrième, nommée "**EventCopy**": les événements situés dans une boîte englobante aléatoire de dimensions $(r_{\text{copy}}H \times r_{\text{copy}}W)$ sont *copiés* à un autre emplacement dans le tenseur d'impulsion, avec le paramètre $r_{\text{copy}} \in [0.05, 0.3]$.

Enfin, nous étendons l'idée de EventDrop [Gu+21] en proposant une EDA nommée "**EventCopyDrop**", qui est essentiellement une relation `OneOf` employant les 4 primitives en découpages abordées ici.

5.2.3.3 Augmentations Géométriques

Les EDAs géométriques désignent les distortions qui transforment un flux d'événements dans sa dimension spatiale. Ce sont essentiellement des adaptations d'augmentations géométriques communes employées en traitement d'images conventionnel [Li+22b].

- **Translation Statique** (`StatTran`) : applique une translation sur tout le tenseur d'impulsions, avec un déplacement vertical de $r_y \in [\pm 0.2 \times H]$ et un déplacement horizontal de $r_x \in [\pm 0.2 \times W]$.
- **Rotation Statique** (`StatRot`): applique une rotation de tout le tenseur d'impulsions d'un angle aléatoire $r_{degrees} \in [-75, 75]$.

De plus, nous proposons des nouvelles augmentations géométriques qui ont la particularité de transformer le flux d'événements progressivement dans le temps :

- **Translation Dynamique** (`DynTran`): applique une translation *progressive dans le temps* de 0 à $r_y \in [\pm 0.2 \times H]$ verticalement, et de 0 à $r_x \in [\pm 0.2 \times W]$ horizontalement le long de l'axe temporel.
- **Rotation Dynamique** (`DynRot`): applique *progressivement* une rotation du tenseur d'impulsions de 0 à un angle aléatoire $r_{degrees} \in [-75, 75]$ le long de l'axe temporel.
- **StatDynGeo**: applique une relation `OneOf` des quatre EDAs géométriques mentionnées : [StatTran, StatRot, DynTran, DynRot].

5.2.4 Stratégie de Composition des Augmentations Événementielles

À chaque échantillonage de la distribution D (c'est-à-dire, quand on applique l'étape d'augmentation de données du flux d'événements lors d'une itération d'entraînement), chaque EDA sélectionnée a une probabilité donnée d'être employée dans la composition de la fonction d'EDA d_A ou d_B. La Table 5.1 résume les scores de probabilités pour chaque EDA étudiée, ainsi que les paramètres aléatoires associés.

Type	Name	Probability	Parameters
Communes	`Noise`	0.5	$r_{\text{noise}} \in [0.5, 20]$
	`PolFlip`	0.2	-
	`Crop`	1.0	$r_{\text{crop}} \in [0.08, 1.0]$
Géométriques	`StatTran`	0.5	$r_y \in [\pm 0.2 \times H]$, $r_x \in [\pm 0.2 \times W]$
	`StatRot`	0.5	$r_{\text{degrees}} \in [-75, 75]$
	`DynTran` (Contrib.)	0.5	$r_y \in [\pm 0.2 \times H]$, $r_x \in [\pm 0.2 \times W]$
	`DynRot` (Contrib.)	0.5	$r_{\text{degrees}} \in [-75, 75]$
	`StatDynGeo` (Contrib.)	0.8	$r_y \in [\pm 0.2 \times H]$, $r_x \in [\pm 0.2 \times W]$, $r_{\text{degrees}} \in [-75, 75]$
en Découpage	`Cutout`	0.3	$r_{\text{cut}} \in [0.05, 0.3]$
	`EventCopy` (Contrib.)	0.5	$r_{\text{copy}} \in [0.05, 0.3]$
	`EventDrop`	0.75	$r_{\text{cut}} \in [0.05, 0.3]$, $r_{\text{time}} \in [0.1, 0.9]$, $r_{\text{drop}} \in [0.1, 0.9]$
	`EventCopyDrop` (Contrib.)	0.8	$r_{\text{cut}} \in [0.05, 0.3]$, $r_{\text{time}} \in [0.1, 0.9]$, $r_{\text{drop}} \in [0.1, 0.9]$, $r_{\text{copy}} \in [0.05, 0.3]$

Table 5.1: Récapitulatif des EDAs étudiées, incluant leur probabilité dans la distribution D et leurs paramètres aléatoires.

5.3 Méthode : Conception de Protocoles d'Évaluations de Performance

Étant donné la nouveauté du SSRL événementiel, les récents travaux dans ce domaine [Kle+22; YPL23; Li+22b] évaluent les performances de leurs approches proposées selon des expérimentations différentes et donc ne formulent pas explicitement de méthodologie commune pour une comparaison équitable.

Pour répondre à ce problème, nous présentons, dans cette section, des protocoles d'évaluation standards, basés sur des bases de données événementielles populaires afin de fournir un cadre d'expérimentations pour les travaux futurs dans le domaine.

La méthodologie d'évaluation proposée est composée de trois protocoles différents basés sur des tâches de **classification** de flux d'événements. Chaque protocole a pour objectif d'évaluer un aspect spécifique recherché en SSRL. Comme ces évaluations se basent sur des tâches de classification, la métrique observée pour tous les protocoles est le **taux de précision** sur l'ensemble de validation de la base de données étudiée. La Figure 5.5 illustre un résumé des étapes à suivre pour chacun des trois protocoles d'évaluation proposés.

5.3.1 Bases de Données Utilisées

Notre approche diffère des travaux antérieurs [Kle+22; YPL23] en évaluant la capacité des méthodes à être pré-entrainées avec des données limitées, en mettant l'accent sur une convergence plus rapide sans dépendre de bases de données à grande échelle telles que N-ImageNet [Kim+21]. Nous sélectionnons les deux types de bases de données introduites en Section 2.3.4 : celles à **comportement statique** et celles à **comportement dynamique**. En comparaison, les travaux similaires [Kle+22; YPL23] se concentrent uniquement sur des bases de données statiques. La Table 5.2 reprend les informations pertinentes des ensembles de données évalués.

5.3.2 Protocole d'Évaluation Linéaire

Un classifieur linéaire est entraîné sur les représentations fixes obtenues à partir du pré-entrainement d'un des ConvEncs avec notre méthode en utilisant l'ensemble d'apprentissage d'une base de données événementielles. Le score de précision obtenu sur l'ensemble de validation est rapporté. L'objectif de ce protocole d'évaluation

Base de Données	Résolution	Durée	#Échantillons			Classes
			Total	Entraînement	Validation	
DVSGesture [Ami+17]	128 × 128	±6s	1342	1078	264	11
DailyAction-DVS [Liu+21a]	346 × 260	±5s	1440	1152	288	12
ASL-DVS [Bi+20]	240 × 180	±0.1s	100800	80640	20160	24
N-Cars [Sir+18]	304 × 240	±0.1s	24029	15422	8607	2
NCaltech-101 [Orc+15]	Variable	±0.3s	8709	7838	871	101

Table 5.2: Récapitulatif des bases de données étudiées.

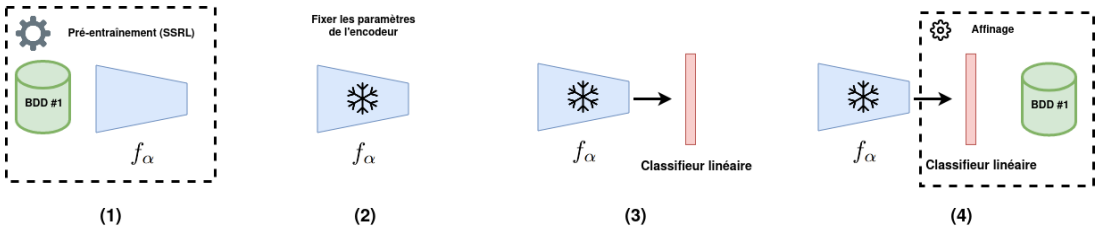
est de démontrer la capacité d'une méthode de SSRL événementiel à extraire des caractéristiques pertinentes. Trois bases de données de tailles différentes sont étudiées : DVSGesture [Ami+17], N-Caltech101 [Orc+15] et ASL-DVS [Bi+20]. La Figure 5.5a montre un résumé des étapes à suivre pour ce protocole.

5.3.3 Apprentissage Semi-supervisé

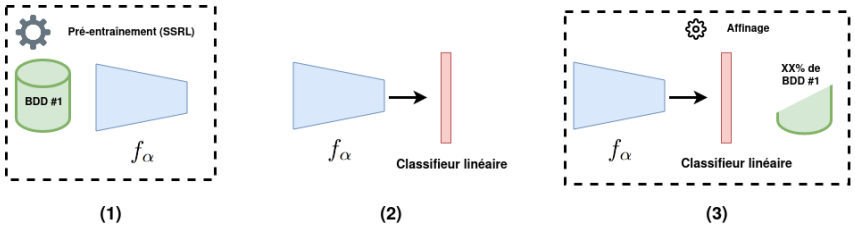
L'objectif de ce protocole est d'évaluer la capacité d'une approche SSRL donnée à réduire le besoin en données annotées. Après un pré-entraînement par SSRL sur une base de données définie, nous affinons l'encodeur convolutif en utilisant un sous-ensemble annoté du même ensemble de données (c'est-à-dire, en utilisant un certain pourcentage de l'ensemble d'apprentissage). Le score de précision obtenu sur l'ensemble de validation est rapporté. Les évaluations sont menées sur DVSGesture [Ami+17], N-Caltech101 [Orc+15] et ASL-DVS [Bi+20]. La Figure 5.5b illustre les étapes à suivre pour ce protocole.

5.3.4 Protocole de Transfert d'Apprentissage

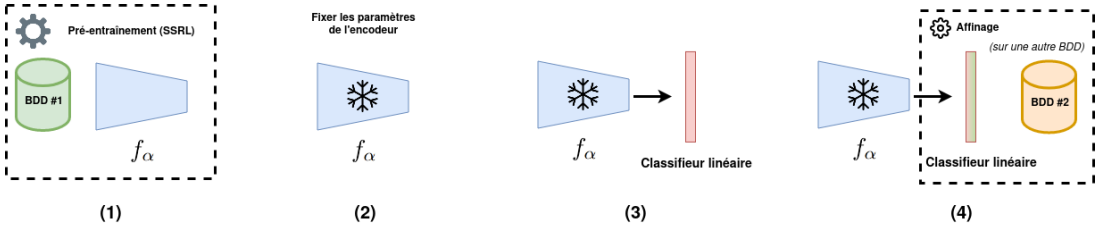
Ce protocole d'évaluation vise à évaluer la capacité de transférer les caractéristiques apprises d'un ensemble de données à un autre. Plus précisément, nous évaluons les performances de transfert d'apprentissage d'un encodeur convolutif pré-entraîné en affinant un classifieur linéaire sur une autre base de données tout en maintenant les paramètres de l'encodeur convolutif fixes. Nous rapportons le taux de précision sur l'ensemble de validation. Nous définissons deux scénarios de transfert d'apprentissage : DVSGesture [Ami+17] (pré-entraînement) vers DailyAction-DVS [Liu+21a] pour les bases de données à comportement dynamique, et ASL-DVS [Bi+20] (pré-entraînement) vers N-Cars [Sir+18] pour les bases de données statiques. La Figure 5.5c montre les étapes de ce protocole.



(a) Évaluation linéaire.



(b) Apprentissage semi-supervisé.



(c) Transfert d'apprentissage.

Figure 5.5: Étapes à suivre pour les protocoles d'évaluation proposés afin d'estimer les performances d'approches en SSRL événementiel.

5.4 Expérimentations : Analyse des Performances

Dans cette section, les résultats de nos expérimentations en utilisant les protocoles décrits en Section 5.3 sont rapportés. En premier lieu, nous abordons la première problématique posée pour les approches de SSRL basées sur des augmentations de données comme la nôtre, à savoir le choix d'une distribution de transformations D qui assure un pré-entraînement efficace. Ensuite, sur base de cette première partie, nous évaluons notre approche de SSRL événementiel en suivant les protocoles d'évaluations introduits.

5.4.1 Détails d'Implémentation

Toutes les expériences sont implémentées avec PyTorch [Pas+19] (+ SpikingJelly [Fan+20] pour simuler les neurones impulsionnels IF) et exécutées sur un GPU NVIDIA A40. Pour toutes les expériences, les modèles sont entraînés à l'aide d'un optimiseur SGD avec un taux d'apprentissage de $0,01$ et un planificateur de réduction de cosinus [29] ("*cosine annealing scheduleur*", en anglais). Chaque échantillon est redimensionné à une résolution de $(H \times W) = (128 \times 128)$. En ce qui concerne la fonction de perte VICReg, nous utilisons les mêmes coefficients que l'article original [BPL22]. Les tenseurs d'impulsions sont générés avec $T = 12$ étapes temporelles. Le code est disponible sur `https://github.com/Barchid/exploring_event_ssl`.

5.4.2 Étude sur les Augmentations de Données Événementielles

Les EDAs discutées dans la section 5.2.3 sont évaluées selon le **Protocole d'Évaluation Linéaire** sur DVSGesture [Ami+17] (voir la Section 5.3.2). Nous divisons cette étude en trois étapes progressives, correspondant aux trois types d'EDAs décrites dans la Section 5.2.3 : **(1)** augmentations communes ; **(2)** augmentations géométriques ; et **(3)** augmentations en découpage. Pour chaque étape, nous conservons la configuration d'EDAs la plus performante de l'étape précédente pour évaluer les EDAs de l'étape courante. La Table 5.3 présente les résultats de l'étude.

En général, nous observons que les CSNNs ont de loin les performances les plus faibles, mais ce problème est grandement atténué par les variantes Étudiant-Professeur, ce qui ouvre des perspectives prometteuses spécifiques aux SNNs similaire à des travaux en apprentissage supervisé [KMM22].

Dans la première étape, nous incorporons progressivement toutes les EDAs communes et observons que les modèles présentent de meilleures performances à mesure

Étape 1: EDAs Communes			Jumeaux			Étudiant-Professeur			
Noise	Crop	PolFlip	CSNN	2D	3D	CSNN	2D	CSNN	3D
			12.12	11.36	57.95	59.47	61.36	58.33	55.68
✓			**60.23**	74.24	69.70	62.12	64.02	68.26	60.23
✓	✓		56.44	81.82	74.62	**73.86**	**76.89**	69.32	64.02
✓	✓	✓	56.44	**83.33**	**76.52**	73.11	74.62	**70.63**	**67.80**

Étape 2: EDAs Géométriques					Jumeaux			Étudiant-Professeur			
StatTran	StatRot	DynTran	DynRot	StatDynGeo	CSNN	2D	3D	CSNN	2D	CSNN	3D
✓	✓				49.24	77.65	65.15	60.98	68.18	65.53	62.5
		✓	✓		68.56	75.38	76.14	**74.24**	**73.11**	**72.73**	67.05
				✓	**68.94**	**79.55**	**77.27**	67.05	70.45	71.97	**68.18**

Étape 3: EDAs en Découpage				Jumeaux			Étudiant-Professeur			
Cutout	EventDrop	EventCopy	EventCopyDrop	CSNN	2D	3D	CSNN	2D	CSNN	3D
✓				**71.59**	85.11	88.64	71.59	72.35	68.56	65.91
	✓			68.18	**87.12**	75	75	76.52	**76.52**	**76.14**
		✓		65.16	81.82	83.33	76.52	**77.52**	73.86	68.94
			✓	70.83	**87.12**	**89.39**	**76.89**	75.76	75	73.48

Table 5.3: Étude sur l'impact des EDAs sur notre méthode. Les scores sont obtenus via le protocole d'évaluation lineaire sur DVSGesture [Ami+17].

que le nombre d'EDAs augmente. La combinaison des trois transformations se classe en première ou deuxième position en termes de performances pour presque tous les encodeurs convolutifs. L'amélioration des performances avec l'augmentation des EDAs peut être attribuée au fait qu'elles ne se superposent pas (c'est-à-dire qu'elles n'appliquent pas le même genre de distortion). En résultat, cela augmente les flux d'événements d'entrée sans sacrifier les informations sémantiques exposées.

Dans la deuxième étape, nous comparons les translations/rotations statiques et dynamiques, et nous observons que les encodeurs convolutifs dotés de capacités de traitement spatio-temporel (c'est-à-dire, CSNN et 3D-CNN) obtiennent de meilleurs résultats avec des transformations dynamiques. L'augmentation **StatDynGeo** proposée atteint, pour la majorité des configurations, des meilleurs résultats, mettant en évidence l'efficacité des relations `OneOf` pour le SSRL événementiel. De manière surprenante, nous constatons que les distorsions géométriques nuisent au pré-entraînement concernant les 2D-CNN par rapport à l'application exclusive des EDAs communes.

Dans la troisième étape, nous comparons toutes les EDAs et observons les avantages d'incorporer une EDA en découpage dans la distribution des transformations. Bien que toutes les transformations en découpage rapportent au moins une performance supérieure pour un encodeur convolutif spécifique, `EventCopyDrop` se distingue régulièrement (en deuxième ou première position), ce qui en fait un concurrent solide. Par conséquent, l'ajout de EventCopy dans la relation `OneOf` de EventDrop [Gu+21] est un choix recommandé pour définir une bonne distribution D générale.

Pour résumer, nous pouvons tirer trois règles générales de cette étude pour concevoir une distribution D généralement efficace :

1. L'incorporation de plus d'augmentations communes conduit à de meilleures performance de pré-entraînement.
2. La sélection d'une EDA géométrique et d'une EDA en découpage renforce considérablement le pré-entraînement par SSRL, et il est plus intéressant d'inclure une EDA géométrique dynamique (`DynTran`, `DynRot` ou `StatDynGeo`) pour les encodeurs convolutifs avec un traitement spatio-temporel (CSNN ou 3D-CNN).
3. Les relations `OneOf` donnent de bons résultats en permettant l'utilisation de plusieurs EDAs similaires sans se chevaucher.

Bien que ces observations soient faites sur notre approche, il est possible qu'elles soient vérifiées dans des travaux futurs en SSRL événementiel qui se basent également sur des augmentations de données (par exemple, en apprentissage contrastif [Che+20]).

Pour le reste de ce chapitre, nous utilisons la distribution $D = \{$`Noise`, `Crop`, `PolFlip`, `StatDynGeo`, `EventCopyDrop`$\}$. Les résultats obtenus par les réseaux "Professeurs" ne sont plus montrés dans la suite des expérimentations, car la conception asymétrique ne leur est pas bénéfique par rapport aux variantes "Jumeaux".

5.4.3 Résultats des Protocoles d'Évaluation

La Table 5.4 présente les résultats des protocoles d'évaluation linéaire et apprentissage semi-supervisé (décrits aux Section 5.3.2 et 5.3.3, respectivement). Notre approche obtient des performances encourageantes sur toutes les bases de données, démontrant l'efficacité des architectures d'encodage conjoint pour le SSRL événementiel. Les résultats prometteurs obtenus avec l'entraînement semi-supervisé suggèrent également que notre approche réduit avec succès la dépendance aux données annotées. Nous observons que les réseaux 2D-CNN et 3D-CNN surpassent l'architecture CSNN en raison de la fonction d'activation binaire du mécanisme impulsionnel, qui est moins expressive que les activations en valeurs réelles des ANNs. Cette constatation suggère que des méthodes SSRL spécialisées pour les neurones impulsionnels pourraient mieux convenir aux SNNs, même si la variante "Étudiant-Professeur" atténue cette baisse de résultats.

La Table 5.5 présente les performances obtenues dans le protocole de transfert

Base de Données	Protocole	CSNN	2D	3D	$CSNN_{2D}$	$CSNN_{3D}$
DVSGesture	Linear	70.83	87.12	**89.39**	76.89	76.52
	SemiSup-10%	60.98	75.52	**81.44**	66.67	69.31
	SemiSup-25%	75.00	87.12	**90.15**	76.14	80.30
N-Caltech101	Linear	64.29	64.39	**69.46**	62.34	65.67
	SemiSup-10%	56.72	**64.64**	62.80	53.96	53.50
	SemiSup-25%	66.02	**72.79**	71.64	62.22	59.93
ASL-DVS	Linear	95.32	**99.38**	98.68	97.87	97.30
	SemiSup-05%	95.66	**97.06**	96.62	93.54	95.66
	SemiSup-10%	99.51	99.64	**99.70**	99.48	99.48

Table 5.4: Résultats des protocoles d'évalutation linéaire et d'apprentissage semi-supervisé. "SemiSup-XX%" qualife le protocole d'apprentissage semi-supervisé, où XX% de l'ensemble d'entraînement est utilisé pour l'affinage. $CSNN_{2D}$ et $CSNN_{3D}$ sont des CSNNs pré-entraînés en utilisant la variante "Étudiant-Professeur" avec un 2D-CNN et un 3D-CNN, respectivement.

Base de Données						
Pré-entraînement	Affinage	CSNN	2D	3D	$CSNN_{2D}$	$CSNN_{3D}$
DVSGesture	**DailyAction-DVS**	77.93	88.28	84.83	**91.03**	87.59
ASL-DVS	**N-CARS**	92.81	94.61	**95.64**	93.30	93.35

Table 5.5: Résultats du protocole de transfert d'apprentissage.

d'apprentissage d'une base de données à une autre. Les performances rapportées démontrent la transférabilité des caractéristiques apprises par notre approche et confirment ainsi l'intérêt du pré-entraînement d'encodeurs convolutifs par SSRL en vision événementielle.

En ce qui concerne les travaux antérieurs sur le SSRL événementiel [Kle+22; YPL23], il est possible de comparer les résultats de notre approche par rapport aux performances sur N-Cars [Sir+18]. Plus précisément, tandis que [Kle+22] et [YPL23] obtiennent respectivement des précisions de 98,55% et 97,93% avec des modèles plus lourds basés sur des architectures ViT et une supervision complète, notre encodeur convolutif léger (ici, un 3D-CNN) atteint une précision légèrement inférieure mais prometteuse de 95,64%. Cependant, la comparaison directe n'est pas simple étant donné que ces travaux appliquent un affinage complet sur l'ensemble d'apprentissage de N-Cars, tandis que notre protocole de transfert d'apprentissage n'entraîne que le classifieur linéaire sur N-Cars (pour rappel, les encodeurs convolutifs sont pré-entraînés sur ASL-DVS de manière non-supervisée). De plus, notre méthode a une portée plus

générale car elle prend également en compte le pré-entraînement sur des bases de données à comportement dynamique.

Les performances compétitives de notre méthode avec des encodeurs convolutifs légers suggèrent que le SSRL événementiel peut apprendre de bonnes représentations, permettant une exploration plus aisée de nouvelles applications en vision événementielle sans devoir construire des bases de données annotées massives et coûteuses. La réduction du besoin d'étiquetage démontré dans cette section ouvre de nouvelles possibilités pour aisément créer des applications innovantes.

5.4.4 Mise en Perspective avec les Approches Supervisées

Pour mettre nos résultats en perspective, nous rapportons les meilleurs résultats obtenus dans nos expériences et les comparons avec les méthodes entièrement supervisées de l'état de l'art. Nous montrons que nos encodeurs convolutifs, sans affinement complet sur l'ensemble d'apprentissage, obtiennent des performances compétitives par rapport à des modèles plus lourds (c'est-à-dire, avec plus de poids entraînables) et sont même capables d'atteindre de meilleures performances (notamment sur DailyAction-DVS [Liu+21a]). Par conséquent, nous démontrons que notre cadre SSRL peut être une alternative compétitive et rentable à l'apprentissage supervisé à grande échelle pour les travaux futurs. Le reste de cette section détaille les comparaisons pour chaque base de données étudiée.

La Table 5.6 présente la comparaison des performances sur ASL-DVS [Bi+20]. Nos modèles 2D-CNN et 3D-CNN obtiennent des performances compétitives sur les protocoles d'évaluation linéaire et d'apprentissage semi-supervisé. En d'autres termes, ces résultats sont atteints en affinant l'encodeur convolutif sur peu de données (voire sans affinage). Nos modèles sont seulement surpassés par ESTF-Net [Wan+22a], une architecture plus lourde (un Vision Transformer) avec $\approx$ 46.7M paramètres. Ces résultats démontrent que notre méthode de SSRL événementiel peut apprendre des représentations qui dépassent les capacités des modèles entièrement supervisés et plus complexes. Par exemple, notre 2D-CNN pré-entraîné (c'est-à-dire une architecture ResNet-18) montre un gain de +1.48% de précision par rapport à EST [Geh+19], un modèle ResNet-34 pré-entraîné sur ImageNet [Den+09].

La Table 5.7 présente la comparaison des performances sur N-Cars [Sir+18]. De manière similaire à nos observations pour ASL-DVS, nos ConvEncs étudiés sur le protocole de transfert d'apprentissage montrent des performances compétitives et sont

Méthode	Description	Précision (%)
RG-CNNs [Bi+20]	Graph Neural Network	90.1
EST [Geh+19]	Représentation optimisée + ResNet34 (pré-entraîné sur ImageNet [Den+09])	97.9
MVF-Net [DCL21]	Graph Neural Network	97.1
EV-VGCNN [Den+22]	Voxel + Graph CNN	98.3
VMV-GCN [Xie+22]	Graph Neural Network	98.9
AMAE [DLC20]	Encodeur de mouvement adaptif + ResNet-34 (pré-entraîné sur ImageNet [Den+09])	98.4
ESTF-Net [Wan+22a]	Vision Transformer Spatial et Temporel	**99.9**
Contrib.	2D-CNN sur Eval. Linéaire (*encodeur non-supervisé*)	99.38
Contrib.	2D-CNN - SemiSup-05%	97.06
Contrib.	3D-CNN - SemiSup-10%	<u>99.70</u>

Table 5.6: Comparaison des performances avec les méthodes supervisées de l'état de l'art sur ASL-DVS [Bi+20].

seulement surpassés par une architecture CNN plus lourde (ResNet-34) entraînée avec EventMix [SZZ22]. Cela suggère la bonne transférabilité des représentations apprises sur des bases de données à comportement statique.

La Table 5.8 détaille la comparaison des performances sur N-Caltech101 [Orc+15]. Notre approche surpasse de nombreux modèles entièrement supervisés basés sur les réseaux de neurones à graphes [Bi+20; SGS22; DCL21], mais obtient des scores inférieurs à d'autres travaux basés sur des architectures CNNs. Cela peut s'expliquer par le fait que la méthode proposé ne peut pas apprendre des caractéristiques discriminantes pour un grand nombre de catégories (par exemple, 101 classes pour N-Caltech101). Néanmoins, les résultats restent encourageants car ils sont obtenus avec peu ou pas de supervision.

La Table 5.9 présente la comparaison des performances sur DVSGesture [Ami+17]. Les résultats indiquent que le modèle 3D-CNN affiné sur 25% de l'ensemble d'entraînement donne de meilleurs résultats que Bina-Rep [BMD22], une architecture ResNet-18

Méthode	Description	Précision (%)
RG-CNNs [Bi+20]	Graph Neural Network	91.4
EST [Geh+19]	Représentation optimisée + ResNet34 (pré-entraîné sur ImageNet [Den+09])	91.9
Bina-Rep [BMD22]	ResNet-18	92.04
MVF-Net [DCL21]	Graph Neural Network	92.7
AsyNet [Mes+20]	Asyn. Sparse VGG13	94.4
EV-VGCNN [Den+22]	Voxel + Graph CNN	95.3
EventMix [SZZ22]	ResNet-34 + EDA	**96.54**
Contrib.	3D-CNN via le transfert d'apprentissage (*encodeur pré-entraîné sur ASL-DVS [Bi+20]*)	<u>95.64</u>
Contrib.	2D-CNN via le transfert d'apprentissage (*encodeur pré-entraîné sur ASL-DVS [Bi+20]*)	94.61
Contrib.	$CSNN_{3D}$ via le transfert d'apprentissage (*encodeur pré-entraîné sur ASL-DVS [Bi+20]*)	93.35

Table 5.7: Comparaison des performances avec les méthodes supervisées de l'état de l'art sur N-Cars [Sir+18].

Méthode	Description	Précision (%)
RG-CNNs [Bi+20]	Graph Neural Network	65.70
AEGNN [SGS22]	Graph Neural Network	66.80
MVF-Net [DCL21]	Graph Neural Network	68.70
AsyNet [Mes+20]	Asyn. Sparse VGG13	74.50
VMV-GCN [Xie+22]	Graph Neural Network	77.80
NDA [Li+22b]	Spiking ResNet-19 + EDAs	<u>78.00</u>
NDA [Li+22b]	Spiking VGG11 + EDAs	**81.70**
Contrib.	3D-CNN sur Eval. Linéaire (*encodeur non-supervisé*)	69.46
Contrib.	2D-CNN - SemiSup-10%	64.64
Contrib.	2D-CNN - SemiSup-25%	72.79

Table 5.8: Comparaison des performances avec les méthodes supervisées de l'état de l'art sur N-Caltech101 [Orc+15].

Méthode	Description	Précision (%)
Bina-Rep [BMD22]	ResNet-18	87.88
TrueNorth [Ami+17]	CSNN (16 couches)	91.77
LIF-Net [He+20]	CSNN (8 couches)	93.40
Rollout [Kug+20]	Spiking VGG16	95.98
EventMix [SZZ22]	ResNet-18 + EDA	<u>96.75</u>
TA-Net [Yao+21]	CSNN + Attention temporelle	**98.61**
Ours	3D-CNN sur Eval. Linéaire (*encodeur non-supervisé*)	89.77
Ours	3D-CNN - SemiSup-10%	81.44
Ours	3D-CNN - SemiSup-25%	90.15

Table 5.9: Comparaison des performances avec les méthodes supervisées de l'état de l'art sur DVSGesture [Ami+17].

Méthode	Description	Précision (%)
[Xia+19]	SNN avec SPA learning	68.30
[Liu+20]	SNN avec SPA learning	76.90
Motion-based SNN [Liu+21a]	neurones sensibles au mouvement + classifieur SNN	<u>90.30</u>
Contrib.	$CSNN_{2D}$ via le transfert d'apprentissage (*encodeur pré-entraîné sur DVSGesture [Ami+17]*)	**91.03**

Table 5.10: Comparaison des performances avec les méthodes supervisées de l'état de l'art sur DailyAction-DVS [Liu+21a].

entraînée avec une technique spécifique de représentation événementielle spatio-temporelle. Cependant, les autres approches entièrement supervisées obtiennent des résultats supérieurs à notre pré-entraînement basé sur l'apprentissage auto-supervisé des événements. Ces résultats suggèrent que notre approche présente certaines limites pour l'apprentissage de représentations optimales pour les tâches de vision spatio-temporelle basées sur les événements, telles que la reconnaissance d'activité (c'est-à-dire, sur les données à comportement dynamique). Cela peut être attribué à la conception de notre méthode, où les encodeurs convolutifs produisent des caractéristiques calculées sur l'ensemble de la séquence, ce qui est plus efficace pour extraire des informations spatiales mais moins pour les informations temporelles.

La Table 5.10 présente la comparaison des performances sur DailyAction-DVS [Liu+21a]. Notre $CSNN_{2D}$ avec des caractéristiques pré-entraînées sur DVSGesture [Ami+17] surpasse tous les travaux précédents sans nécessiter d'affinage de l'encodeur. Cela met en évidence la grande transférabilité de notre cadre de SSRL événementiel.

Cependant, il convient de noter que les travaux précédents évalués sur cette base de données ne sont pas des réseaux de neurones profonds comme notre ConvEnc. Par conséquent, la différence de complexité doit être prise en compte lors de la comparaison des résultats.

5.5 Expérimentations : Analyses Quantitatives des Représentations

Bien que mesurer la performance des encodeurs convolutifs pré-entraînés sur plusieurs bases de données est une étape primordiale pour estimer l'efficacité d'une approche de SSRL, les métriques obtenues (dans ce travail, le taux de précision) restent des mesures indirectes de l'intérêt des représentations apprises.

Dans cette section, nous conduisons une série d'analyses expérimentales sur les propriétés de ces représentations à l'aide de métriques populaires pour l'étude de méthodes non-supervisées [GS22; WL21; WI20]. Dans le cadre de notre travail, ces analyses quantitatives permettent de comprendre les différences entre tous les types d'encodeurs convolutifs abordés (2D-CNN, 3D-CNN et CSNN) lorsqu'ils sont pré-entraînés via une approche de SSRL événementiel.

5.5.1 Analyse d'Uniformité et Tolérance

5.5.1.1 Compromis d'Uniformité - Tolérance

Pour évaluer la qualité des représentations apprises, nous utilisons deux métriques : Uniformité et Tolérance [WL21; WI20]. La métrique d'Uniformité, notée $\mathcal{L}_{\text{uni}}$, mesure la proximité des représentations avec une distribution uniforme sur l'hypersphère des caractéristiques, ce qui indique la capacité de l'encodeur convolutif à apprendre des représentations séparables. D'autre part, la métrique de Tolérance, notée $\mathcal{L}_{\text{tol}}$, utilise les annotations de l'ensemble de données pour estimer dans quelle mesure les représentations capturent les relations sémantiques entre les échantillons. Les formulations pour l'Uniformité et la Tolérance sont fournies respectivement dans les équations 5.9 et 5.10.

$$\mathcal{L}_{\text{uni}} = \log \mathop{\mathbb{E}}_{x,y \sim p_{data}} [e^{-t||f(x)-f(y)||_2^2}] \tag{5.9}$$

$$\mathcal{L}_{\text{tol}} = \mathop{\mathbb{E}}_{x,y \sim p_{data}} [(||f(x)||_2^T ||f(y)||_2) \cdot I_{gt(x)=gt(y)}] \tag{5.10}$$

, où $f(\cdot)$ désigne un encodeur convolutif qui extrait des représentations à partir d'un échantillon de la base de données p_{data}, $I_{gt(x)=gt(y)}$ désigne la fonction indicatrice utilisée pour déterminer si une paire donnée d'échantillons x et y partage la même

catégorie, avec 1 indiquant une correspondance ($gt(x) = gt(y)$) et 0 indiquant une incompatibilité. Un hyperparamètre d'échelle $t = 2$ est utilisé, de manière similaire aux travaux antérieurs [WI20; WL21].

Les recherches antérieures [GS22; WL21] suggèrent que les approches en SSRL atteignent une qualité de représentation optimale en équilibrant les mesures d'Uniformité et de Tolérance. Notre objectif dans cette phase d'expérimentation est de comparer les encodeurs convolutifs pré-entraînés en se basant sur ce compromis.

5.5.1.2 Résultats

La Figure 5.6 illustre les résultats des mesures d'Uniformité et Tolérance de tous les encodeurs convolutifs sur DVSGesture [Ami+17] et ASL-DVS [Bi+20]. En général, nous observons des différences importantes entre tous les encodeurs pré-entraînés, ce qui suggère que notre approche n'affecte pas tous les types de CNNs de manière égale, même lorsque la méthodologie de SSRL est la même. De manière intéressante, nous observons des résultats fortement déséquilibrés pour les 2D/3D-CNNs sur les bases de données à comportement statique (ASL-DVS [Bi+20]), tandis que ces encodeurs convolutifs obtiennent des résultats équilibrés sur les bases de données à comportement dynamique (DVS-Gesture [Ami+17]). Ce déséquilibre prononcé en faveur de l'Uniformité suggère que l'hypothèse du compromis Uniformité-Tolérance [WI20] ne prévaut pas pour de meilleures performances après le pré-entraînement par SSRL événementiel. En revanche, les CSNNs présentent des mesures équilibrées, mais nous notons une augmentation de l'Uniformité avec les variantes Étudiant-Professeur, ce qui confirme les performances améliorées constatées en Section 5.4.

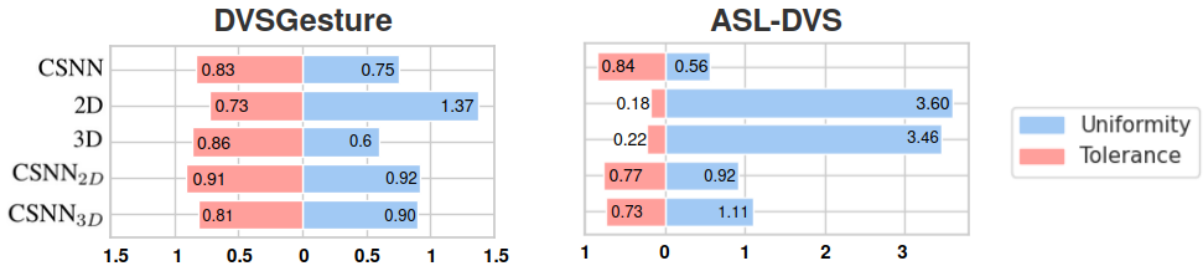


Figure 5.6: Résultats de l'analyse d'Uniformité - Tolérance sur les bases de données DVSGesture [Ami+17] et ASL-DVS [Bi+20].

5.5.2 Étude de Similarité des Représentations

5.5.2.1 Analyse par Alignement de Noyau Centré Linéaire

En suivant des études similaires en SSRL basé sur des images [GS22], nous calculons le CKA linéaire pour évaluer la similarité des représentations extraites par différents encodeurs convolutifs pré-entraînés. Pour ce faire, nous obtenons les matrices des représentations pour deux encodeurs convolutifs différents (par exemple un 2D-CNN et un CSNN), notées A et B, et calculons leurs matrices de Gram respectives $E = AA^T$ and $L = BB^T$. La valeur CKA est calculée comme le critère d'indépendance normalisé de Hilbert-Schmidt (HSIC) [Gre+07]:

$$\text{CKA}(E, L) = \frac{\text{HSIC}(E, L)}{\sqrt{\text{HSIC}(E, E)\text{HSIC}(L, L)}} \tag{5.11}$$

Pour faire simple, la valeur obtenue est une valeur entre 0 et 1 qui évalue la similarité entre les deux ensembles de représentations extraites par les encodeurs. Grâce à cette mesure, il est possible de comparer les informations extraites pour chacun des encodeurs convolutifs étudiés et de rapidement déterminer les similarités ou différences entre ceux-ci.

5.5.2.2 Résultats

Nous mesurons les valeurs de CKA linéaire de tous les encodeurs convolutifs pré-entraînés sur DVSGesture [Ami+17], ainsi que ces mêmes encodeurs entraînés avec supervision (à des fins de comparaison). Les représentations utilisées pour le calcul du CKA linéaire sont celles extraites sur l'ensemble de validation de DVSGesture. En plus des représentations finales (c'est-à-dire, $\mathbf{Y}^d$), nous comparons les vecteurs de caractéristiques intermédiaires à la fin de tous les blocs résiduels des architectures ResNet [He+16] (voir Section 4.3.6.3 pour une explicaiton sur les 5 blocs résiduels d'une architecture ResNet). La Figure 5.7 montre les résultats de cette analyse sous la forme de matrices de confusion.

Nous observons des similarités considérables avec les études précédentes sur le SSRL basé sur des images [GS22] : les caractéristiques extraites ont tendance à diverger de plus en plus lorsque l'on observe des couches profondes, et les représentations finales montrent les plus grandes différences. Dans les couches de bas niveau (Conv 1 et Res. 2), nous observons des différences entre les 2D/3D-CNNs et les CSNNs, ce qui peut

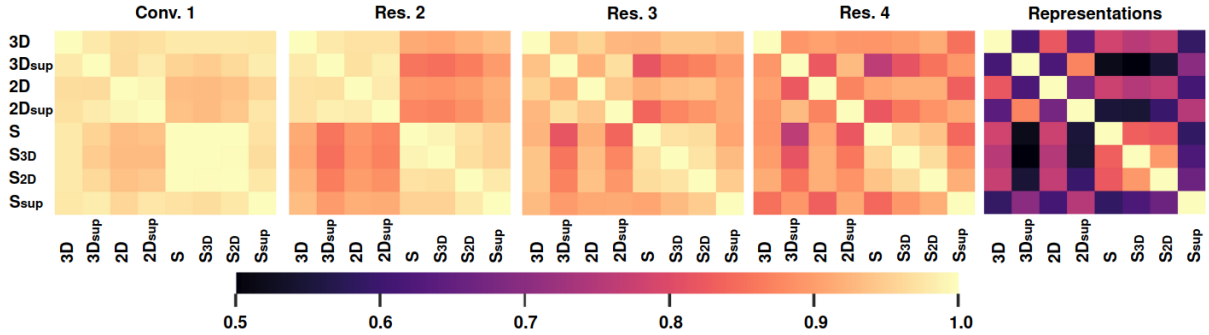


Figure 5.7: Matrices de confusion des valeurs de CKA linéaire pour tous les encodeurs convolutifs étudiés. Une matrice est présentée pour la première couche de convolution (Conv. 1), pour les trois premiers blocs résiduels des architectures ResNet (Res. 2/3/4), et pour les représentations finales des encodeurs. **3D**, **2D**, and **S** sont les **3D**-CNN, **2D**-CNN et C**S**NN, respectivement. $\mathbf{S_{2D}}$ et $\mathbf{S_{3D}}$ désignent les CSNNs couplé à un 2D/3D-CNN dans la variante Étudiant-Professeur. L'indice $_{\mathbf{sup}}$ désigne un apprentissage supervisé du ConvEnc lié.

être expliqué par les différences entre les neurones impulsionnels et artificiels. D'autre part, nous vérifions les constatations de [GS22] selon lesquels les représentations supervisées et non supervisées divergent le plus dans la dernière couche.

Lorsque nous nous concentrons uniquement sur l'analyse des représentations finales, nous constatons que les variantes "Étudiant-Professeur" font en sorte que les CSNNs étudiants apprennent des caractéristiques qui divergent non seulement des CSNNs jumeaux, mais aussi des encodeurs convolutifs professeurs associés. Alors que les différences entre l'étudiant et le professeur peuvent être expliquées par les différences entre les neurones impulsionnels et artificiels, la divergence observée entre tous les CSNNs confirme les impacts des variantes Étudiant-Professeur sur l'entraînement des SNNs. Fait intéressant, les représentations des CSNNs ne sont pas particulièrement plus proches des 3D-CNNs que des 2D-CNNs, malgré leurs capacités communes à traiter la dimension temporelle des tenseurs impulsionnels.

5.6 Conclusion

5.6.1 Récapitulatif des Contributions

Dans ce chapitre, nous présentons une approche de référence pour l'apprentissage auto-supervisé de représentations (SSRL) événementiel en utilisant différents types d'encodeurs convolutifs, notamment des 2D-/3D-CNN et CSNN dans une architecture d'encodage conjoint. De plus, nous introduisons deux variantes de cette méthode : "Jumeaux" (c'est-à-dire, des réseaux siamois avec des poids partagés) et une conception asymétrique "Étudiant-Professeur" qui améliore le pré-entraînement d'un CSNN avec un réseau 2D/3D-CNN en tant que réseau enseignant.

Vu que notre approche se base sur la distorsion de données événementielles en entrée, nous utilisons notre méthode comme un contexte de comparaison pour étudier l'intérêt de diverses EDAs pour pré-entraîner un modèle. De plus, nous concevons des nouvelles transformations événementielles afin d'améliorer les résultats obtenus avec notre approche. Comme étape préliminaire à nos expérimentations, nous observons une série de règles générales pour définir une distribution d'EDAs généralement efficace.

En constatant la difficulté de comparer les travaux antérieurs en SSRL événementiel [Li+22b; Kle+22; YPL23] à cause des différences dans leurs évaluations, nous établissons des protocoles standards pour comparer les méthodes futures, sur base des propriétés recherchées en SSRL. Durant nos expériences, nous constatons que notre méthode obtient des résultats prometteurs vis-à-vis des objectifs fixés pour les protocoles. De plus, les scores de précision atteints sont aussi compétitifs (voire supérieurs) aux travaux en apprentissage supervisé, incluant des réseaux de neurones plus complexes.

Outre des évaluations de performances, nous analysons les représentations apprises par les encodeurs convolutifs pré-entraînés sur certaines propriétés recherchées en apprentissage non-supervisé. Via des analyses d'Uniformité-Tolérance et d'alignement de noyau centré linéaire, nous observons des différences considérables entre les représentations apprises selon le type d'encodeur convolutif et la variante employés, suggérant que notre méthode n'impacte pas toutes les configurations de la même manière. D'autre part, cette étude permet de confirmer/infirmer certaines conclusions tirées des recherches en SSRL basé sur les images [GS22], menant à des observations spécifiques aux flux d'événements.

5.6.2 Perspectives

Un des principaux freins à la conception de réseaux de neurones profonds pour la vision étant le besoin important en annotations, le SSRL offre la possibilité d'un pré-entraînement efficace et non supervisé. De plus, si la captation de données est aisée, ce pré-entraînement peut être réalisé dans le même domaine applicatif que la tâche cible, contrairement au pré-entraînement supervisé sur des bases de données massives et généralistes. À travers ce chapitre, il a été montré que ces bénéfices du SSRL sont aussi vérifiés pour la vision événementielle, notamment grâce à la méthode proposée.

En conséquence, notre méthode permet de mitiger le coût important de la phase d'annotation, ce qui permet la conception facilitée de nouvelles applications. Concrètement, il serait possible de voir l'apparition de nouveaux ensembles de données événementielles où on retrouve une grande quantité de données non-labélisées pour le pré-entraînement et une partie réduite munie d'annotations. Cela aurait pour effet d'accélérer l'adoption de caméras événementielles pour des nouveaux cas d'usage.

Notre méthode a mis en lumière l'efficacité du SSRL événementiel, mais aussi des potentielles pistes d'amélioration. Premièrement, notre méthode est conçue pour des encodeurs convolutifs qui extraient une représentation sur l'ensemble de toutes les étapes temporelles. Cela limite l'utilisation d'une telle technique à des tâches de vision où les informations cruciales sont susceptibles d'évoluer dans le temps (par exemple, le tracking d'objets). Deuxièmement, les évaluations ont montré que certains types d'encodeurs convolutifs, notamment les CSNNs, ne bénéficient pas autant de notre approche. Ceci met l'accent sur l'intérêt de réfléchir la conception de futurs travaux en prenant en compte le modèle de réseau de neurones employé.

6

Conclusion et Travaux Futurs

Sommaire

6.1 Bilan des Contributions

Récemment, les technologies neuromorphiques, en particulier les réseaux de neurones impulsionnels (SNNs) et les caméras événementielles, se sont imposées comme des solutions prometteuses pour la création de systèmes de vision artificielle économes en énergie. Cette adoption croissante dans la communauté scientifique soulève des défis substantiels dans le domaine de la vision par ordinateur. Ces défis se concentrent principalement sur l'amélioration des approches neuromorphiques basées sur l'apprentissage profond et sur une compréhension approfondie de ces technologies pour mieux appréhender leurs aspects fondamentaux.

La motivation de ce manuscrit est de relever ces deux défis majeurs. Dans nos travaux, nous explorons trois axes de recherche fondamentaux dans le domaine des approches neuromorphiques :

1. L'amélioration de la représentation des flux d'événements sous forme d'images événementielles.

2. Le développement et l'analyse de modèles SNNs profonds pour résoudre des problèmes de vision artificielle, que ce soit avec des images statiques ou des flux d'événements.

3. La réduction de la dépendance aux données événementielles annotées en vision événementielle, afin d'accélérer le développement de modèles d'apprentissage.

Chacune de ces pistes de recherche est associée à une contribution principale proposée dans nos travaux.

6.1.1 Première Contribution : Images Événementielles Bina-Rep

La première contribution de nos travaux réside dans le développement de "*Bina-Rep*", une nouvelle technique de représentation d'événements en images événementielles. Ce qui distingue cette méthode, c'est son intégration d'une sorte d'information temporelle sans nécessiter de calculs supplémentaires. Cette information temporelle est exprimée sous forme de l'ordre d'arrivée des événements à chaque pixel encodé en une représentation numérique à T bits. En utilisant une architecture de réseaux de neurones artificiels (ANN) de type convolutif (abrégée en 2D-CNN), nous avons démontré la compétitivité de notre approche de représentation d'événements. Nous l'avons comparée à d'autres méthodes de représentation populaires parfois plus complexes

que notre technique. Nos expérimentations ont montré que les images événementielles Bina-Rep permettent d'atteindre des performances compétitives par rapport à d'autres méthodes de représentation d'événements similaires. Par exemple, l'emploi de 3 images Bina-Rep surpasse l'usage d'une séquence plus lourde de 10 images événementielles binaires avec 87.88% contre 82.95% de précision sur DVSGesture [Ami+17]. De plus, en mettant en perspective les résultats atteints par Bina-Rep avec les performances des travaux de l'état de l'art, nous avons montré que Bina-Rep est capable d'atteindre de meilleurs scores de précision. Par exemple, nous avons observé un score de 93.25% de précision par Bina-Rep sur CIFAR10-DVS [Li+17], soit un gain de au moins +18.45% par rapport aux autres travaux évalués.

Plus spécifiquement, nous avons constaté que l'intégration d'informations temporelles dans une image Bina-Rep se traduit par des performances supérieures par rapport aux séquences plus lourdes d'images événementielles classiques. Outre les performances sur les flux d'événements bruts, nous avons également évalué la robustesse de Bina-Rep face à des corruptions courantes liées aux caméras événementielles. Nos résultats montrent que Bina-Rep maintient un niveau de robustesse compétitif par rapport aux autres représentations étudiées, notamment lorsque les images Bina-Rep sont utilisées en séquence.

6.1.2 Deuxième Contribution : Développement et Analyse des SNNs en Vision Artificielle

Notre deuxième contribution s'articule autour du développement et de l'analyse de modèles de réseaux de neurones impulsionnels convolutifs (CSNNs) profonds, visant à accomplir des tâches en vision par ordinateur. Plus précisément, nous explorons deux domaines : la localisation d'objet pour les images statiques et les flux d'événements, ainsi qu'une nouvelle tâche de vision événementielle appelée "Reconnaissance d'Expressions Faciales (FER) Événementielle".

Nous avons d'abord réalisé une preuve de concept basée sur une architecture CSNN profonde de type encodeur-décodeur pour la localisation d'objet, une tâche de régression dépendante du traitement de l'information spatiale. Les résultats prometteurs de cette preuve de concept (63.2% mIoU) ont confirmé la pertinence des technologies neuromorphiques pour nos travaux tout en fournissant des enseignements pour la conception de CSNNs adaptés à la vision artificielle.

Fort de ces enseignements, nous avons développé un modèle générique CSNN capable de traiter divers problèmes (classification, régression, ...) tout en facilitant l'analyse des capacités d'extraction d'informations utiles du CSNN sous la forme d'un encodeur convolutif. Nous avons testé ce modèle générique sur la localisation d'objet, à la fois pour les images statiques et les flux d'événements. En comparant ce modèle CSNN à un modèle ANN similaire, nous avons montré que le modèle CSNN obtient des performances légèrement inférieures à l'ANN pour les images statiques (jusqu'à 4.09% mIoU en moins que l'ANN), mais excelle dans le cas des flux d'événements (71.53% mIoU pour le CSNN vs 70.44% mIoU pour l'ANN). Parallèlement à ces comparaisons, nous avons analysé divers aspects fondamentaux des SNNs (codage neuronal, latence temporelle, etc.) dans le contexte de la localisation d'objet, révélant des conclusions significativement différentes par rapport aux études antérieures sur les SNNs. De plus, en estimant la consommation énergétique des deux types de réseaux de neurones, nous avons mis en lumière l'efficacité énergétique nette de notre modèle CSNN par rapport à l'ANN, avec une consommation d'énergie étant de 44.82× à 126.6× moins importante.

Au-delà de l'étude des CSNNs pour les tâches de régression, nous avons introduit "Spiking-Fer", une adaptation de notre modèle CSNN générique pour une nouvelle tâche de vision événementielle jusqu'alors inexplorée, à savoir la FER événementielle. Pour explorer cette nouvelle tâche, nous avons créé des bases de données événementielles synthétiques et comparé les performances de notre modèle Spiking-Fer à un ANN similaire. Dans cette étude exploratoire de la FER événementielle, notre modèle Spiking-Fer a montré des résultats compétitifs par rapport à l'ANN, tout en consommant significativement moins d'énergie (de 47.42× à 65.38× moins d'énergie consommée). Nous avons également profité de cette nouvelle tâche pour étudier l'impact de l'augmentation de données événementielles (EDAs) sur l'apprentissage des modèles de vision événementielle. Notre étude a révélé que, bien que les EDAs aient amélioré les performances des deux types de réseaux de neurones, leurs configurations ont eu des impacts différents sur les CSNNs par rapport aux ANNs.

6.1.3 Troisième Contribution : la Réduction du Besoin en Données Événementielles Annotées

Notre troisième contribution se concentre sur le développement d'une technique visant à réduire la nécessité de disposer de données annotées pour l'entraînement de modèles profonds (ANN ou SNN) en vision événementielle. La méthode que nous proposons repose sur une approche d'apprentissage auto-supervisé de représentation (SSRL)

visant à pré-entraîner de manière non supervisée un encodeur convolutif (CSNN, 2D-CNN ou 3D-CNN). Cette approche se base sur une architecture d'encodage conjoint et une distribution d'EDAs préalablement définie. Nos encodeurs convolutifs pré-entraînés parviennent à obtenir de bonnes performances en utilisant uniquement une fraction réduite des annotations d'une base de données (par exemple, un score de précision de 99.7% sur ASL-DVS [Bi+20] en utilisant uniquement 10% des annotations). Étant donné le caractère novateur du SSRL événementiel, nous avons établi des protocoles d'évaluation pour évaluer notre méthode, ainsi que pour faciliter les comparaisons avec les travaux futurs du domaine. De plus, nous avons étudié l'influence de différentes EDAs sur la qualité du pré-entraînement. Les conclusions tirées de nos nombreuses expérimentations fournissent des informations essentielles sur les distorsions à inclure dans la distribution d'EDAs en SSRL événementiel, ainsi que sur les différences dans les caractéristiques extraites par les réseaux de neurones étudiés (CSNN, 2D-CNN et 3D-CNN).

6.2 Travaux Futurs

Au cours de nos recherches couvrant divers aspects des technologies neuromorphiques, nous avons repéré des axes d'amélioration pour nos méthodes et saisi des opportunités pour étendre nos travaux à de nouveaux contextes. Ces perspectives ouvrent la voie aux futures travaux qui viendront compléter les contributions exposées dans ce manuscrit.

6.2.1 De la Simulation Logicielle au Déploiement sur Matériel Neuromorphique

Les récentes avancées en matière de conception de matériel neuromorphique, en particulier les processeurs neuromorphiques à haute capacité neuronale et synaptique (comme l'Intel Loihi 2 [Orc+21]), ont montré que les SNNs de grande envergure, tels que ceux développés au cours de nos travaux, peuvent désormais être exécutés de manière adéquate. L'une des orientations les plus cruciales de nos futures recherches consistera à migrer nos simulations logicielles actuelles sur GPU vers un déploiement concret sur un processeur neuromorphique. Ceci permettra d'explorer les spécificités d'un déploiement réel par rapport à la simulation, offrant ainsi des perspectives essentielles pour le développement de modèles CSNNs en vision par ordinateur. La prise en compte de ces spécificités revêt un intérêt significatif pour la communauté scientifique [Dav+21]. De plus, le déploiement de nos modèles sur du matériel neuromorphique nous permettra d'évaluer directement leur efficacité énergétique, éliminant ainsi le besoin d'estimations abstraites. L'évaluation de ces mesures concrètes revêt un intérêt considérable pour mettre en évidence l'efficacité énergétique des SNNs par rapport aux ANNs.

6.2.2 Améliorations des Approches Proposées

Les méthodes élaborées au sein de ce manuscrit, à savoir les images événementielles Bina-Rep, le modèle générique CSNN, et la méthode de SSRL événementiel, ont démontré des avancées significatives dans leurs contextes respectifs. Toutefois, chacune de ces contributions offre des opportunités d'amélioration qui pourraient potentiellement accroître leur efficacité ou leur applicabilité dans d'autres travaux de recherche.

Des Images Bina-Rep Configurées Automatiquement. L'utilisation des images événementielles Bina-Rep présente une difficulté considérable : la sélection appropriée des hyperparamètres requis. Des valeurs mal ajustées peuvent entraîner une saturation

de la représentation, entravant ainsi un traitement adéquat par l'algorithme de vision ultérieur. Toutefois, la recherche des valeurs d'hyperparamètres adéquates peut potentiellement nécessiter des sessions d'optimisation longues et coûteuses en termes de temps et de puissance de calcul. Pour répondre à cette problématique, une amélioration significative de Bina-Rep consisterait à permettre la configuration automatique de ces hyperparamètres pendant la phase d'apprentissage. Cela aurait pour effet de produire une représentation d'événements plus simple d'utilisation et plus efficace. Une solution envisagée à cet égard serait d'intégrer une phase de pré-entraînement visant à maximiser la différence entre chaque image événementielle binaire composant l'image Bina-Rep. Cette intuition aurait pour effet d'augmenter la quantité d'informations exposées par la représentation. Une telle piste d'amélioration pourrait aussi être évaluée sur d'autres bases de données à comportement dynamique (par exemple, DVS-Lip [Tan+22a]), plus complexes que DVSGesture [Ami+17] dont certaines classes peuvent être discriminées sans nécessiter d'informations temporelles.

Des Améliorations de l'Extraction de Caractéristiques par le Modèle Générique CSNN. Le modèle générique CSNN que nous avons développé comprend un composant spécifique appelé l'accumulateur d'impulsions. Cet accumulateur est chargé de produire un unique vecteur de caractéristiques à valeurs réelles en traitant le tenseur impulsionnel sur une période de T étapes temporelles. Cependant, cette forme de caractéristiques extraites par le modèle CSNN n'est pas universellement adaptée à toutes les tâches de vision. Certaines applications pourraient potentiellement bénéficier de l'extraction de caractéristiques sous d'autres dimensions, voire même pour chaque étape temporelle distincte. D'un autre côté, le perfectionnement de l'accumulateur d'impulsions pourrait améliorer les capacités d'extraction d'informations du modèle. Par exemple, en intégrant des mécanismes d'attention spatio-temporelle [Zhu+22b] lors de l'accumulation des impulsions, il serait possible de sélectionner les caractéristiques impulsionnelles cruciales pour une reconstruction en valeurs réelles de meilleure qualité.

Un SSRL Événementiel Adapté au Modèle Pré-entraîné. La méthode de SSRL événementiel que nous avons développée s'est révélée efficace pour réduire la dépendance aux données annotées. Cependant, nous avons noté que les avantages de cette méthode n'ont pas le même impact sur tous les types de réseaux de neurones que nous avons étudiés. Plus précisément, nous avons constaté qu'un encodeur convolutif CSNN tire moins profit de cette méthode de SSRL événementiel que les autres encodeurs tels que

les 2D-CNN et 3D-CNN. Cette observation met en évidence la nécessité de concevoir une méthode de SSRL événementiel mieux adaptée au pré-entraînement d'un type précis de réseaux de neurones. Spécifiquement pour les SNNs, cela pourrait impliquer le développement d'une architecture d'encodage conjoint travaillant directement dans le domaine des impulsions, sans nécessiter de conversion vers des représentations en valeurs réelles. De plus, une telle technique de SSRL orientée uniquement vers les impulsions pourrait potentiellement être déployée sur du matériel neuromorphique, en supposant que des algorithmes de rétropropagation du gradient soient implémentés sur ces plateformes à l'avenir. Cette implémentation neuromorphique ouvrirait des perspectives importantes, car elle permettrait l'apprentissage non supervisé directement intégré à la puce neuromorphique.

6.2.3 Exploration de Nouveaux Contextes Applicatifs

Aujourd'hui, les progrès réalisés dans le domaine des technologies neuromorphiques ont atteint un stade avancé, permettant ainsi la conception de systèmes de vision efficaces pouvant être déployés dans des environnements réels. Ces avancées ont notamment suscité un intérêt croissant au sein de la communauté scientifique pour ces technologies. Au cours de nos expérimentations, nous avons observé que les approches que nous avons développées au cours de cette thèse ont produit des résultats prometteurs dans divers contextes tels que la FER, la reconnaissance d'actions, la localisation d'objets, ...

Dans la continuité de notre exploration de la FER événementielle avec Spiking-Fer, il est pertinent d'envisager l'exploration de nouveaux domaines d'application en utilisant les technologies neuromorphiques. Cette démarche vise à élargir l'exploitation de ces technologies à d'autres domaines et à mieux comprendre leurs particularités par rapport aux méthodes conventionnelles existantes. Il est à noter que cette perspective serait cohérente avec notre proposition de déploiement sur du matériel neuromorphique indiquée dans la Section 6.2.1. En effet, l'intégration de ces approches neuromorphiques dans des cas applicatifs concrets ajouterait une valeur significative à l'exploration de nouveaux domaines, ce qui pourrait favoriser l'adoption de ces technologies par les chercheurs travaillant dans ces contextes d'application.

A

Annexes

Sommaire

A.1 Baisse de Précision Relative

Le score de **baisse de précision relative** (*"Relative Accuracy Drop"*, en anglais) [KCP21] a pour objectif de mesurer la perte de précision d'un algorithme, lorsque celui-ci est confronté à des données bruitées par rapport au score obtenu avec ces mêmes données non-altérées. De cette manière, il est possible de comparer les robustesses de plusieurs modèles face à des corruptions de données.

On considère qu'une corruption *corr* donnée (*par exemple :* un bruit additif gaussien) est paramétrée par un niveau de sévérité *sev* croissant de 1 à 5, où chaque niveau de sévérité augmente l'importance de la corruption sur la donnée (voir un exemple en Figure 3.4 avec le bruit d'activité de fond).

À partir d'un modèle entraîné sur un ensemble d'apprentissage (non-altéré) donné, on peut mesurer $Perf_{clean}$, le score de performance sur l'ensemble de validation non-corrompu. Le score de performance à proprement parler dépend de la tâche de vision visée (exemple : le taux de précision pour un problème de classification, la $mIoU$ pour la localisation d'objet, ...). On note $Perf_{sev}^{corr}$, le score de performance obtenu sur l'ensemble de validation altéré par la corruption *corr* à un niveau de sévérité *sev* spécifié. Le score de baisse de précision relative RAD_{sev}^{corr} est alors calculé de la manière suivante :

$$RAD_{sev}^{corr} = \frac{Perf_{clean} - Perf_{sev}^{corr}}{Perf_{clean}} \times 100 \tag{A.1}$$

. Pour finir, on peut estimer la robustesse générale d'un modèle à l'aide du score de **baisse de précision relative moyenne**:

$$mRAD^{corr} = \frac{1}{5} \times \sum_{sev=1}^{5} RAD_{sev}^{corr} \tag{A.2}$$

Il est important de noter que le score de baisse de précision relative permet de comparer la robustesse des approches, non leurs performances absolues. Une approche peut être plus robuste qu'une autre, même si ses performances brutes ne sont pas meilleures. Par exemple, un classifieur aléatoire peut avoir un score de baisse de précision relative "parfait" ($RAD_{sev}^{corr} \approx 0$), contrairement à un modèle d'apprentissage de pointe. Il ne faut donc pas confondre ce score avec une mesure de performances.

A.2 Utilisations de l'Intersection sur l'Union en Localisation d'Objet

Cette annexe décrit les utilisations de la mesure de l'Intersection sur l'Union (*"Intersection over Union"* ou "*IoU*", en anglais) dans le cadre de nos travaux sur la localisation d'objet dans les Sections 4.1 et 4.3. Dans ces travaux, l'IoU est utilisée soit en tant que métrique pour évaluer un modèle de régression de boîte englobante (tel que la détection d'objets [WSH20] ou notre tâche de localisation d'objet désignée en Section 4.1.2), soit en tant que terme dans une fonction de coût pour la régression de boîte englobante [Zhe+20; Rez+19].

A.2.1 Métrique d'Intersection sur l'Union

Dans les tâches de vision telles que la détection d'objets qui impliquent une régression de boîtes englobantes [WSH20], l'IoU est une métrique utilisée pour évaluer la précision d'une boîte englobante prédite par un modèle de vision par rapport à la boîte englobante de la vérité terrain. Dans le cadre de notre problème de localisation d'objet, la métrique IoU (*IoU*) est calculée en utilisant la boîte englobante prédite $\mathbf{B} = \{x_{min}, y_{min}, x_{max}, y_{max}\}$ et la boîte englobante de la vérité terrain associée $\mathbf{B}^{gt} = \{x^{gt}_{min}, y^{gt}_{min}, x^{gt}_{max}, y^{gt}_{max}\}$:

$$IoU = \frac{|\mathbf{B} \cap \mathbf{B}^{gt}|}{|\mathbf{B} \cup \mathbf{B}^{gt}|} \tag{A.3}$$

où $|\mathbf{B} \cap \mathbf{B}^{gt}|$ désigne l'intersection entre les deux boîtes englobantes (c'est-à-dire, la zone de chevauchement entre la boîte englobante prédite et la boîte englobante de la vérité terrain), et $|\mathbf{B} \cup \mathbf{B}^{gt}|$ fait référence à l'union entre les deux boîtes englobantes (c'est-à-dire, la zone englobée à la fois par la boîte englobante prédite et la boîte englobante de la vérité terrain). La Figure A.1 montre un exemple illustré du calcul de l'IoU.

Grâce à cette métrique, l'évaluation du score de IoU sur tout l'ensemble de validation/test d'une base de données est possible, ce qui permet de comparer des modèles de régression de boîtes englobantes entre eux. Pour ce faire, il suffit de calculer la moyenne des scores de IoU sur tout l'ensemble de validation/test afin d'obtenir une métrique finale. On parle alors de "Intersection sur l'Union Moyenne" (*"Mean Intersection over Union"* ou *mIoU*, en anglais).

(a) Les Boîtes englobantes. **(b) Intersection.** **(c) Union**

(d) Calcul de l'IoU.

Figure A.1: Illustration du calcul de l'IoU sur un échantillon provenant de Oxford-IIIT-Pet [Par+12].

A.2.2 Fonction de Coût DIoU : Améliorer l'Intersection sur l'Union par un Terme de Pénalité

Des travaux antérieurs [Yu+16] ont montré que l'IoU peut être utilisée directement en tant que fonction de coût ($\mathcal{L}_{IoU}$) pour optimiser un modèle d'apprentissage :

$$\mathcal{L}_{IoU} = 1 - IoU \tag{A.4}$$

.

Plusieurs études [Rez+19; Zhe+20] constatent des désavantages à l'emploi direct d'une fonction de coût uniquement basée sur l'IoU [Yu+16], tel qu'une convergence lente et des prédictions imprécises. Pour corriger ces problèmes, ces études proposent l'incorporation d'autres termes de pénalité en plus de l'IoU afin d'améliorer le processus d'apprentissage.

Dans nos travaux, nous nous concentrons sur une de ces approches, nommée la fonc-

tion de coût "Distance-IoU" (DIoU) [Zhe+20]. Cette fonction de coût $\mathcal{L}_{DIoU}$ utilise un terme de pénalité qui minimise directement la distance entre les centres des deux boîtes englobantes $\mathbf{B}$ et $\mathbf{B}^{gt}$. Les coordonnées de ces centres, désigné par $\mathbf{b} = (x_{center}, y_{center})$ pour $\mathbf{B}$ et $\mathbf{b}^{gt} = (x^{gt}_{center}, y^{gt}_{center})$ pour $\mathbf{B}^{gt}$, sont déterminés de la manière suivante :

$$\mathbf{b} = (x_{center}, y_{center}) = (\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2}) \tag{A.5}$$

$$\mathbf{b}^{gt} = (x^{gt}_{center}, y^{gt}_{center}) = (\frac{x^{gt}_{min} + x^{gt}_{max}}{2}, \frac{y^{gt}_{min} + y^{gt}_{max}}{2}) \tag{A.6}$$

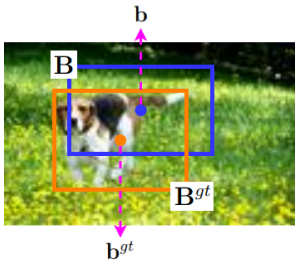
En pratique, le terme de pénalité proposé par [Zhe+20] est la distance normalisée entre ces deux centres des boites englobantes :

$$\mathcal{R} = \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{q^2} \tag{A.7}$$

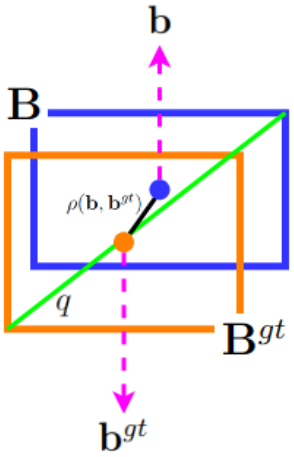
où $\rho(\cdot)$ est la distance euclidienne et q est la longueur diagonale de la plus petite boite qui détoure les deux boîtes englobantes. Finalement, la fonction de coût $\mathcal{L}_{DIoU}$ est calculé de la manière suivante :

$$\mathcal{L}_{DIoU} = \mathcal{L}_{IoU} + \mathcal{R} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{q^2} \tag{A.8}$$

. La Figure A.2 réutilise l'exemple montré en Figure A.1 et illustre les variables additionnelles utilisées pour $\mathcal{L}_{DIoU}$.



(a) Centres des deux boîtes englobantes.



(b) Variables additionnelles utilisées dans le calcul de $\mathcal{L}_{DIoU}$. La longueur diagonale q est colorée en vert. La distance $\rho(\mathbf{b}, \mathbf{b}^{gt})$ est en noir.

Figure A.2: Reprise de l'exemple de la Figure A.1 avec les variables utilisées pour le calcul de la fonction de coût $\mathcal{L}_{DIoU}$.

A.3 Estimation du Coût Énergétique d'un Réseau de Neurones Impulsionnels

La méthodologie utilisée dans nos travaux pour estimer la consommation énergétique (en Sections 4.3.6.3, 4.3.8.3 et 4.4.10) du CSNN composant le modèle générique provient de [KCP21]. De plus, cette méthode nous permet de comparer avec la consommation énergétique d'un modèle ANN possédant la même architecture que le CSNN. Néanmoins, il faut mentionner que cette estimation est uniquement concentrée sur la partie CSNN du modèle générique décrit en Section 4.2, sans considérer l'accumulateur d'impulsions et la couche dense linéaire.

La méthodologie d'estimation [KCP21] est décrite comme suit : tout d'abord, nous quantifions le taux d'impulsions de chaque couche, car les neurones impulsionnels ne consomment de l'énergie que lorsqu'ils génèrent une impulsion. Le taux d'impulsions d'une couche l donnée est calculé comme suit :

$$Rs(l) = \frac{\text{\# spikes of } l \text{ over all time-steps}}{\text{\# neurons of } l} \tag{A.9}$$

Deuxièmement, nous calculons le total des opérations en virgule flottante (FLOPS) d'une couche de neurones impulsionnels ($FLOPs_{SNN}$) en utilisant les FLOPS de la même couche dans un réseau de neurones non impulsionnel ($FLOPs_{ANN}$) et le taux d'impulsions de la couche de neurones impulsionnels :

$$FLOPs_{SNN}(l) = FLOPs_{ANN}(l) \times Rs(l) \tag{A.10}$$

$$FLOPs_{ANN}(l) = \begin{cases} k^2 \times O^2 \times C_{in} \times C_{out} & \text{if } l \text{ is Conv.} \\ C_{in} \times C_{out} & \text{if } l \text{ is Linear.} \end{cases} \tag{A.11}$$

Dans l'équation A.11, k représente la taille du filtre de convolution, O représente la taille des cartes de caractéristiques en sortie, C_{in} représente le nombre de canaux d'entrée et C_{out} représente le nombre de canaux de sortie.

Enfin, la consommation d'énergie totale d'un modèle peut être estimée sur la technologie CMOS [Hor14] en utilisant le total des FLOPS sur l'ensemble des couches. Le tableau A.1 présente le coût énergétique des opérations pertinentes dans un processus CMOS 45nm. L'opération MAC dans les ANNs nécessite une addition (`ADD 32 bits`

Operation	**Energy** (pJ)
32bit FP MULT (E_{MULT})	3.7
32bit FP ADD (E_{ADD})	0.9
32bit FP MAC (E_{MAC})	4.6 ($= E_{MULT} + E_{ADD}$)
32bit FP AC (E_{AC})	0.9

Table A.1: Coût énergétique des opérations ayant lieu dans un encodeur convolutif. Ces coût sont basés sur un processus CMOS de 45nm [KCP21].

FP) et une multiplication FP (`MULT 32 bits FP`) [Sze+17a], tandis que les SNNs ne nécessitent qu'une seule addition FP par opération MAC en raison du traitement binaire des impulsions. La consommation d'énergie totale des ANNs et des SNNs est représentée respectivement par E_{ANN} et E_{SNN} :

$$E_{ANN} = \sum_{l} FLOPs_{ANN}(l) \times E_{MAC} \tag{A.12}$$

$$E_{SNN} = \sum_{l} FLOPs_{SNN}(l) \times E_{AC} \tag{A.13}$$

B

Bibliographie

Sommaire

Références

[Lap07] LM Lapicque. "Recherches quantitatives sur l'excitation electrique des nerfs". In: *J Physiol Paris* 9 (1907), pp. 620–635.

[AZ26] Edgar D Adrian and Yngve Zotterman. "The impulses produced by sensory nerve-endings: Part II. The response of a Single End-Organ". In: *The Journal of physiology* 61.2 (1926), p. 151.

[MP43] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[HH52] Alan L Hodgkin and Andrew F Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of physiology* 117.4 (1952), p. 500.

[Bro54] George H. Brown. "Mathematical Formulations of the NTSC Color Television Signal". In: *Proceedings of the IRE* 42.1 (1954), pp. 66–71.

[Ros58] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[HW59] David H Hubel and Torsten N Wiesel. "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of physiology* 148.3 (1959), p. 574.

[HW62] David H Hubel and Torsten N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1 (1962), p. 106.

[Fuk80] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[RMP86] David E Rumelhart, James L McClelland, and CORPORATE PDP Research Group. *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations*. MIT press, 1986.

[LF87] Yann Le Cun and Françoise Fogelman-Soulié. “Modèles connexionnistes de l’apprentissage”. In: *Intellectica* 2.1 (1987), pp. 114–143.

[Mar88] N Justin Marshall. “A unique colour and polarization vision system in mantis shrimps”. In: *Nature* 333.6173 (1988), pp. 557–560.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.

[Wer90] Paul J Werbos. “Backpropagation through time: what it does and how to do it”. In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.

[PM92] William B Pennebaker and Joan L Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[OR93] John O’Keefe and Michael L Recce. “Phase relationship between hippocampal place units and the EEG theta rhythm”. In: *Hippocampus* 3.3 (1993), pp. 317–330.

[MM94] Misha Mahowald and Misha Mahowald. “The silicon retina”. In: *An Analog VLSI System for Stereoscopic Vision* (1994), pp. 4–65.

[TFM96] Simon Thorpe, Denis Fize, and Catherine Marlot. “Speed of processing in the human visual system”. In: *nature* 381.6582 (1996), pp. 520–522.

[BP98] Guo-qiang Bi and Mu-ming Poo. “Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type”. In: *Journal of neuroscience* 18.24 (1998), pp. 10464–10472.

[LeC+98] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[TG98] Simon Thorpe and Jacques Gautrais. “Rank order coding”. In: *Computational Neuroscience: Trends in Research, 1998*. Springer, 1998, pp. 113–118.

[Abb99] Larry F Abbott. “Lapicque’s introduction of the integrate-and-fire model neuron (1907)”. In: *Brain research bulletin* 50.5-6 (1999), pp. 303–304.

[BKL00] Sander M Bohte, Joost N Kok, and Johannes A La Poutré. "SpikeProp: backpropagation for networks of spiking neurons." In: *ESANN*. Vol. 48. Bruges. 2000, pp. 419–424.

[Bra00] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[LF00] Simon P Liversedge and John M Findlay. "Saccadic eye movements and cognition". In: *Trends in cognitive sciences* 4.1 (2000), pp. 6–14.

[RR00] Pamela Reinagel and R Clay Reid. "Temporal coding of visual information in the thalamus". In: *Journal of neuroscience* 20.14 (2000), pp. 5392–5400.

[SMA00] Sen Song, Kenneth D Miller, and Larry F Abbott. "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity". In: *Nature neuroscience* 3.9 (2000), pp. 919–926.

[TDV01] Simon Thorpe, Arnaud Delorme, and Rufin Van Rullen. "Spike-based strategies for rapid processing". In: *Neural networks* 14.6-7 (2001), pp. 715–725.

[GK02a] Wulfram Gerstner and Werner M Kistler. "Mathematical formulations of Hebbian learning". In: *Biological cybernetics* 87.5 (2002), pp. 404–415.

[GK02b] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[Izh03] Eugene M Izhikevich. "Simple model of spiking neurons". In: *IEEE Transactions on neural networks* 14.6 (2003), pp. 1569–1572.

[Boa04] Kwabena A Boahen. "A burst-mode word-serial address-event link-I: Transmitter design". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.7 (2004), pp. 1269–1280.

[FFP04] Li Fei-Fei, Rob Fergus, and Pietro Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". In: *2004 conference on computer vision and pattern recognition workshop*. IEEE. 2004, pp. 178–178.

[Izh04] Eugene M Izhikevich. "Which model to use for cortical spiking neurons?" In: *IEEE transactions on neural networks* 15.5 (2004), pp. 1063–1070.

[JB04] Roland S Johansson and Ingvars Birznieks. "First spikes in ensembles of human tactile afferents code complex spatial fingertip events". In: *Nature neuroscience* 7.2 (2004), pp. 170–177.

[Toy+04] Taro Toyoizumi et al. "Spike-timing dependent plasticity and mutual information maximization for a spiking neuron model". In: *Advances in neural information processing systems* 17 (2004).

[Heb05] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[Pan+05] Maja Pantic et al. "Web-based database for facial expression analysis". In: *2005 IEEE international conference on multimedia and Expo*. IEEE. 2005, 5–pp.

[Ber06] Raphael Berner. "High-speed USB2. 0 AER interfaces". In: *proyecto fin de master dirigido por Dr. Tobias Delbruck, Prof. Anton Civit Balcells y Dr. Alejandro Linares Barranco, Universidad de Sevilla e Instituto de Neuroinformática* (2006).

[CH06] Nicholas T Carnevale and Michael L Hines. *The NEURON book*. Cambridge University Press, 2006.

[BSF07] Joseph M Brader, Walter Senn, and Stefano Fusi. "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics". In: *Neural computation* 19.11 (2007), pp. 2881–2912.

[GD07] Marc-Oliver Gewaltig and Markus Diesmann. "Nest (neural simulation tool)". In: *Scholarpedia* 2.4 (2007), p. 1430.

[Gre+07] Arthur Gretton et al. "A kernel statistical test of independence". In: *Advances in neural information processing systems* 20 (2007).

[LPD08] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. "A 128×128 120 dB 15μ s latency asynchronous temporal contrast vision sensor". In: *IEEE journal of solid-state circuits* 43.2 (2008), pp. 566–576.

[Dav+09] Andrew P Davison et al. "PyNN: a common interface for neuronal network simulators". In: *Frontiers in neuroinformatics* 2 (2009), p. 388.

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[KSK09] Jurgen Kogler, Christoph Sulzbachner, and Wilfried Kubinger. "Bio-inspired stereo vision system with silicon retina imagers". In: *International Conference on Computer Vision Systems*. Springer. 2009, pp. 174–183.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[Luc+10] Patrick Lucey et al. "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression". In: *2010 ieee computer society conference on computer vision and pattern recognition-workshops*. IEEE. 2010, pp. 94–101.

[PMW10] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level

video compression and time-domain CDS". In: *IEEE Journal of Solid-State Circuits* 46.1 (2010), pp. 259–275.

[Ben+11] Ryad Benosman et al. "Asynchronous event-based Hebbian epipolar geometry". In: *IEEE Transactions on Neural Networks* 22.11 (2011), pp. 1723–1734.

[Kog+11] Jürgen Kogler et al. "Address-event based stereo vision with bio-inspired silicon retina imagers". In: *Advances in theory and applications of stereo vision* (2011), pp. 165–188.

[Mas+11] Jonathan Masci et al. "Stacked convolutional auto-encoders for hierarchical feature extraction". In: *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*. Springer. 2011, pp. 52–59.

[Van+11] Job Van Der Schalk et al. "Moving faces, looking places: validation of the Amsterdam Dynamic Facial Expression Set (ADFES)." In: *Emotion* 11.4 (2011), p. 907.

[Zha+11] Guoying Zhao et al. "Facial expression recognition from near-infrared videos". In: *Image and Vision Computing* 29.9 (2011), pp. 607–619.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[Lin12] Tony Lindeberg. "Scale invariant feature transform". In: (2012).

[Par+12] Omkar M. Parkhi et al. "Cats and dogs". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 3498–3505.

[PB12] Hélene Paugam-Moisy and Sander M Bohte. "Computing with spiking neuron networks." In: *Handbook of natural computing* 1 (2012), pp. 1–47.

[Ben+13] Ryad Benosman et al. "Event-based visual flow". In: *IEEE transactions on neural networks and learning systems* 25.2 (2013), pp. 407–417.

[Car+13] Kristofor D Carlson et al. "Biologically plausible models of homeostasis and STDP: stability and learning in spiking neural networks". In: *The 2013 international joint conference on neural networks (IJCNN)*. IEEE. 2013, pp. 1–8.

[OFB13] Joao P Oliveira, Mario AT Figueiredo, and Jose M Bioucas-Dias. "Parametric blur estimation for blind restoration of natural images: Linear motion and out-of-focus". In: *IEEE Transactions on Image Processing* 23.1 (2013), pp. 466–477.

[Que+13] Damien Querlioz et al. “Immunity to device variations in a spiking neural network with memristive nanodevices”. In: *IEEE transactions on nanotechnology* 12.3 (2013), pp. 288–295.

[RS13] Kishore K Reddy and Mubarak Shah. “Recognizing 50 human action categories of web videos”. In: *Machine vision and applications* 24.5 (2013), pp. 971–981.

[Xu+13] Yan Xu et al. “A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks”. In: *Neural Networks* 43 (2013), pp. 99–113.

[Bek+14] Trevor Bekolay et al. “Nengo: a Python tool for building large-scale functional brain models”. In: *Frontiers in neuroinformatics* 7 (2014), p. 48.

[Bra+14] Christian Brandli et al. “A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor”. In: *IEEE Journal of Solid-State Circuits* 49.10 (2014), pp. 2333–2341.

[Chi+14] Elisabetta Chicca et al. “Neuromorphic electronic circuits for building autonomous cognitive systems”. In: *Proceedings of the IEEE* 102.9 (2014), pp. 1367–1388.

[DVL14] Tobi Delbruck, Vicente Villanueva, and Luca Longinotti. “Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision”. In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2014, pp. 2636–2639.

[Fur+14] Steve B Furber et al. “The spinnaker project”. In: *Proceedings of the IEEE* 102.5 (2014), pp. 652–665.

[Gam+14] João Gama et al. “A survey on concept drift adaptation”. In: *ACM computing surveys (CSUR)* 46.4 (2014), pp. 1–37.

[Hor14] Mark Horowitz. “Computing’s energy problem (and what we can do about it). In 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)”. In: *IEEE, feb*. 2014.

[KB14] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[Pos+14] Christoph Posch et al. “Retinomorphic event-based vision sensors: bioinspired cameras with spiking output”. In: *Proceedings of the IEEE* 102.10 (2014), pp. 1470–1484.

[SJE14] Amit Satpathy, Xudong Jiang, and How-Lung Eng. “LBP-based edge-texture features for object recognition”. In: *IEEE Transactions on image Processing* 23.5 (2014), pp. 1953–1964.

[SZ14] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[Zha+14] Bo Zhao et al. “Feedforward categorization on AER motion events using cortex-like features in a spiking neural network”. In: *IEEE transactions on neural networks and learning systems* 26.9 (2014), pp. 1963–1978.

[Ako+15] Filipp Akopyan et al. “Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip”. In: *IEEE transactions on computer-aided design of integrated circuits and systems* 34.10 (2015), pp. 1537–1557.

[Bre15] Romain Brette. “Philosophy of the spike: rate-based vs. spike-based theories of the brain”. In: *Frontiers in systems neuroscience* 9 (2015), p. 151.

[CCK15] Yongqiang Cao, Yang Chen, and Deepak Khosla. “Spiking deep convolutional neural networks for energy-efficient object recognition”. In: *International Journal of Computer Vision* 113.1 (2015), pp. 54–66.

[Die+15] Peter U Diehl et al. “Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing”. In: *2015 International joint conference on neural networks (IJCNN)*. ieee. 2015, pp. 1–8.

[DT15] Cho Dong-il and L Tae-jae. “A review of bioinspired vision sensors and their applications”. In: *Sens. Mater* 27 (2015), pp. 447–463.

[HE15] Eric Hunsberger and Chris Eliasmith. “Spiking deep networks with LIF neurons”. In: *arXiv preprint arXiv:1510.08829* (2015).

[IS15] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. pmlr. 2015, pp. 448–456.

[KZS+15] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop*. Vol. 2. 1. Lille. 2015.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[Li+15] Chenghan Li et al. “Design of an RGBW color VGA rolling and global shutter dynamic and active-pixel vision sensor”. In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2015, pp. 718–721.

[Lio+15] Sze-Teng Liong et al. “Automatic apex frame spotting in micro-expression database”. In: *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. IEEE. 2015, pp. 665–669.

[LBE15] Zachary C Lipton, John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning". In: *arXiv preprint arXiv:1506.00019* (2015).

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[Mar+15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[MCG15] Nathan Matsuda, Oliver Cossairt, and Mohit Gupta. "Mc3d: Motion contrast 3d scanning". In: *2015 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 2015, pp. 1–10.

[Orc+15] Garrick Orchard et al. "Converting static image datasets to spiking neuromorphic datasets using saccades". In: *Frontiers in neuroscience* 9 (2015), p. 437.

[PS15] SGOPAL Patro and Kishore Kumar Sahu. "Normalization: A preprocessing stage". In: *arXiv preprint arXiv:1503.06462* (2015).

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[SNB15] Stephan Schraml, Ahmed Nabil Belbachir, and Horst Bischof. "Event-driven stereo matching for real-time 3D panoramic vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 466–474.

[Tra+15] Du Tran et al. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[BS16] Pierre Baldi and Peter Sadowski. "A theory of local learning, the learning channel, and the optimality of backpropagation". In: *Neural Networks* 83 (2016), pp. 51–74.

[He+16] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[Hu+16] Yuhuang Hu et al. “DVS benchmark datasets for object tracking, action recognition, and object recognition”. In: *Frontiers in neuroscience* 10 (2016), p. 405.

[HE16] Eric Hunsberger and Chris Eliasmith. “Training spiking deep networks for neuromorphic hardware”. In: *arXiv preprint arXiv:1611.05141* (2016).

[KLD16] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. “Real-time 3D reconstruction and 6-DoF tracking with an event camera”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, pp. 349–364.

[Kov+16] Adriana Kovashka et al. “Crowdsourcing in computer vision”. In: *Foundations and Trends® in computer graphics and Vision* 10.3 (2016), pp. 177–243.

[Kri+16] Matej Kristan et al. “A novel performance evaluation methodology for single-target trackers”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.11 (2016), pp. 2137–2155.

[Lag+16] Xavier Lagorce et al. “Hots: a hierarchy of event-based time-surfaces for pattern recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.7 (2016), pp. 1346–1359.

[Lil+16] Timothy P Lillicrap et al. “Random synaptic feedback weights support error backpropagation for deep learning”. In: *Nature communications* 7.1 (2016), p. 13276.

[LH16] Ilya Loshchilov and Frank Hutter. “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).

[NPL16] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. “Phased lstm: Accelerating recurrent network training for long or event-based sequences”. In: *Advances in neural information processing systems* 29 (2016).

[Nøk16] Arild Nøkland. “Direct feedback alignment provides learning in deep neural networks”. In: *Advances in neural information processing systems* 29 (2016).

[Yu+16] Jiahui Yu et al. “Unitbox: An advanced object detection network”. In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, pp. 516–520.

[Ami+17] Arnon Amir et al. “A low power, fully event-based gesture recognition system”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7243–7252.

[BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[Bin+17] Jonathan Binas et al. "DDD17: End-to-end DAVIS driving dataset". In: *arXiv preprint arXiv:1711.01458* (2017).

[DT17] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).

[GHC17] Menghan Guo, Jing Huang, and Shoushun Chen. "Live demonstration: A 768× 640 pixels 200Meps dynamic vision sensor". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2017, pp. 1–1.

[Hes+17] Joel Hestness et al. "Deep learning scaling is predictable, empirically". In: *arXiv preprint arXiv:1712.00409* (2017).

[How+17] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[Hua+17] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[JT17] Xin Jin and Xiaoyang Tan. "Face alignment in-the-wild: A survey". In: *Computer Vision and Image Understanding* 162 (2017), pp. 1–22.

[Kau17] Eric Kauderer-Abrams. "Quantifying translation-invariance in convolutional neural networks". In: *arXiv preprint arXiv:1801.01450* (2017).

[Kir+17] James Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[Li+17] Hongmin Li et al. "Cifar10-dvs: an event-stream dataset for object classification". In: *Frontiers in neuroscience* 11 (2017), p. 309.

[Lin+17] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[Mor+17] Saber Moradi et al. "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors

(DYNAPs)". In: *IEEE transactions on biomedical circuits and systems* 12.1 (2017), pp. 106–122.

[Mue+17] Elias Mueggler et al. "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM". In: *The International Journal of Robotics Research* 36.2 (2017), pp. 142–149.

[Rue+17] Bodo Rueckauer et al. "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification". In: *Frontiers in neuroscience* 11 (2017), p. 682.

[Smi17] Leslie N Smith. "Cyclical learning rates for training neural networks". In: *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2017, pp. 464–472.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[Son+17] Bongki Son et al. "4.1 A 640× 480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation". In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE. 2017, pp. 66–67.

[Sze+17a] Vivienne Sze et al. "Efficient processing of deep neural networks: A tutorial and survey". In: *Proceedings of the IEEE* 105.12 (2017), pp. 2295–2329.

[Sze+17b] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

[Tra+17] Du Tran et al. "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *CoRR* abs/1711.11248 (2017). arXiv: 1711.11248. URL: http://arxiv.org/abs/1711.11248.

[Vas+17] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[Yun+17] Sang-Seok Yun et al. "Social skills training for children with autism spectrum disorder using a robotic behavioral intervention system". In: *Autism Research* 10.7 (2017), pp. 1306–1323.

[Aga18] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". In: *arXiv preprint arXiv:1803.08375* (2018).

[All+18] Benjamin Allaert et al. "Impact of the face registration techniques on facial expressions recognition". In: *Signal Processing: Image Communication* 61 (2018), pp. 44–53.

[AC18] Ignacio Alzugaray and Margarita Chli. "ACE: An efficient asynchronous corner tracker for event cameras". In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 653–661.

[AZA18] Jamil Azzeh, Bilal Zahran, and Ziad Alqadi. "Salt and pepper noise: Effects and removal". In: *JOIV: International Journal on Informatics Visualization* 2.4 (2018), pp. 252–256.

[Dav+18] Mike Davies et al. "Loihi: A neuromorphic manycore processor with on-chip learning". In: *Ieee Micro* 38.1 (2018), pp. 82–99.

[Dev+18] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[GRS18] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3867–3876.

[Haz+18] Hananel Hazan et al. "Bindsnet: A machine learning-oriented spiking neural networks library in python". In: *Frontiers in neuroinformatics* 12 (2018), p. 89.

[Jia+18] Huaizu Jiang et al. "Super slomo: High quality estimation of multiple intermediate frames for video interpolation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9000–9008.

[Khe+18] Saeed Reza Kheradpisheh et al. "STDP-based spiking deep convolutional neural networks for object recognition". In: *Neural Networks* 99 (2018), pp. 56–67.

[Kim+18] Jaehyun Kim et al. "Deep neural networks with weighted spikes". In: *Neurocomputing* 311 (2018), pp. 373–386.

[Lee+18] Jinmook Lee et al. "UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision". In: *IEEE Journal of Solid-State Circuits* 54.1 (2018), pp. 173–185.

[Maq+18] Ana I Maqueda et al. "Event-based vision meets deep learning on steering prediction for self-driving cars". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5419–5427.

[Mar+18] Alexandre Marcireau et al. "Event-based color segmentation with a high dynamic range sensor". In: *Frontiers in neuroscience* 12 (2018), p. 135.

[Mit+18] Anton Mitrokhin et al. "Event-based moving object detection and tracking". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–9.

[Moz+18] Milad Mozafari et al. “First-spike-based visual categorization using reward-modulated STDP”. In: *IEEE transactions on neural networks and learning systems* 29.12 (2018), pp. 6178–6190.

[Nec+18] Alexander Neckar et al. “Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model”. In: *Proceedings of the IEEE* 107.1 (2018), pp. 144–164.

[Pou+18] Delphine Poux et al. “Mastering occlusions by using intelligent facial frameworks based on the propagation of movement”. In: *2018 International conference on content-based multimedia indexing (CBMI)*. IEEE. 2018, pp. 1–6.

[Rad+18] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).

[RGS18] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. “ESIM: an open event camera simulator”. In: *Conference on robot learning*. PMLR. 2018, pp. 969–982.

[RL18] B Rueckauer and SC Liu. *Conversion of analog to spiking neural networks using sparse temporal coding, 2018 IEEE Int. Symp. Circuits and Systems (ISCAS)*. 2018.

[SO18] Sumit B Shrestha and Garrick Orchard. “Slayer: Spike layer error reassignment in time”. In: *Advances in neural information processing systems* 31 (2018).

[Sia+18] Mennatullah Siam et al. “A comparative study of real-time semantic segmentation for autonomous driving”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 587–597.

[Sir+18] Amos Sironi et al. “HATS: Histograms of averaged time surfaces for robust event-based object classification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1731–1740.

[SPR18] Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. “Stdp-based unsupervised feature learning using convolution-over-time in spiking neural networks for energy-efficient neuromorphic computing”. In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 14.4 (2018), pp. 1–12.

[Tan+18] Chuanqi Tan et al. “A survey on deep transfer learning”. In: *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*. Springer. 2018, pp. 270–279.

[Tav+18] Gemma Taverni et al. “Front and back illuminated dynamic and active pixel vision sensors comparison”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 65.5 (2018), pp. 677–681.

[TN18] Luke Taylor and Geoff Nitschke. “Improving deep learning with generic data augmentation”. In: *2018 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2018, pp. 1542–1547.

[WTV18] Runchun M Wang, Chetan S Thakur, and Andre Van Schaik. “An FPGA-based massively parallel neuromorphic cortex simulator”. In: *Frontiers in neuroscience* 12 (2018), p. 213.

[Yao+18] Yongqiang Yao et al. “Texture and geometry scattering representation-based facial expression recognition in 2D+ 3D videos”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.1s (2018), pp. 1–23.

[ZG18] Friedemann Zenke and Surya Ganguli. “Superspike: Supervised learning in multilayer spiking neural networks”. In: *Neural computation* 30.6 (2018), pp. 1514–1541.

[Zha+18a] Rumin Zhang et al. “An algorithm for obstacle detection based on YOLO and light filed camera”. In: *2018 12th International Conference on Sensing Technology (ICST)*. IEEE. 2018, pp. 223–226.

[Zha+18b] Tong Zhang et al. “Spatial–temporal recurrent neural network for emotion recognition”. In: *IEEE transactions on cybernetics* 49.3 (2018), pp. 839–847.

[ZW18] Shibo Zhou and Wei Wang. “Object detection based on lidar temporal pulses using spiking neural networks”. In: *arXiv preprint arXiv:1810.12436* (2018).

[Zhu+18a] Alex Zihao Zhu et al. “EV-FlowNet: Self-supervised optical flow estimation for event-based cameras”. In: *arXiv preprint arXiv:1802.06898* (2018).

[Zhu+18b] Alex Zihao Zhu et al. “The multivehicle stereo event camera dataset: An event camera dataset for 3D perception”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 2032–2039.

[Zih+18] Alex Zihao Zhu et al. “Unsupervised event-based optical flow using motion compensation”. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

[Afs+19] Saeed Afshar et al. “Investigation of event-based surfaces for high-speed detection, unsupervised feature extraction, and object recognition”. In: *Frontiers in neuroscience* 12 (2019), p. 1047.

[AM19] Inigo Alonso and Ana C Murillo. "EV-SegNet: Semantic segmentation for event-based cameras". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[Bi+19] Yin Bi et al. "Graph-based object classification for neuromorphic vision sensing". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 491–501.

[Blo+19] Peter Blouw et al. "Benchmarking keyword spotting efficiency on neuromorphic hardware". In: *Proceedings of the 7th annual neuro-inspired computational elements workshop*. 2019, pp. 1–8.

[Cal+19] Enrico Calabrese et al. "Dhp19: Dynamic vision sensor 3d human pose dataset". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019, pp. 0–0.

[CN19] Daniel Canedo and Antonio JR Neves. "Facial expression recognition using computer vision: a systematic review". In: *Applied Sciences* 9.21 (2019), p. 4678.

[CG19a] Shoushun Chen and Menghan Guo. "Live demonstration: CeleX-V: A 1M pixel multi-mode event-based sensor". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2019, pp. 1682–1683.

[CG19b] Shoushun Chen and Menghan Guo. "Live demonstration: CeleX-V: A 1M pixel multi-mode event-based sensor". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2019, pp. 1682–1683.

[Fal19] Pierre Falez. "Improving spiking neural networks trained with spike timing dependent plasticity for image recognition". PhD thesis. Université de Lille, 2019.

[Fal+19] Pierre Falez et al. "Multi-layered spiking neural network with target timestamp threshold adaptation and stdp". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[Gar+19] Eva Garcia-Martin et al. "Estimation of energy consumption in machine learning". In: *Journal of Parallel and Distributed Computing* 134 (2019), pp. 75–88.

[Geh+19] Daniel Gehrig et al. "End-to-end learning of representations for asynchronous event-based data". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5633–5643.

[HD19] Dan Hendrycks and Thomas Dietterich. “Benchmarking neural network robustness to common corruptions and perturbations”. In: *arXiv preprint arXiv:1903.12261* (2019).

[Man+19] Jacques Manderscheid et al. “Speed invariant time surface for learning to detect corner points with event-based cameras”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10245–10254.

[Mia+19] Shu Miao et al. “Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection”. In: *Frontiers in neurorobotics* 13 (2019), p. 38.

[NMZ19] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks”. In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63.

[Pan+19] Xianzhang Pan et al. “Deep temporal–spatial aggregation for video-based facial expression recognition”. In: *Symmetry* 11.1 (2019), p. 52.

[Pas+19] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[Ram+19] Bharath Ramesh et al. “Dart: distribution aware retinal transform for event-based cameras”. In: *IEEE transactions on pattern analysis and machine intelligence* 42.11 (2019), pp. 2767–2780.

[Ras19] Daniel Rasmussen. “NengoDL: Combining deep learning and neuromorphic modelling methods”. In: *Neuroinformatics* 17.4 (2019), pp. 611–628.

[Reb+19a] Henri Rebecq et al. “Events-to-video: Bringing modern computer vision to event cameras”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3857–3866.

[Reb+19b] Henri Rebecq et al. “High speed and high dynamic range video with an event camera”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.6 (2019), pp. 1964–1980.

[Rez+19] Hamid Rezatofighi et al. “Generalized intersection over union: A metric and a loss for bounding box regression”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 658–666.

[Ryu19] Hyunsurk Eric Ryu. “Industrial DVS design; key features and applications”. In: *Conf. on Computer Vision and Pattern Recognition*. Vol. 3. 2019.

[Sen+19a] Abhronil Sengupta et al. “Going deeper in spiking neural networks: VGG and residual architectures”. In: *Frontiers in neuroscience* 13 (2019), p. 95.

[Sen+19b] Abhronil Sengupta et al. “Going deeper in spiking neural networks: VGG and residual architectures”. In: *Frontiers in neuroscience* 13 (2019), p. 95.

[SBG19] Marcel Stimberg, Romain Brette, and Dan FM Goodman. “Brian 2, an intuitive and efficient neural simulator”. In: *elife* 8 (2019), e47314.

[SK19] Timo Stoffregen and Lindsay Kleeman. “Event cameras, contrast maximization and reward functions: An analysis”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12300–12308.

[Tav+19] Amirhossein Tavanaei et al. “Deep learning in spiking neural networks”. In: *Neural networks* 111 (2019), pp. 47–63.

[Tul+19] Stepan Tulyakov et al. “Learning an event sequence embedding for dense event-based deep stereo”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1527–1537.

[Wan+19] Jianbo Wang et al. “Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance”. In: *Proceedings of the 27th ACM international conference on multimedia*. 2019, pp. 374–382.

[WHY+19] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. “Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10081–10090.

[WJ19] Yue Wu and Qiang Ji. “Facial landmark detection: A literature survey”. In: *International Journal of Computer Vision* 127 (2019), pp. 115–142.

[Wu+19] Yujie Wu et al. “Direct training for spiking neural networks: Faster, larger, better”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1311–1318.

[Xia+19] Rong Xiao et al. “An event-driven categorization model for AER image sensors using multispike encoding and learning”. In: *IEEE transactions on neural networks and learning systems* 31.9 (2019), pp. 3649–3657.

[YS19] Jong Chul Ye and Woon Kyoung Sung. “Understanding geometry of encoder-decoder CNNs”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7064–7073.

[ZL19] Wenrui Zhang and Peng Li. "Information-theoretic intrinsic plasticity for online unsupervised learning in spiking neural networks". In: *Frontiers in neuroscience* 13 (2019), p. 31.

[Zhu+19] Alex Zihao Zhu et al. "Unsupervised event-based learning of optical flow, depth, and egomotion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 989–997.

[Bal+20] R Baldwin et al. "Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1701–1710.

[Bi+20] Yin Bi et al. "Graph-based spatio-temporal feature learning for neuromorphic vision sensing". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9084–9098.

[Bra20] Brainchip. *Akida 1000 Ref SoC*. https://brainchip.com/akida-neural-processor-soc/. Accessed: 2023-08-01. 2020.

[Can+20] Marco Cannici et al. "A differentiable recurrent surface for asynchronous event-based data". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer. 2020, pp. 136–152.

[Che+20] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[Cra+20] Benjamin Cramer et al. "The heidelberg spiking data sets for the systematic evaluation of spiking neural networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7 (2020), pp. 2744–2757.

[De +20] Pierre De Tournemire et al. "A large scale event-based detection dataset for automotive". In: *arXiv preprint arXiv:2001.08499* (2020).

[DLC20] Yongjian Deng, Youfu Li, and Hao Chen. "Amae: Adaptive motion-agnostic encoder for event-based object classification". In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4596–4603.

[Dos+20] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[FTB20] Pierre Falez, Pierre Tirilly, and Ioan Marius Bilasco. "Improving stdp-based visual feature learning with whitening". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–8.

[Fan+20] Wei Fang et al. *SpikingJelly*. https://github.com/fangwei123456/spikingjelly. Accessed: 2022-02-01. 2020.

[Fen+20] Yang Feng et al. "Event density based denoising method for dynamic vision sensor". In: *Applied Sciences* 10.6 (2020), p. 2024.

[Fin+20a] T Finateu et al. "A 1280x720 Back-Illuminated Stacked Temporal Contrast Event-Based Vision Sensor with 4.86 um Pixels, 1.066 GEPS Readout, Programmable Event-Rate Controller and Compressive Data-Formatting Pipeline". In: *IEEE International Solid-State Circuits Conference*. 2020.

[Fin+20b] Thomas Finateu et al. "5.10 a 1280× 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 GEPS readout, programmable event-rate controller and compressive data-formatting pipeline". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 112–114.

[Gal+20] Guillermo Gallego et al. "Event-based vision: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.1 (2020), pp. 154–180.

[Geh+20] Daniel Gehrig et al. "Video to events: Recycling video datasets for event cameras". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3586–3595.

[HSR20] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. "Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13558–13567.

[Han+20] Jin Han et al. "Neuromorphic camera guided high dynamic range imaging". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1730–1739.

[He+20] Weihua He et al. "Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences". In: *Neural Networks* 132 (2020), pp. 108–120.

[He20] Zhengyu He. "Deep learning in image classification: A survey report". In: *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. IEEE. 2020, pp. 174–177.

[HGS20] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. "Learning monocular dense depth from events". In: *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 534–542.

[Hu+20] Yuhuang Hu et al. “Ddd20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–6.

[Jia+20a] Rui Jiang et al. “Object tracking on event cameras with offline–online learning”. In: *CAAI Transactions on Intelligence Technology* 5.3 (2020), pp. 165–171.

[Jia+20b] Zhe Jiang et al. “Learning Event-Based Motion Deblurring”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[KMN20] Jacques Kaiser, Hesham Mostafa, and Emre Neftci. “Synaptic plasticity dynamics for deep continuous local learning (DECOLLE)”. In: *Frontiers in Neuroscience* 14 (2020), p. 424.

[Kep+20] Daniel R Kepple et al. “Jointly learning visual motion and confidence from local patches in event cameras”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer. 2020, pp. 500–516.

[Kim+20] Seijoon Kim et al. “Spiking-yolo: spiking neural network for energy-efficient object detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 11270–11277.

[Kug+20] Alexander Kugele et al. “Efficient processing of spatio-temporal data streams with spiking neural networks”. In: *Frontiers in Neuroscience* 14 (2020), p. 439.

[Lee+20] Chankyu Lee et al. “Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 366–382.

[LIB20] Gregor Lenz, Sio-Hoi Ieng, and Ryad Benosman. “Event-based face detection and tracking using the dynamics of eye blinks”. In: *Frontiers in Neuroscience* 14 (2020), p. 587.

[LD20a] Shan Li and Weihong Deng. “Deep Facial Expression Recognition: A Survey”. In: *IEEE Transactions on Affective Computing* (2020), pp. 1–1.

[LD20b] Shan Li and Weihong Deng. “Deep facial expression recognition: A survey”. In: *IEEE transactions on affective computing* 13.3 (2020), pp. 1195–1215.

[Lin+20] Shijie Lin et al. “Efficient spatial-temporal normalization of sae representation for event camera”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4265–4272.

[Liu+20] Qianhui Liu et al. “Effective AER object classification using segmented probability-maximization learning in spiking neural networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 02. 2020, pp. 1308–1315.

[Mes+20] Nico Messikommer et al. “Event-based asynchronous sparse convolutional networks”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 415–431.

[Par+20] Seongsik Park et al. “T2FSNN: Deep spiking neural networks with time-to-first-spike coding”. In: *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2020, pp. 1–6.

[Per+20] Etienne Perot et al. “Learning to detect objects with a 1 megapixel event camera”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16639–16652.

[Pro20] Prophesee. *Prophesee Evaluation Kits*. https://www.prophesee.ai/event-based-evk/. Accessed: 2023-08-01. 2020.

[SB20] Arman Savran and Chiara Bartolozzi. “Face pose alignment with event cameras”. In: *Sensors* 20.24 (2020), p. 7079.

[SSK20] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. “A u-net based discriminator for generative adversarial networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8207–8216.

[Sch+20] Roy Schwartz et al. “Green ai”. In: *Communications of the ACM* 63.12 (2020), pp. 54–63.

[Sri+20] Gopalakrishnan Srinivasan et al. “Training deep spiking neural networks for energy-efficient neuromorphic computing”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 8549–8553.

[Sto+20] Timo Stoffregen et al. “Reducing the sim-to-real gap for event cameras”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*. Springer. 2020, pp. 534–549.

[Suh+20a] Yunjae Suh et al. “A 1280× 960 dynamic vision sensor with a 4.95-μm pixel pitch and motion artifact minimization”. In: *2020 IEEE international symposium on circuits and systems (ISCAS)*. IEEE. 2020, pp. 1–5.

[Suh+20b] Yunjae Suh et al. “A 1280× 960 dynamic vision sensor with a 4.95-μm pixel pitch and motion artifact minimization”. In: *2020 IEEE international symposium on circuits and systems (ISCAS)*. IEEE. 2020, pp. 1–5.

[TXC20] Madhumita A Takalkar, Min Xu, and Zenon Chaczko. “Manifold feature integration for micro-expression recognition”. In: *Multimedia Systems* 26.5 (2020), pp. 535–551.

[TO20] Guray Tonguç and Betul Ozaydın Ozkara. “Automatic recognition of student emotions from facial expressions during a lecture”. In: *Computers & Education* 148 (2020), p. 103797.

[WI20] Tongzhou Wang and Phillip Isola. “Understanding contrastive representation learning through alignment and uniformity on the hypersphere”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9929–9939.

[WL20] Steven Euijong Whang and Jae-Gil Lee. “Data collection and quality challenges for deep learning”. In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 3429–3432.

[WSH20] Xiongwei Wu, Doyen Sahoo, and Steven CH Hoi. “Recent advances in deep learning for object detection”. In: *Neurocomputing* 396 (2020), pp. 39–64.

[Xu+20] Lan Xu et al. “Eventcap: Monocular 3d capture of high-speed human motions using an event camera”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4968–4978.

[Zha+20] Malu Zhang et al. “Spike-timing-dependent back propagation in deep spiking neural networks”. In: *arXiv preprint arXiv:2003.11837* (2020).

[Zhe+20] Zhaohui Zheng et al. “Distance-IoU loss: Faster and better learning for bounding box regression”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 12993–13000.

[Aou+21] Mouath Aouayeb et al. “Learning vision transformer with squeeze and excitation for facial expression recognition”. In: *arXiv preprint arXiv:2107.03107* (2021).

[BMD21] Sami Barchid, José Mennesson, and Chaabane Djéraba. “Review on indoor RGB-D semantic segmentation with deep convolutional neural networks”. In: *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE. 2021, pp. 1–4.

[Car+21] Mathilde Caron et al. “Emerging properties in self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[CLR21] Sayeed Shafayet Chowdhury, Chankyu Lee, and Kaushik Roy. “Towards understanding the effect of leak in spiking neural networks”. In: *Neurocomputing* 464 (2021), pp. 83–94.

[CRR21] Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. “One timestep is all you need: Training spiking neural networks with ultra low latency”. In: *arXiv preprint arXiv:2110.05929* (2021).

[Dav+21] Mike Davies et al. “Advancing neuromorphic computing with loihi: A survey of results and outlook”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 911–934.

[DCL21] Yongjian Deng, Hao Chen, and Youfu Li. “MVF-Net: A multi-view fusion network for event-based object classification”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.12 (2021), pp. 8275–8284.

[Des21] Radosvet Desislavov Georgiev. “Analysis of Deep Learning Inference Compute and Energy Consumption Trends”. In: (2021).

[Din+21] Jianhao Ding et al. “Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks”. In: *arXiv preprint arXiv:2105.11654* (2021).

[Esh+21] Jason K Eshraghian et al. “Training spiking neural networks using lessons from deep learning”. In: *arXiv preprint arXiv:2109.12894* (2021).

[Fan+21a] Wei Fang et al. “Deep residual learning in spiking neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[Fan+21b] Wei Fang et al. “Incorporating learnable membrane time constant to enhance learning of spiking neural networks”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2661–2671.

[Geh+21a] Daniel Gehrig et al. “Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2822–2829.

[Geh+21b] Mathias Gehrig et al. “Dsec: A stereo event camera dataset for driving scenarios”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4947–4954.

[Gu+21] Fuqiang Gu et al. “EventDrop: Data Augmentation for Event-based Learning”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. Ed. by Zhi-Hua Zhou. Main Track. International Joint

Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 700–707. URL: https://doi.org/10.24963/ijcai.2021/97.

[Guo+21] Wenzhe Guo et al. "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems". In: *Frontiers in Neuroscience* 15 (2021), p. 212.

[Hu+21] Xia Hu et al. "Model complexity of deep learning: A survey". In: *Knowledge and Information Systems* 63 (2021), pp. 2585–2619.

[HLD21] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. "v2e: From video frames to realistic DVS events". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1312–1321.

[Inn+21] Simone Undri Innocenti et al. "Temporal binary representation for event-based action recognition". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10426–10432.

[ICL21] Laxmi R Iyer, Yansong Chua, and Haizhou Li. "Is neuromorphic mnist neuromorphic? analyzing the discriminative power of neuromorphic datasets in the time domain". In: *Frontiers in neuroscience* 15 (2021), p. 297.

[Kim+21] Junho Kim et al. "N-imagenet: Towards robust, fine-grained object recognition with event cameras". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2146–2156.

[KCP21] Youngeun Kim, Joshua Chough, and Priyadarshini Panda. "Beyond Classification: Directly Training Spiking Neural Networks for Semantic Segmentation". In: *arXiv preprint arXiv:2110.07742* (2021).

[Len+21] Gregor Lenz et al. *Tonic: event-based datasets and transformations.* Version 0.4.0. Documentation available under https://tonic.readthedocs.io. July 2021. URL: https://doi.org/10.5281/zenodo.5079802.

[Li+21a] Yijin Li et al. "Graph-based Asynchronous Event Processing for Rapid Object Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 934–943.

[Li+21b] Zhe Li et al. "Facial expression-based analysis on emotion correlations, hotspots, and potential occurrence of urban crimes". In: *Alexandria Engineering Journal* 60.1 (2021), pp. 1411–1420.

[Liu+21a] Qianhui Liu et al. "Event-based Action Recognition Using Motion Information and Spiking Neural Networks." In: *IJCAI*. 2021, pp. 1743–1749.

[Liu+21b] Yuanyuan Liu et al. "Expression Snippet Transformer for Robust Video-based Facial Expression Recognition". In: *arXiv preprint arXiv:2109.08409* (2021).

[Min+21] Shervin Minaee et al. “Image segmentation using deep learning: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3523–3542.

[MYC21] Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. “Event-intensity stereo: Estimating depth by the best of both worlds”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4258–4267.

[Orc+21] Garrick Orchard et al. “Efficient neuromorphic signal processing with loihi 2”. In: *2021 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE. 2021, pp. 254–259.

[PC21] Federico Paredes-Vallés and Guido CHE de Croon. “Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3446–3455.

[PP21] Christian Pehle and Jens Egholm Pedersen. *Norse - A deep learning library for spiking neural networks*. Version 0.0.7. Documentation: https://norse.ai/docs/. Jan. 2021. URL: https://doi.org/10.5281/zenodo.4422025.

[PZM21] Thomas Pellegrini, Romain Zimmer, and Timothée Masquelier. “Low-activity supervised convolutional spiking neural networks applied to speech commands recognition”. In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2021, pp. 97–103.

[Pou+21] Delphine Poux et al. “Dynamic Facial Expression Recognition under Partial Occlusion with Optical Flow Reconstruction”. In: *IEEE Transactions on Image Processing* 31 (2021), pp. 446–457.

[Rad+21] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[Ran+21] Ulysse Rançon et al. “StereoSpike: Depth Learning with a Spiking Neural Network”. In: *arXiv preprint arXiv:2109.13751* (2021).

[Rid+21] Tal Ridnik et al. “Imagenet-21k pretraining for the masses”. In: *arXiv preprint arXiv:2104.10972* (2021).

[Rya+21] Cian Ryan et al. “Real-time face & eye tracking and blink detection using event cameras”. In: *Neural Networks* 141 (2021), pp. 87–97.

[Tul+21] Stepan Tulyakov et al. “Time lens: Event-based video frame interpolation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 16155–16164.

[WL21] Feng Wang and Huaping Liu. "Understanding the behaviour of contrastive loss". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 2495–2504.

[Wan+21] Xiao Wang et al. "Visevent: Reliable object tracking via collaboration of frame and event flows". In: *arXiv preprint arXiv:2108.05015* (2021).

[WP21] Timo C Wunderlich and Christian Pehle. "Event-based backpropagation can compute exact gradients for spiking neural networks". In: *Scientific Reports* 11.1 (2021), p. 12829.

[Yao+21] Man Yao et al. "Temporal-wise attention spiking neural networks for event streams classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10221–10230.

[Zbo+21] Jure Zbontar et al. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". In: *arXiv preprint arXiv:2103.03230* (2021).

[ZYS21] Jiaming Zhang, Kailun Yang, and Rainer Stiefelhagen. "ISSAFE: Improving semantic segmentation in accidents by fusing event-based data". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1132–1139.

[Zha+21] Jiqing Zhang et al. "Object tracking by jointly exploiting frame and event domain". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13043–13052.

[Zhe+21] Hanle Zheng et al. "Going deeper with directly-trained larger spiking neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 11062–11070.

[Zhu+21] Alex Zihao Zhu et al. "Eventgan: Leveraging large scale image datasets for event cameras". In: *2021 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 2021, pp. 1–11.

[Zou+21] Shihao Zou et al. "Eventhpe: Event-based 3d human pose and shape estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10996–11005.

[All+22] B. Allaert et al. "A comparative study on optical flow for facial expression analysis". In: *Neurocomputing* 500 (2022), pp. 434–448.

[Bal+22] R Wes Baldwin et al. "Time-ordered recent event (TORE) volumes for event cameras". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.2 (2022), pp. 2519–2532.

[BMD22] Sami Barchid, José Mennesson, and Chaabane Djéraba. "Bina-rep event frames: A simple and effective representation for event-based cameras".

In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 3998–4002.

[BPL22] Adrien Bardes, Jean Ponce, and Yann LeCun. "VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning". In: *ICLR*. 2022.

[Bas+22] Arindam Basu et al. "Spiking neural network integrated circuits: A review of trends and future directions". In: *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE. 2022, pp. 1–8.

[BPD22] Federico Becattini, Federico Palai, and Alberto Del Bimbo. "Understanding human reactions looking at facial microexpressions with an event camera". In: *IEEE Transactions on Industrial Informatics* 18.12 (2022), pp. 9112–9121.

[Bon+22] Lina Bonilla et al. "Analyzing time-to-first-spike coding schemes: A theoretical approach". In: *Frontiers in Neuroscience* 16 (2022), p. 971937.

[Bou+22] Mohamed Sadek Bouanane et al. "Impact of spiking neurons leakages and network recurrences on event-based spatio-temporal pattern recognition". In: *arXiv preprint arXiv:2211.07761* (2022).

[CRR22] Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. "Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning". In: *European Conference on Computer Vision*. Springer. 2022, pp. 709–726.

[Cor22] Loïc Cordone. "Performance of spiking neural networks on event data for embedded automotive applications". PhD thesis. Université Côte d'Azur, 2022.

[CMT22] Loic Cordone, Benoît Miramond, and Philippe Thierion. "Object Detection with Spiking Neural Networks on Automotive Event Data". In: *arXiv preprint arXiv:2205.04339* (2022).

[Del+22] Tobi Delbruck et al. "Utility and feasibility of a center surround event camera". In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 381–385.

[Den+22] Yongjian Deng et al. "A Voxel Graph CNN for Object Classification with Event Cameras". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1172–1181.

[Eri+22] Linus Ericsson et al. "Self-supervised representation learning: Introduction, advances, and challenges". In: *IEEE Signal Processing Magazine* 39.3 (2022), pp. 42–62.

[GD22] Shasha Guo and Tobi Delbruck. "Low cost and latency event camera background activity denoising". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2022), pp. 785–795.

[GS22] Matthew Gwilliam and Abhinav Shrivastava. "Beyond supervised vs. unsupervised: Representative benchmarking and analysis of image representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9642–9652.

[Int22] Intel. *LAVA, a Software Framework for Neuromorphic Computing*. https://github.com/lava-nc/lava. 2022.

[Kaa+22] Lynn H Kaack et al. "Aligning artificial intelligence with climate change mitigation". In: *Nature Climate Change* 12.6 (2022), pp. 518–527.

[KMM22] Saeed Reza Kheradpisheh, Maryam Mirsadeghi, and Timothée Masquelier. "Spiking neural networks trained via proxy". In: *IEEE Access* 10 (2022), pp. 70769–70778.

[Kim+22a] Taewoo Kim et al. "Event-guided deblurring of unknown exposure time videos". In: *European Conference on Computer Vision*. Springer. 2022, pp. 519–538.

[Kim+22b] Youngeun Kim et al. "Neural architecture search for spiking neural networks". In: *European Conference on Computer Vision*. Springer. 2022, pp. 36–56.

[Kle+22] Simon Klenk et al. "Masked Event Modeling: Self-Supervised Pretraining for Event Cameras". In: *arXiv preprint arXiv:2212.10368* (2022).

[KB22] Ankith Jain Rakesh Kumar and Bir Bhanu. "Three Stream Graph Attention Network Using Dynamic Patch Selection for the Classification of Micro-Expressions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2476–2485.

[Lee+22] Alex Junho Lee et al. "ViViD++: Vision for visibility dataset". In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 6282–6289.

[Lem+22] Edgar Lemaire et al. "An analytical estimation of spiking neural networks energy efficiency". In: *International Conference on Neural Information Processing*. Springer. 2022, pp. 574–587.

[Li+22a] Yante Li et al. "Deep learning for micro-expression recognition: A survey". In: *IEEE Transactions on Affective Computing* (2022).

[Li+22b] Yuhang Li et al. "Neuromorphic data augmentation for training spiking neural networks". In: *Computer Vision–ECCV 2022: 17th European Confer-*

ence, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII. Springer. 2022, pp. 631–649.

[Lia+22] Zichen Liang et al. "Global-local feature aggregation for event-based object detection on eventkitti". In: *2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE. 2022, pp. 1–7.

[LD22] Min Liu and Tobi Delbruck. "EDFLOW: Event driven optical flow camera with keypoint detection and adaptive block matching". In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.9 (2022), pp. 5776–5789.

[Mai+22] Zheda Mai et al. "Online continual learning in image classification: An empirical survey". In: *Neurocomputing* 469 (2022), pp. 28–51.

[Men+22] Qingyan Meng et al. "Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12444–12453.

[Nae+22] Fariborz Baghaei Naeini et al. "Event augmentation for contact force measurements". In: *IEEE Access* 10 (2022), pp. 123651–123660.

[Nun+22] João D Nunes et al. "Spiking neural networks: A survey". In: *IEEE Access* 10 (2022), pp. 60738–60764.

[SGS22] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. "AEGNN: Asynchronous event-based graph neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12371–12381.

[She+22] Sadique Sheik et al. *Sinabs (Sinabs Is Not A Brain Simulator), A deep learning library for spiking neural networks which is based on PyTorch, focuses on fast training and supports inference on neuromorphic hardware.* https://github.com/synsense/sinabs. Accessed: 2023-08-01. 2022.

[SZZ22] Guobin Shen, Dongcheng Zhao, and Yi Zeng. "EventMix: An Efficient Augmentation Strategy for Event-Based Data". In: *arXiv preprint arXiv:2205.12054* (2022).

[SAG22] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. "Secrets of event-based optical flow". In: *European Conference on Computer Vision*. Springer. 2022, pp. 628–645.

[Sun+22] Zhaoning Sun et al. “Ess: Learning event-based semantic segmentation from still images”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 341–357.

[Tan+22a] Ganchao Tan et al. “Multi-grained spatio-temporal features perceived network for event-based lip-reading”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20094–20103.

[Tan+22b] Chuanming Tang et al. “Revisiting Color-Event based Tracking: A Unified Network, Dataset, and Metric”. In: *arXiv preprint arXiv:2211.11010* (2022).

[Tan+22c] Jianxiong Tang et al. “Relaxation LIF: A gradient-based spiking neuron for direct training deep spiking neural networks”. In: *Neurocomputing* 501 (2022), pp. 499–513.

[Wan+22a] Xiao Wang et al. “HARDVS: Revisiting Human Activity Recognition with Dynamic Vision Sensors”. In: *arXiv preprint arXiv:2211.09648* (2022).

[Wan+22b] Yuchen Wang et al. “Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion”. In: *International Joint Conference on Artificial Intelligence*. 2022.

[Wu+22] Xiaoshan Wu et al. “MSS-DepthNet: Depth Prediction with Multi-Step Spiking Neural Network”. In: *arXiv preprint arXiv:2211.12156* (2022).

[Xia+22] Shuiying Xiang et al. “Spiking vgg7: Deep convolutional spiking neural network with direct training for object recognition”. In: *Electronics* 11.13 (2022), p. 2097.

[Xie+22] Bochen Xie et al. “Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1976–1983.

[Yam+22] Kashu Yamazaki et al. “Spiking neural networks and their applications: A Review”. In: *Brain Sciences* 12.7 (2022), p. 863.

[Zha+22a] Anguo Zhang et al. “Event-Driven Intrinsic Plasticity for Spiking Convolutional Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.5 (2022), pp. 1986–1995.

[Zha+22b] Jiqing Zhang et al. “Spiking Transformers for Event-Based Single Object Tracking”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8801–8810.

[Zho+22] Zhaokun Zhou et al. “Spikformer: When Spiking Neural Network Meets Transformer”. In: *arXiv preprint arXiv:2209.15425* (2022).

[Zhu+22a] Lin Zhu et al. “Event-based video reconstruction via potential-assisted spiking neural network”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3594–3604.

[Zhu+22b] Rui-Jie Zhu et al. “Tcja-snn: Temporal-channel joint attention for spiking neural networks”. In: *arXiv preprint arXiv:2206.10177* (2022).

[Ber+23] Lorenzo Berlincioni et al. “Neuromorphic Event-based Facial Expression Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4108–4118.

[Bor+23] Chiara Boretti et al. “PEDRo: An Event-Based Dataset for Person Detection in Robotics”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4064–4069.

[Bul+23] Hugo Bulzomi et al. “End-to-End Neuromorphic Lip-Reading”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4100–4107.

[Cha+23] Kenneth Chaney et al. “M3ED: Multi-Robot, Multi-Sensor, Multi-Environment Event Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4015–4022.

[Cua+23] Javier Cuadrado et al. “Optical flow estimation from event-based cameras and spiking neural networks”. In: *Frontiers in Neuroscience* 17 (2023), p. 1160034.

[GS23] Mathias Gehrig and Davide Scaramuzza. “Recurrent vision transformers for object detection with event cameras”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13884–13893.

[GHM23] Yufei Guo, Xuhui Huang, and Zhe Ma. “Direct learning-based deep spiking neural networks: a review”. In: *Frontiers in Neuroscience* 17 (2023), p. 1209795.

[Hae+23] Germain Haessig et al. “PDAVIS: Bio-inspired Polarization Event Camera”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3962–3971.

[IKA23] Craig Iaboni, Thomas Kelly, and Pramod Abichandani. “NU-AIR–A Neuromorphic Urban Aerial Dataset for Detection and Localization of Pedestrians and Vehicles”. In: *arXiv preprint arXiv:2302.09429* (2023).

[Jin+23] Seong Min Jin et al. “BPLC+ NOSO: backpropagation of errors based on latency code with neurons that only spike once at most”. In: *Complex & Intelligent Systems* (2023), pp. 1–18.

[Kie+23] Paul Kielty et al. “Neuromorphic Driver Monitoring Systems: A Proof-of-Concept for Yawn Detection and Seatbelt State Detection using an Event Camera”. In: *IEEE Access* (2023).

[Liu+23] Yang Liu et al. “A survey of visual transformers”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[Mes+23] Nico Messikommer et al. “Data-driven feature tracking for event cameras”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 5642–5651.

[Niu+23] Ben Niu et al. “Lip print recognition based on convolutional spiking neural network”. In: *International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2023)*. Vol. 12707. SPIE. 2023, pp. 890–894.

[PI23] Goran Paulin and Marina Ivasic-Kos. “Review and analysis of synthetic dataset generation methods and techniques for application in computer vision”. In: *Artificial Intelligence Review* (2023), pp. 1–45.

[Qu+23] Jinye Qu et al. “Spiking Neural Network for Ultra-low-latency and High-accurate Object Detection”. In: *arXiv preprint arXiv:2306.12010* (2023).

[Ran+23] Veenu Rani et al. “Self-supervised learning: A succinct review”. In: *Archives of Computational Methods in Engineering* 30.4 (2023), pp. 2761–2775.

[Saj+23] Muhammad Sajjad et al. “A comprehensive survey on deep facial expression recognition: challenges, applications, and future guidelines”. In: *Alexandria Engineering Journal* 68 (2023), pp. 817–840.

[SF23] Nikolaus Salvatore and Justin Fletcher. “Dynamic Vision-Based Satellite Detection: A Time-Based Encoding Approach with Spiking Neural Networks”. In: *Computer Vision Systems*. Ed. by Henrik I. Christensen et al. Cham: Springer Nature Switzerland, 2023, pp. 285–298.

[Sam+23] Ali Samadzadeh et al. “Convolutional spiking neural networks for spatio-temporal feature extraction”. In: *Neural Processing Letters* (2023), pp. 1–17.

[Tou+23] Hugo Touvron et al. “Llama: Open and efficient foundation language models”. In: *arXiv preprint arXiv:2302.13971* (2023).

[Wan+23] Xueyi Wang et al. “Fall detection with event-based data: A case study”. In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2023, pp. 33–42.

[Xu+23] Qi Xu et al. “Constructing deep spiking neural networks from artificial neural networks with knowledge distillation”. In: *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 7886–7895.

[YPL23] Yan Yang, Liyuan Pan, and Liu Liu. "Event Camera Dat a Pre-training". In: *arXiv preprint arXiv:2301.01928* (2023).

[Yao+23] Man Yao et al. "Spike-driven Transformer". In: *arXiv preprint arXiv:2307.01694* (2023).

[Zha+23] Qiugang Zhan et al. "Bio-inspired Active Learning method in spiking neural network". In: *Knowledge-Based Systems* 261 (2023), p. 110193.

[Zhe+23] Xu Zheng et al. "Deep learning for event-based vision: A comprehensive survey and benchmarks". In: *arXiv preprint arXiv:2302.08890* (2023).

[Zho+23] Chenlin Zhou et al. "Enhancing the Performance of Transformer-based Spiking Neural Networks by Improved Downsampling with Precise Gradient Backpropagation". In: *arXiv preprint arXiv:2305.05954* (2023).

[Zou+23a] Shihao Zou et al. "Event-based human pose tracking by spiking spatiotemporal transformer". In: *arXiv preprint arXiv:2303.09681* (2023).

[Zou+23b] Zhengxia Zou et al. "Object detection in 20 years: A survey". In: *Proceedings of the IEEE* (2023).

Liste des Travaux

Conférences internationales à comité de lecture

- Sami Barchid, José Mennesson, and Chaabane Djéraba. “Deep spiking convolutional neural network for single object localization based on deep continuous local learning”. In: *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE. 2021, pp. 1–5.
- Sami Barchid, José Mennesson, and Chaabane Djéraba. “Review on indoor RGB-D semantic segmentation with deep convolutional neural networks”. In: *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE. 2021, pp. 1–4.
- Sami Barchid, José Mennesson, and Chaabane Djéraba. “Bina-rep event frames: A simple and effective representation for event-based cameras”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 3998–4002.
- Sami Barchid, José Mennesson, and Chaabane Djéraba. “Exploring Joint Embedding Architectures and Data Augmentations for Self-Supervised Representation Learning in Event-Based Vision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3902–3911.
- Sami Barchid et al. “Spiking-Fer: Spiking Neural Network for Facial Expression Recognition With Event Cameras”. In: *Proceedings of the 2023 20th International Conference on Content-based Multimedia Indexing*. CBMI ’23. Orléans, FR: Association for Computing Machinery, 2023.

Journaux internationaux à comité de lecture

- Sami Barchid et al. “Spiking neural networks for frame-based and event-based single object localization”. In: *Neurocomputing* 559 (2023), p. 126805. URL: https://www.sciencedirect.com/science/article/pii/S0925231223009281.

Publications

2021

Review on indoor RGB-D Semantic Segmentation with Deep Convolutional Neural Networks

SCAN ME

Deep Spiking Convolutional Neural Network for Single Object Localization based on Deep Continuous Local Learning

SCAN ME

2022

Bina-Rep Event Frames: A Simple and Effective Representation for Event-based Cameras

SCAN ME

Spiking Neural Networks for Frame-based and Event-based Single Object Localization

SCAN ME

2023

Spiking-Fer: Spiking Neural Network for Facial Expression Recognition With Event Cameras

SCAN ME

Exploring Joint Embedding Architectures and Data Augmentations for Self-Supervised Representation Learning in Event-Based Vision

SCAN ME