# Classification ensembliste de vidéos de mouvements de foule

Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller

# Classification ensembliste de vidéos de mouvements de foule

M. Bendali-Braham J. Weber G. Forestier L. Idoumghar P.-A. Muller

Institut IRIMAS, Université de Haute-Alsace

12 Rue des Frères Lumière, 68093 Mulhouse
prénom.nom@uha.fr

## Résumé

*Les méthodes ensemblistes améliorent souvent les résultats dans des problèmes traités par des méthodes individuelles issues de l'apprentissage automatique. Partant de ce constat, nous appliquons la classification ensembliste aux vidéos de foule. Dans un premier temps, nous constituons des ensembles de modèles homogènes pour comparer leurs performances sur le jeu de données Crowd-11. Dans un second temps, nous évaluons toutes les combinaisons possibles de ces ensembles, et nous analysons la combinaison des ensembles qui obtient les meilleurs résultats.*

## Mots Clef

Classification ensembliste, Apprentissage profond, Analyse de vidéos, Analyse de comportements de foule.

## Abstract

*Ensemble methods often improve results in problems addressed by single machine learning methods. Based on this observation, we use ensemble classification. First, we build ensembles of homogeneous models to compare their performance on the Crowd-11 dataset. Secondly, we evaluate all the possible combinations of these ensembles, and we analyze the combination of ensembles that achieves the best results.*

## Keywords

Ensemble classification, Deep learning, Video analysis, Crowd behavior analysis.

## 1 Introduction

De plus en plus de villes françaises sont équipées en caméras de vidéo-protection [1]. Certaines de ces villes sont sujettes à des mouvements de foule massifs dus à des manifestations à caractère culturel ou politique [2]. Afin de permettre un déroulement pacifique et en toute sécurité de ces manifestations, les forces de l'ordre ont recours à diverses stratégies de gestion de mouvements de foule. Celles-ci ont beaucoup évolué au cours des dernières décennies suite à diverses décisions politiques nationales et directives européennes [3].

Lors d'un mouvement de foule, les forces de l'ordre s'appuient sur la vidéo-surveillance et peuvent compter sur des approches récentes permettant un déploiement optimal de caméras [4] ou de drones de vidéo-protection [5]. Toutefois, même si la collecte des données vidéo est de plus en plus répandue, l'analyse automatique de ces vidéos n'est pas faite en temps réel, de manière systématique, ce qui peut retarder la réaction des forces de l'ordre. L'une des raisons à cela est l'inexistence de modèles statistiques génériques pouvant être utilisés en temps réel pour détecter n'importe quel type d'anomalies pouvant survenir lors d'un mouvement de foule [6]. Par ailleurs, la recherche éprouve des difficultés à développer de tels modèles. Cela peut s'expliquer par la rareté de données annotées [7]. Toutefois, les dernières années ont vu l'émergence de jeux de données illustrant des mouvements de foule massifs et variés tels que le jeu de données Crowd-11 [8].

Les auteurs de Crowd-11 ont entraîné des modèles à classifier des scènes de foule sur le jeu de données. Le modèle qui obtient les meilleurs résultats de classification dans leur article est un modèle issu de l'architecture C3D [9]. Dans de précédents travaux, nous avons obtenu de meilleures performances [10], en employant un modèle issu de l'architecture TwoStream Inflated 3D (2S-I3D) ayant dépassé les performances des modèles C3D sur des jeux de données de reconnaissance d'actions [7].

Dans cet article, nous visons à améliorer les performances de classification sur Crowd-11, en constituant des Ensembles de modèles. Nous constituons, dans un premier temps, des ensembles de modèles homogènes provenant de différentes architectures et disposant de différentes pré-conditions d'entraînement. L'aspect théorique de cette comparaison est développée dans la section 4.1. Dans un second temps, une évaluation de toutes les combinaisons possibles de ces modèles, nous permet d'élire un ensemble global rassemblant des ensembles de modèles hétérogènes. L'aspect théorique de cette combinaison est développée dans la section 4.3.

Cet article est organisé comme suit : dans la Section 2, nous présentons des méthodes ensemblistes appliquées à la classification de vidéos de manière générale ou plus spécifiquement à l'analyse de foules. Dans la Section 3, nous

présentons nos approches. Nous discutons les expériences que nous avons menées dans la Section 4.

## 2 État-de-l'art

Les méthodes ensemblistes réalisent de très bonnes performances dans plusieurs tâches liées à l'apprentissage automatique [11]. Zhou [12] divise les méthodes ensemblistes en trois grandes familles représentatives, qu'ils présentent comme suit :

- Boosting, illustrée par son algorithme le plus connu AdaBoost [13], qui consiste à apprendre $T$ modèles en associant à chaque fois des poids différents aux exemples d'apprentissage. Au début similaires, ces poids changent à chaque itération $t$ de l'algorithme AdaBoost en prenant en compte l'erreur obtenue d'un modèle entraîné à l'itération $t-1$. À la fin, un vote majoritaire pondéré est utilisé pour combiner les décisions des $T$ modèles.
- Bagging, contraction de Bootstrap Aggregating, où des méthodes statistiques sont entraînées sur des échantillons créés par des échantillonnages Bootstrap [14]. Par la suite, ces méthodes sont combinées dans un ensemble par un vote majoritaire.
- Stacking, où différentes méthodes statistiques sont entraînées sur un jeu de données. Par la suite, une seconde méthode statistique, appelée méta-classifieur, apprend à combiner les modèles entraînés.

Par ailleurs, Zhou et al. envisagent que des méthodes ensemblistes ne fassent partie d'aucune de ces trois grandes catégories.

L'apparition des premières méthodes ensemblistes en apprentissage automatique supervisé remonte aux années 1970s [15]. Plusieurs articles d'état-de-l'art ont présenté les méthodes ensemblistes [12, 15, 16]. Nous explorons ici quelques approches ensemblistes récentes liées à l'analyse d'images et de vidéos, et à l'analyse de foule.

Dans le cadre de l'analyse d'images et de vidéos, Lia et al. [17] appliquent une méthode ensembliste pour apporter une solution aux classes faiblement pourvues dans un jeu de données de classification des images de véhicules. Pour ce faire, ils appliquent un échantillonnage équilibré et l'augmentation de données. Leur méthode ensembliste consiste en une combinaison des décisions de multiples modèles ResNets [18] (ResNet-50, ResNet-101, et ResNet-152, pré-entraînés sur ImageNet [19]), en utilisant le vote majoritaire.

Pouyanfar et al. [20] proposent EDL "Ensemble Deep Learning" qu'ils utilisent pour la classification de vidéos sur les jeux de données Trecvid [21] et Disaster [22]. EDL est une suite de modèles profonds extracteurs de caractéristiques sur des images, qui sont des Réseaux de Neurones à Convolutions (CNNs) [23] pré-entrainés sur ImageNet, et chaque extracteur est chapeauté par une machine à vecteurs de support (SVM) [24] qui sert d'apprenant faible dans un ensemble de modèles. Les caractéristiques sont extraites de la dernière couche Fully Convolutional Network (FCN) de chaque modèle. Les architectures utilisées sont : AlexNet [25], CaffeNet [26], R-CNN (Region based CNN) [27], GoogleNet [28], ResNet, la combinaison des décisions se fait via un vote pondéré.

Inspirés par Liu et al. [29], Chen et al. [30] utilisent une méthode ensembliste qu'ils ont nommée EnwMi (pour Ensemble Weighted Multi-Instance Learning). Ils commencent par échantillonner plusieurs sous-ensembles de la classe majoritaire, et en combinant à chaque fois un sous-ensemble de la classe majoritaire avec la totalité de la classe minoritaire, ils entraînent, en utilisant AdaBoost [13], un modèle. Les modèles entraînés sont combinés pour la décision finale.

Dans le cadre de l'analyse de foule, Walach et al. [31] appliquent du gradient boosting et l'échantillonnage sélectif sur une architecture simple de CNNs pour compter le nombre d'objets dans une image. Leur approche est appliquée sur des jeux de données de comptage de cellules microscopiques de bactéries, et sur des jeux de données de comptage des statistiques de foule.

Wu et al. [32] empilent plusieurs modèles (stacking) dont les résultats sont considérés comme de nouvelles caractéristiques qui seront utilisées en entrée pour un nouveau modèle.

Au vu du peu de données annotées, Gong et al. [33] apprennent, dans un contexte semi-supervisé, un ensemble de pose-sensitive DPM (Deformable Part-based Model) mixtures [34] pour la détection de personnes dans une scène quelque soit leur posture. Les classes de postures considérées sont : front, rear, left, right. Chaque DPM mixture, sensible à une posture spécifique, est entraîné par un Latent-SVM [35].

Contrairement à des approches précédemment citées [17, 29], nous ne visons pas à résoudre le problème des données non équilibrées. Nous ne faisons pas de Boosting, comme dans Walach et al. [31], car dans le Boosting les entraînements sont répétés plusieurs fois en changeant la pondération des exemples d'apprentissage, or un seul entraînement sur des vidéos requiert déjà un temps de calcul de plusieurs heures. Répéter plusieurs entraînements nécessiterait, dans ces conditions, plusieurs jours. Nous optons pour un compromis entre une forme de Stacking [32], sans méta-classifieur car nous combinons les modèles non pas à l'entraînement mais lors de la prédiction, et une forme de Bagging, car nous réalisons une agrégation de modèles sans recourir à l'échantillonnage Bootstrap. Ici, les échantillons sont déjà découpés pour la validation croisée. Ce découpage est stratifié, car chaque échantillon respecte la distribution des classes du jeu de données original. Nous ne faisons pas d'apprentissage semi-supervisé combiné à l'usage de méthodes ensemblistes, comme dans Gong et al. [33], car nous n'avons pas de problème de manque d'annotation dans les données que nous utilisons. Tous les vidéo-clips y sont annotés.

Zhou et al. [36] soutiennent qu'il n'est pas utile de mettre

un très grand nombre de modèles dans un ensemble. Un choix d'un petit nombre de modèles qui produisent déjà de bonnes performances, suffit à produire de meilleures performances lorsqu'ils sont combinés dans un ensemble. De ce fait, nous avons décidé de découper le jeu de données en 5 échantillons, tel qu'il a été déjà été réalisé dans Bendali-Braham et al. [10], ce qui permet de doter chaque ensemble de 4 modèles individuels qui extraient des connaissances différentes du jeu de données Crowd-11.

# 3 Classification ensembliste

Au vu des approches citées dans la Section 2, et en nous basant sur la définition proposée par Zhou [12], l'apprentissage ensembliste consiste à entraîner, ou évaluer, un ensemble de méthodes statistiques, qu'elles soient de même nature ou non, entraînées dans des conditions semblables ou non.

En application à l'esprit de la définition proposée par Zhou [12], nous mettons en place, dans un premier temps, des approches ensemblistes constituées de modèles homogènes disposant des mêmes conditions de pré-entraînement. Dans un second temps, nous proposons de mettre en place des approches ensemblistes globales qui mêlent des modèles hétérogènes ayant différentes architectures et disposant de conditions de pré-entraînement différentes.

Les probabilités prédites par chaque modèle sont combinées au niveau de chaque Ensemble à l'aide de la somme.

Dans cet article, - quand nous parlons d'un ensemble homogène de modèles, ces modèles disposent d'une même architecture et ont été soient entraînés ou ajustés, par exemple : un ensemble de modèles C3D ajustés ; - quand nous parlons d'un ensemble global de modèles hétérogènes, cet ensemble global dispose de plusieurs groupes différents d'ensembles de modèles homogènes, par exemple : un ensemble global constitué d'un ensemble de modèles C3D ajustés et un ensemble de modèles I3D entraînés de zéro.

**Composition d'ensembles de modèles homogènes.** Dans de précédents travaux sur le jeu de données Crowd-11, nous avions montré que les modèles issus du réseau TwoStream Inflated 3D (2S-I3D) obtiennent de meilleures performances que les modèles issus du réseau 3D ConvNets (C3D) et du réseau à branche unique I3D [10]. Toutefois les performances des réseaux plafonnent en moyenne à 68% d'accuracy. Ces résultats ont été confirmés par une validation croisée à 5 échantillons.

Dans cet article, nous redécoupons le jeu de données en 5 échantillons, et nous entraînons, pour chaque combinaison possible des échantillons, un modèle issu d'une des architectures suivantes :

- L'architecture 2S-I3D [7],
- L'architecture I3D [7].
- L'architecture C3D [9].
- L'architecture Resnet 3D (R3D) [37].

Dans chaque combinaison, 3 échantillons sont dédiés à l'apprentissage, 1 à la validation, et 1 au test. En sélectionnant à chaque fois, un échantillon de test, nous pouvons produire 4 modèles sur les combinaisons des échantillons restants. Lors de l'évaluation sur un échantillon de test, les décisions des 4 modèles sont combinées pour former un ensemble de modèles.

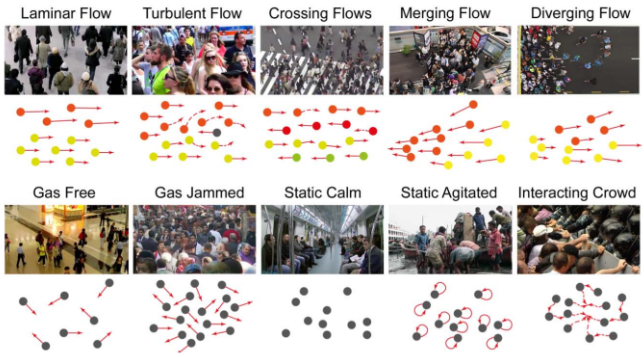


FIGURE 1 – Illustration de chaque classe de comportement de foule tirée du travail de Dupont et al. [8]

Avec cette opération, nous nous retrouvons avec 20 combinaisons différentes des ensembles d'apprentissage, de validation, et de test.

Nous rappelons dans les paragraphes qui suivent la constitution de chacune des architecture 2S-I3D, I3D, C3D, et nous vous présentons l'architecture R3D.

L'architecture I3D est composée d'une base de 2 couches de réseaux à convolutions 3D (3D ConvNets), chacune appuyée par une normalisation par lots et suivie d'une opération de MaxPooling 3D. Ces 2 couches sont suivies par 9 modules Inception dont la composition interne change légèrement d'un module à un autre. Le dernier module Inception est connecté à un AveragePooling 3D qui est relié à un SoftMax de classification.

L'architecture TwoStream-I3D est composée de deux branches. Chaque branche reprend l'architecture du réseau I3D. Une des deux branches extrait des caractéristiques d'un clip vidéo RVB, et une autre extrait les caractéristiques d'un clip vidéo en flot optique. Les sorties de ces deux branches sont connectées à la fonction de classification Softmax.

L'architecture C3D, proposée par Tran et al. [9], est constituée de 5 couches de convolutions 3D, suivies de deux couches de FCN et d'une fonction de classification Softmax.

L'architecture R3D, proposée par Hara et al. [37], est constituée de plusieurs blocs résiduels. Chaque bloc résiduel est constitué de deux couches de convolutions 3D. Nous choisissons la version à 34 couches cachées du fait de ses bonnes performances qui sont démontrées par Hara et al. [37] sur le jeu de données de reconnaissance d'actions Sports-1m. Cette version de l'architecture R3D est constituée d'une première couche de convolutions 3D suivie de 16 blocs résiduels, et se termine par une couche FCN avant la fonction de classification Softmax.

**Constitution des ensembles d'apprentissage, de validation, et de test.** La version du jeu de données Crowd-11 dont nous disposons correspond à la version sur laquelle nous avons déjà travaillé dans un article précédent [10]. Cette version du jeu de données est constituée de 1641 scènes que l'on peut considérer comme des frontières contextuelles entre les clips vidéo. De ces scènes, proviennent 5769 vidéo clips qui sont classés en 11 classes présentées dans Dupont et al. [8]. Ces 11 classes, illustrées dans la Figure 1, sont : Gas Free, Gas Jammed, Laminar Flow, Turbulent Flow, Crossing Flows, Merging Flows, Diverging Flow, Static Calm, Static Agitated, Interacting Crowd, No Crowd.
Afin de créer des échantillons devant participer à la réalisation de la validation croisée, dans Bendali-Braham et al. [10], nous avons découpé les échantillons en partant des scènes. Pour maintenir la diversité et une quantité similaire de vidéos entre les échantillons nous avons appliqué les algorithmes 1 et 2. Ces deux algorithmes servent à constituer les échantillons qui seront utilisés pour la création des ensembles d'apprentissage, de validation, et de test, telle qu'elle est illustrée dans la Figure 2 [1].
Cette opération a permis de découper le jeu de données en 5 échantillons.

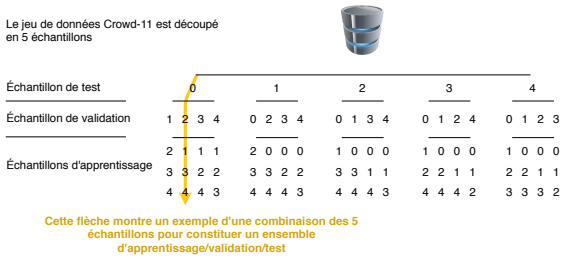


FIGURE 2 – Illustration de la procédure de constitution des ensembles d'apprentissage, de validation, et de test à partir de différentes combinaisons des échantillons résultant du découpage du jeu de données

**Ensembles globaux de modèles hétérogènes.** Nous créons des ensembles globaux de modèles ayant soient différentes architectures, par exemple des ensembles 2S-I3D ajustés couplés avec des ensembles C3D ajustés, soient différentes conditions d'entraînement, par exemple des ensembles I3D ajustés et I3D entraînés de zéro, ou divers ensembles de modèles cumulant les deux différences, par exemple des ensembles C3D entraînés de zéro, des ensembles I3D ajustés, et des ensembles 2S-I3D ajustés. Nous composons des ensembles globaux à partir des ensembles de modèles homogènes comparés dans la section 4.1. **Par la suite, nous évaluons sur l'ensemble de test toutes les combinaisons possibles à partir de ces ensembles de modèles homogènes.** L'équation 1 calcule le

1. Le code source de ce travail est disponible ici : https://github.com/MounirB/Crowded-scenes-Ensemble-classification

**Données :** Pré-calculer $Sc\_freq$ et $Cls\_freq$
$Sc$ : liste des scènes;
$Nb\_echs$ : nombre des échantillons;
$Sc\_freq$ : nombre de vidéos par scène;
$Cls\_freq$ : nombre de vidéos par classe;
**Résultat :** Découpage du jeu de données en scènes $Echs\_scenes$.
$Echs\_scenes$ : listes de scènes pour tous les échantillons initialisés à des listes vides;
$Echs\_distribs$ : distributions des scores des échantillons en fonction des vidéos qu'ils contiennent et de la diversité de ces dernières;
**tant que** *$Sc$ n'est pas vide* **faire**
  Sélectionner l'échantillon avec le score le plus faible à partir de $Echs\_distribs$;
  Sélectionner la scène qui contient le plus de vidéos $Sc\_freq$;
  Supprimer la scène sélectionnée de $Sc\_freq$ et $Sc$;
  Ajouter la scène sélectionnée à l'échantillon sélectionné dans $Echs\_scenes$;
  Mettre à jour le score de l'échantillon sélectionné dans $Echs\_distribs$;
**fin**
**Algorithme 1 :** Découper le jeu de données en échantillons

**Données :** En arguments : $Ech\_distrib$, $s$, $Nb\_echs$, $Cls\_freq$
$s$ : scène précédemment sélectionnée;
$Vids$ : liste de toutes les vidéos;
$Database$ : informations sur le jeu de données qui lient des scènes à leurs vidéos;
**Résultat :** Nouveau score de l'échantillon $Ech\_distrib$ dans $Echs\_distribs$
$Sc\_vids$ : jointure entre la scène $s$ et $Database$, et récupération des vidéos de la scène à partir de $Vids$;
$Sc\_classes$ : établir la liste des classes présentes dans la scène $s$ à partir de $Sc\_vids$;
**pour chaque** *classe $c$ dans $Sc\_classes$* **faire**
  $Ech\_distrib_c = Ech\_distrib_c + \frac{Nb\_echs}{Cls\_freq_c}$
**fin**
**Algorithme 2 :** Mettre à jour le score d'un échantillon

nombre de combinaisons sans répétition pouvant donner lieu à des ensembles globaux de modèles.

$$nb\_combinaisons = \sum_{i=2}^{K} C(K, i) \quad (1)$$

Où $K$ représente la taille maximale d'une combinaison constituant un ensemble global. $C(K, i)$ représente la fonction de calcul d'une combinaison sans répétition où le nombre de choix est $i$. $i$ est la longueur du tuple qui représente le nombre d'ensembles de modèles homogènes donnant lieu à une combinaison d'un ensemble global de

modèles. Comme nous évaluons déjà les ensembles de modèles homogènes dans la section 4.1, $i$ commence à partir de 2 qui est considéré comme la taille minimale d'une combinaison.

# 4 Expériences

Dans cette section, nous détaillons les différents types d'expériences que nous avons menées :

— Nous avons comparé les performances d'Ensembles formés de modèles individuels pré-entraînés à d'autres Ensembles formés de modèles individuels qui n'ont pas bénéficié du pré-entraînement ;

— Nous avons comparé des ensembles de modèles ayant bénéficié d'un entraînement sur le jeu de données Crowd-11 augmenté et comparé à des modèles n'ayant pas bénéficié d'un entraînement sur le jeu de données augmenté ;

Dans ce travail, les hyperparamètres choisis pour les apprentissages correspondent aux hyperparamètres que nous avons utilisés dans Bendali-Braham et al. [10]. Les clips du jeu de données Crowd-11 durent en moyenne $\approx 5$ secondes. Pour les architectures I3D et 2S-I3D, 20 images sont sélectionnées d'un clip vidéo. Ces images se trouvent à intervalles réguliers tout au long du clip. La taille de chaque image est de $224 \times 224$ pixels. Pour les architectures R3D et C3D, 16 images sont sélectionnées d'un clip vidéo et la taille de chaque image est de $112 \times 112$ pixels. Dans les expériences de comparaisons de modèles ajustés avec les modèles non ajustés, la version flot optique de chaque clip est obtenue via l'algorithme TV-L1 [38]. Dans les expériences portant sur la comparaison de modèles ayant bénéficié ou non d'une augmentation des données, nous allons tester l'usage de l'algorithme d'extration de flot optique Farnebäck du fait de sa rapidité par rapport à l'algorithme TV-L1. Nous vérifierons également si le recours à l'algorithme Farnebäck ne réduit pas beaucoup les performances des modèles 2S-I3D.

## 4.1 Comparaison d'ensembles de modèles ayant des architectures homogènes

Nous souhaitons vérifier si un ensemble de modèles ajustés est plus performant qu'un ensemble de modèles entraînés de zéro, et quelle condition de pré-entraînement favorise le mieux la création d'ensembles de modèles. Les modèles 2S-I3D et I3D ajustés sur Crowd-11, ont été pré-entraînés sur les jeux de données ImageNet [19] et Kinetics [39]. Les modèles C3D ajustés sur Crowd-11, ont été pré-entraînés sur le jeu de données Sports-1m [40]. D'un autre côté, nous entraînons de zéro sur Crowd-11 les modèles 2S-I3D, I3D, C3D, et R3D n'ayant pas bénéficié d'un pré-entraînement. Dans toutes les situations, nous créons 5 grands contextes d'apprentissage, de validation et de test, où nous fixons un échantillon de test, en amont, et nous varions les échantillons de validation (différents du test), en aval. Dans ces conditions, pour chaque ensemble de modèles, nous réservons trois échantillons, différents du test et à chaque fois différents de la validation, pour l'apprentissage de 4 modèles individuels. Au total, pour les 5 échantillons de test, 20 modèles individuels sont appris. Chaque groupe de 4 modèles individuels, qu'ils soient ajustés ou entraînés de zéro, forme un ensemble de modèles.
Les résultats des prédictions de ces modèles sont illustrées dans la Table 1. À partir de cette table, nous voyons globalement que la constitution de modèles ensemblistes à partir des modèles individuels accroît l'accuracy lors de la classification.

## 4.2 Augmentation des données

Nous avons mené des expériences en augmentant les données vidéo afin de tester l'apport de l'augmentation de données sur la classification des vidéos de mouvements de foule. L'augmentation de données n'a été appliquée que sur les données de l'ensemble d'apprentissage du jeu de données Crowd-11. Le type d'opérations d'augmentation de données effectuées sont : découpage aléatoire (Random Crop), effet poivre et sel (Salt and Pepper), effet miroir vertical (Video Flip). Dans nos expériences, nous comparons deux stratégies d'augmentation de données :

— Une augmentation de données fixe pré-calculée avant le lancement de la phase d'entraînement. Les méthodes d'augmentation choisies aléatoirement de manière exclusive ou combinée avec d'autres méthodes d'augmentation. Suite à l'augmentation de données, la taille de l'ensemble d'apprentissage a été augmenté 3 fois. À chaque époque de la phase d'entraînement, le modèle explore les données augmentées ainsi que les données non augmentées.

— Une augmentation de données à-la-volée qui se renouvelle à chaque fois lors de la phase d'entraînement. Nous accordons à cette augmentation $75\%$ de probabilité de se produire à chaque itération pendant les époques d'entraînement. Les époques sont répétés 4 fois afin de permettre à l'augmentation de données à-la-volée de correspondre à l'augmentation de données pré-calculée en termes de quantité et de variété des données augmentées et non augmentées.

Préalablement à l'augmentation des données, nous avons décidé de changer l'agorithme d'extraction du flot optique TVL1 par l'algorithme de Färneback. Ce changement est motivée par la vitesse d'extraction de l'algorithme de Färneback qui est théoriquement connu pour être moins précis que l'algorithme TVL1.

**Discussion des résultats de l'augmentation des données**
À la suite de nos expériences, dont les résultats sont illustrées dans la Table 2, nous constatons que le recours à l'algorithme d'extraction Farnebäck réduit légèrement les performances des modèles 2S-I3D. La moyenne des scores passe de $69.02\%$ à $68.41\%$. Cette réduction est toutefois minime et peut être négligée pour la suite des expériences illustrées dans la Table 2. L'augmentation des données cal-

| **Échantillon de test** | **0** | **1** | **2** | **3** | **4** | $\mu$ | $\sigma$ |
|---|---|---|---|---|---|---|---|
| **Éch de val : accuracy par modèle individuel associé** | 1 : 67.86 | 0 : 66.55 | 0 : 66.04 | 0 : 63.09 | 0 : 69.73 | | |
| | 2 : 69.08 | 2 : 66.55 | 1 : 67.28 | 1 : 66.26 | 1 : 70.60 | | |
| | 3 : 67.33 | 3 : 66.29 | 3 : 68.52 | 2 : 63.52 | 2 : 69.30 | | |
| | 4 : 69.86 | 4 : 65.44 | 4 : 66.66 | 4 : 62.15 | 3 : 67.39 | | |
| **Écart-type des accuracies par échantillon de test** | 0.99 | 0.45 | 0.91 | 1.52 | 1.17 | | |
| **Moyenne des accuracies par échantillon de test** | 68.53 | 66.21 | 67.13 | 63.76 | 69.26 | 66.98 | **1.92** |
| **Accuracy par ensemble** | **70.48** | **67.57** | **68.61** | **66.43** | **72.00** | **69.02** | 1.99 |

TABLE 1 – Comparaison entre les résultats obtenus par les ensembles de modèles 2S-I3D ajustés et leurs modèles individuels

| **Échantillon de test** | **0** | **1** | **2** | **3** | **4** | $\mu$ | $\sigma$ |
|---|---|---|---|---|---|---|---|
| Accuracy par ensemble FarneBack Non Augmented | 70.56 | 66.55 | 69.93 | 64.21 | 70.78 | 68.41 | 2.59 |
| Accuracy par ensemble FarneBack Augmented Precomputed | **71.00** | **69.10** | **71.88** | **65.58** | **71.47** | **69.81** | 2.31 |
| Accuracy par ensemble FarneBack Augmented On The Fly | 68.47 | 67.23 | 69.23 | 64.21 | 69.30 | 67.69 | **1.89** |

TABLE 2 – Comparaison des performances des ensembles par échantillon de test avec ou sans augmentation de données

culée à la volée n'améliore pas les résultats. Pire : elle réduit les performances des modèles 2S-I3D. L'augmentation des données pré-calculée améliore considérablement les résultats des modèles 2S-I3D ajustés alimentés dans leurs secondes branches par des clips en flot optique extraits à l'aide de l'algorithme Farnebäck. Ces ensembles progressent même, en moyenne, d'1 point d'accuracy passant de $68.41\%$ à $69.81\%$ battant ainsi les modèles 2S-I3D ajustés dont les secondes branches ont été alimentés par les clips en flot optique extraits à l'aide de l'algorithme TVL1. Ces bonnes performances sont contrebalancées par le temps d'entraînement requis par les modèles bénéficiant de l'augmentation des données pré-calculée. Ce temps d'entraînement est approximativement 5 fois plus important que le temps d'entraînement des modèles dont les secondes branches sont alimentés par les clips en flot optique extraits à l'aide de TVL1.

### 4.3 Évaluation des ensembles globaux de modèles avec des architectures hétérogènes

Nous constituons ces ensembles globaux en combinant les modèles d'ensembles homogènes illustrés dans la Table 3. Ces 8 ensembles peuvent donner lieu à 247 combinaisons formant des ensembles globaux de modèles hétérogènes.

La combinaison qui obtient les meilleures résultats associe les ensembles 2S-I3D ajustés ayant bénéficié d'un entrainement sur les données augmentées pré-calculées de Crowd-11, avec les ensembles 2S-I3D ajustés n'ayant pas bénéficié d'un entraînement sur des données augmentées, les ensembles C3D ajustés, et les ensembles I3D entrainés de zéro. Les résultats de cette combinaison sont illustrés dans la Table 4. Nous constatons que cette combinaison améliore les performances globales de $1.5\%$ en termes d'accuracy.

Les performances illustrées dans la Table 4 montrent que les ensembles se complètent dans la classification. Dans ce contexte, les modèles les moins performants ne représentent pas un frein dans les performances de l'ensemble global lors de la classification.

## 5 Conclusions et perspectives

Nous avons constaté au cours de nos expériences que l'ensemble de modèles 2S-I3D ajustés améliorent les résultats de leurs modèles individuels qui ont été entraînés sur des échantillons d'apprentissage différents.

Nous avons exploré plusieurs ensembles de modèles à partir d'architectures différentes. Tout d'abord, nous avons créé des ensembles de modèles homogènes, et les avons comparés. Par la suite, nous avons augmenté les données et comparé deux approches d'augmentation des données sur les modèles issues de l'architecture 2S-I3D. Nous avons vu que l'augmentation des données pré-calculée est la méthode de régularisation permettant aux modèles de mieux généraliser. Enfin, nous avons évalué toutes les combinaisons possibles des modèles ensemblistes homogènes donnant lieu à un ensemble global de modèles hétérogènes, et nous avons analysé la meilleure combinaison possible constituée par 4 ensembles de modèles ayant différentes architectures et différentes pré-conditions d'entraînement.

Actuellement, nos modèles ne permettent pas des prédictions en temps réel. Pour y remédier, nous souhaitons nous focaliser dans nos futurs travaux sur le transfert des connaissances apprises par nos ensembles vers un réseau plus léger, qui serait permettrait idéalement une prédiction en temps réel, en appliquant la technique de la distillation des connaissances [41].

## Remerciements

Les auteurs tiennent à remercier NVIDIA Corporation pour nous avoir fourni des GPUs et le Mésocentre de Strasbourg pour leur avoir permis de mener des calculs sur le cluster de GPUs. Ce travail a été soutenu par le projet ANR OPMoPS (subvention ANR-16-SEBM-0004) financé par l'Agence nationale de la recherche.

| **Échantillon de test** | **0** | **1** | **2** | **3** | **4** | $\mu$ | $\sigma$ |
|---|---|---|---|---|---|---|---|
| C3D scratch | 31.26 | 32.76 | 32.27 | 31.76 | 38.08 | 33.23 | 2.47 |
| C3D pretrained | 61.13 | 60.59 | 61.00 | 58.13 | 61.56 | 60.48 | **1.21** |
| I3D scratch | 54.93 | 55.91 | 58.53 | 53.85 | 58.86 | 56.42 | 1.97 |
| I3D pretrained | 64.10 | 60.25 | 62.24 | 57.70 | 60.95 | 61.05 | 2.12 |
| R3D (w 34 layers) scratch | 47.42 | 52.00 | 50.13 | 48.63 | 50.43 | 49.72 | 1.57 |
| 2S I3D scratch (TVL1) | 54.41 | 56.42 | 60.83 | 54.45 | 61.30 | 57.48 | 3.01 |
| **2S I3D pretrained (TVL1) w/o DA** | 70.48 | 67.57 | 68.61 | **66.43** | **72.00** | 69.02 | 1.99 |
| **2S I3D pretrained (Farnebäck) w DA** | **71.00** | **69.10** | **71.88** | 65.58 | 71.47 | **69.81** | 2.31 |

TABLE 3 – Comparaison entre les ensembles de modèles ayant des architectures homogènes. Quelques explications : **w/o DA** sans augmentation de données ; **w DA** avec augmentation de données.

| **Échantillon de test** | **0** | **1** | **2** | **3** | **4** | $\mu$ | $\sigma$ |
|---|---|---|---|---|---|---|---|
| (1) C3D pretrained | 61.13 | 60.59 | 61.00 | 58.13 | 61.56 | 60.48 | **1.21** |
| (2) I3D scratch | 54.93 | 55.91 | 58.53 | 53.85 | 58.86 | 56.42 | 1.97 |
| (3) 2S I3D pretrained (TVL1) w/o DA | 70.48 | 67.57 | 68.61 | **66.43** | 72.00 | 69.02 | 1.99 |
| (4) 2S I3D pretrained (Farnebäck) w DA | 71.00 | 69.10 | 71.88 | 65.58 | 71.47 | 69.81 | 2.31 |
| **Ensemble global (1) + (2) + (3) + (4)** | **72.05** | **70.04** | **73.20** | 66.35 | **74.95** | **71.32** | 2.95 |

TABLE 4 – Comparaison entre la meilleure combinaison donnant lieu à un ensemble global et les modèles ensemblistes le constituant

# Références


[1] C. Rotily and J. Kritter, "Optimisation technique des moyens de vidéoprotection," *Préventique*, no. 166, pp. 63–64, 2019.

[2] X. Latour and B. Pauvert, *Libertés publiques et droits fondamentaux*. Studyrama, 2018.

[3] O. Fillieule, P. Viot, and G. Descloux, "Vers un modèle européen de gestion policière des foules protestataires ?," *Revue française de science politique*, vol. 66, no. 2, pp. 295–310, 2016.

[4] J. Kritter, M. Brévilliers, J. Lepagnot, and L. Idoumghar, "On the real-world applicability of state-of-the-art algorithms for the optimal camera placement problem," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1103–1108, IEEE, 2019.

[5] S. Ghambari, L. Idoumghar, L. Jourdan, and J. Lepagnot, "A* based differential evolution algorithm for uav path planning problem in urban environment," in *Evolution Artificielle (EA)*, 2019.

[6] M. Thida, Y. L. Yong, P. Climent-Pérez, H.-l. Eng, and P. Remagnino, "A literature review on video analytics of crowded scenes," in *Intelligent multimedia surveillance*, pp. 17–36, Springer, 2013.

[7] J. Carreira and A. Zisserman, "Quo vadis, action recognition ? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

[8] C. Dupont, L. Tobias, and B. Luvison, "Crowd-11 : A dataset for fine grained crowd behaviour analysis," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 2017-July, (Honolulu, United States), pp. 2184–2191, 2017.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[10] M. Bendali-Braham, J. Weber, G. Forestier, L. Idoumghar, and P.-A. Muller, "Transfer learning for the classification of video-recorded crowd movements," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 271–276, IEEE, 2019.

[11] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep neural network ensembles for time series classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, 2019.

[12] Z.-H. Zhou, "Ensemble learning.," *Encyclopedia of biometrics*, vol. 1, pp. 270–273, 2009.

[13] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*, pp. 23–37, Springer, 1995.

[14] B. Efron, "Bootstrap methods : another look at the jackknife," in *Breakthroughs in statistics*, pp. 569–593, Springer, 1992.

[15] O. Sagi and L. Rokach, "Ensemble learning : A survey," *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[16] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[17] W. Liu, M. Zhang, Z. Luo, and Y. Cai, "An ensemble deep learning method for vehicle type classification

on visual traffic surveillance sensors," *IEEE Access*, vol. 5, pp. 24417–24425, 2017.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet : A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

[20] S. Pouyanfar and S.-C. Chen, "Automatic video event detection for imbalance data using enhanced ensemble deep learning," *International Journal of Semantic Computing*, vol. 11, no. 01, pp. 85–109, 2017.

[21] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pp. 321–330, 2006.

[22] S. Pouyanfar and S.-C. Chen, "Semantic concept detection using weighted discretization multiple correspondence analysis for disaster information management," in *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, pp. 556–564, IEEE, 2016.

[23] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe : Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, 2014.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[29] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.

[30] G. Chen, M. Giuliani, D. Clarke, A. Gaschler, and A. Knoll, "Action recognition using ensemble weighted multi-instance learning," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4520–4525, IEEE, 2014.

[31] E. Walach and L. Wolf, "Learning to count with cnn boosting," in *European conference on computer vision*, pp. 660–676, Springer, 2016.

[32] C. Wu, T. Yin, S. Ge, and K. Yu, "Ensemble learning for crowd flows prediction on campus," in *International Conference on Smart Computing and Communication*, pp. 103–113, Springer, 2017.

[33] S. Gong, T. Xiang, and S. Hongeng, "Learning human pose in crowd," in *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis*, pp. 47–52, 2010.

[34] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[35] C.-N. J. Yu and T. Joachims, "Learning structural svms with latent variables," in *Proceedings of the 26th annual international conference on machine learning*, pp. 1169–1176, 2009.

[36] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks : many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[37] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3d residual networks for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 3154–3160, 2017.

[38] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint pattern recognition symposium*, pp. 214–223, Springer, 2007.

[39] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *ArXiv*, 2017.

[40] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[41] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, 2015.