Des pixels aux segments pour la classification de séries temporelles d'images via des réseaux de neurones convolutionnels

Mohamed Chelali, Camille Kurtz, Anne Puissant, Nicole Vincent

Des pixels aux segments pour la classification de séries temporelles d'images via des réseaux de neurones convolutionnels

Mohamed Chelali[1] Camille Kurtz[1] Anne Puissant[2] Nicole Vincent[1]

[1] Université de Paris, LIPADE (EA 2517), Paris, France
[2] Université de Strasbourg, LIVE (UMR 7362), Strasbourg, France

{prénom.nom}@{u-paris, unistra}.fr

Résumé

Les séries temporelles d'images, telles que les séries temporelles d'images satellites (STIS) ou les séquences fonctionnelles d'IRM dans le domaine médical, fournissent des informations spatiales et temporelles importantes sur une scène observée. Dans de nombreuses applications telles que la classification d'images, la prise en compte de ces informations peut être cruciale et discriminatoire lors de la prise de décision. Cependant, l'extraction de caractéristiques spatio-temporelles à partir de séries temporelles d'images est difficile en raison de la complexité de la structure du cube de données. Dans cet article, nous présentons une stratégie basée sur les marches aléatoires (Random Walk) pour construire une nouvelle représentation des données reposant sur des segments, passant d'une dimension $2D + t$ à une dimension $2D$, plus facilement manipulable et conservant une information spatiale partielle. Cette nouvelle représentation est ensuite utilisée pour définir des caractéristiques spatio-temporelles dans un réseau de neurones convolutionnel (CNN) classique lors d'un apprentissage bout en bout ayant pour objectif de classer des séries temporelles d'images. L'intérêt de cette approche est mis en évidence sur une application de télédétection pour la classification et la cartographie de parcelles agricoles.

Mots Clef

Séries Temporelles d'Images Satellites (STIS), caractéristiques spatio-temporelles, marche aléatoire, réseaux de neurones convolutionels.

Abstract

Image time series, such as Satellite Image Time Series (SITS) or MRI functional sequences in the medical domain, carry both spatial and temporal information of the sensed scene. In many applications such as image classification, taking into account such rich information may be crucial and discrimative during the decision making stage. However, the extraction of spatio-temporal features from image time series is difficult due to the complex representation of the data cube. In this article, we present a strategy based on Random Walk to build a novel segment-based representation of the data, passing from a $2D + t$ dimension to a $2D$ one, more easily usable and without loosing too much spatial information. Such new representation is then used to feed a classical Convolutional Neural Network (CNN) in order to learn spatio-temporal features with only $2D$ convolutions and to classify image time series data for a particular classification problem. The interest of this approach is highlighted on a remote sensing application for the classification and the mapping of complex agricultural crops.

Keywords

Satellite Image Time Series, spatio-temporal features, Random Walk, Convolutional Neural Networks.

1 Introduction

Une série temporelle d'images est un ensemble ordonné d'images de la même scène acquises à différentes dates. Ces données fournissent des informations riches sur l'évolution temporelle de la zone étudiée. Dans les applications de télédétection, de nombreuses constellations de satellites acquièrent des images avec une haute résolution spatiale, spectrale et temporelle à travers le monde, conduisant à des séries temporelles d'images satellites (STIS). Par exemple, les capteurs Sentinel-2 produisent des STIS optiques avec un temps de re-visite de 5 jours et une résolution spatiale de 10 à 20 mètres.

Les STIS aident à comprendre l'évolution de l'environnement, à étudier les causes de divers changements et à prévoir l'évolution future. Les informations temporelles, intégrées aux dimensions spectrales et spatiales, permettent notamment l'analyse de motifs complexes liés à la cartographie de la couverture du sols (e.g. zones agricoles, zones urbaines) ou l'identification des changements d'utilisation des sols (e.g. urbanisation, déforestation) et même à la production de cartes précises de la couverture du sols d'un territoire [1].

Un problème majeur lors de l'analyse des séries temporelles d'images est de considérer simultanément le domaine temporel et le domaine spatial du cube de données $2D+t$. Dans ce contexte, les méthodes d'analyse des STIS

sont principalement basées sur des informations temporelles [2] attachées aux pixels. Dans certaines applications, cela peut cependant ne pas être suffisant pour obtenir des résultats satisfaisants. La prise en compte à la fois de l'aspect temporel et spatial peut, par exemple, faciliter la discrimination entre différentes classes complexes de couverture du sol (e.g. les pratiques agricoles, les zones urbaines par rapport aux zones péri-urbaines). Notre objectif est ici de discriminer des classes complexes de couverture du sol qui sont sujettes aux confusions lorsqu'une seule image (une seule date) est utilisée.

Cet article se focalise sur le problème de l'extraction de caractéristiques spatio-temporelles pour la classification de séries temporelles d'images, en utilisant l'apprentissage profond. Dans ce contexte, nous définissons une nouvelle représentation spatio-temporelle des séries temporelles d'images qui permet l'utilisation des réseaux de neurones convolutionnels (CNN) classiques (proposés pour l'analyse d'images $2D$). Notre principale contribution est la proposition d'une transformation des données $2D + t$ sous forme d'une image $2D$ sans perdre trop d'information spatiale. Elle s'appuie sur la construction d'ensembles de segments ($1D$) en utilisant le paradigme des marches aléatoires (*Random Walk*) pour diminuer la dimension spatiale des données. Cette nouvelle représentation des données est ensuite utilisée pour alimenter un CNN afin : (1) d'apprendre des caractéristiques spatio-temporelles avec des filtres $2D$ impliquant à la fois des informations temporelles et spatiales, et (2) de classer les séries temporelles d'images quant à un problème thématique particulier.

La suite de cet article est organisée comme suit. En Section 2, nous présentons un état de l'art sur les méthodes d'analyse des STIS. Notre proposition de représentation planaire des séries temporelles d'images pour l'analyse avec les CNN est introduite dans la Section 3. Les validations expérimentales sont présentées dans la Section 4. Enfin, la Section 6 propose quelques conclusions et perspectives de recherche.

2 Approches existantes pour l'analyse de STIS

Les STIS constituent une source de données importantes permettant l'observation de la surface de la Terre. Ces données améliorent notamment notre connaissance et notre compréhension de l'évolution et des changements environnementaux, qui peuvent être de différents types, origines et durées. Pour une étude détaillée, voir [3].

Les méthodes pionnières traitent des STIS « image par image » en calculant différentes caractéristiques par pixel puis en les utilisant dans des procédures classiques reposant sur un apprentissage automatique. Les méthodes conçues pour l'analyse bi-temporelle localisent et étudient les changements apparus entre deux observations. Ces méthodes incluent la différence d'images [4], le rapport [5] ou l'analyse vectorielle de changement [6].

D'autres familles de méthodes sont plus directement dédiées à l'analyse des séries temporelles d'images. La plupart d'entre elles sont basées sur une classification à plusieurs dates. Parmi elles, nous trouvons l'analyse de trajectoires radiométriques [7]. Ces méthodes exploitent l'évolution de l'aspect du sol à travers le temps (e.g. saison, évolution de la végétation [8]), et prennent en compte l'ordre des dates en utilisant des méthodes d'analyse de séries temporelles [9]. Chaque pixel est associé à une série de mesures ordonnées dans le temps (et alignées), et les changements des mesures dans le temps sont analysés pour trouver des motifs (temporels), en utilisant des approches statistiques ou symboliques.

Certaines méthodes proposent d'abord de représenter les STIS dans un nouvel espace. Nous pouvons citer les approches dans le « domaine fréquentiel » qui incluent une analyse spectrale ou une analyse en ondelettes [10]. D'autres méthodes extraient des caractéristiques « handcrafted », plus discriminantes, dans un nouvel espace enrichi [11–13]. Concernant l'étape de classification, les approches classiques mesurent la similarité entre tout échantillon entrant (qui peut être enrichi avec les caractéristiques expertes) et l'ensemble d'apprentissage. Elles attribuent à la série temporelle le label de la classe la plus similaire en utilisant par exemple une distance euclidienne et un algorithme des plus proche voisins ou / et la méthode *Dynamic Time Warping* [14].

Plus récemment, plusieurs méthodes d'apprentissage profond ont été envisagées pour classer les images de télédétection et générer des cartes d'occupation du sol. Généralement, des réseaux de neurones convolutifs (CNN) sont utilisés pour traiter le domaine spatial des données en appliquant des convolutions $2D$ [15]. Pour les séries temporelles d'images, les convolutions peuvent être appliquées dans le domaine temporel [2]. Par ailleurs, les réseaux de neurones récurrents (RNN), comme les *Long short-term memory* (LSTM), sont bien adaptés pour les données temporelles et ont été utilisés avec succès dans [16, 17]. Dans ce contexte, les approches d'apprentissage profond surpassent les algorithmes de classification traditionnels tels que *Random Forest* [18], mais elles ne prennent pas directement en compte la dimension spatiale des données car elles considèrent les pixels des images de manière indépendante. Certaines approches ont été proposées pour considérer les dimensions temporelles et spatiales du cube de données $2D + t$ [19]. Une stratégie commune consiste à entraîner deux modèles, l'un pour la dimension spatiale et l'autre pour la dimension temporelle, puis à fusionner leurs résultats au niveau décisionnel. En analyse de vidéo, les caractéristiques spatio-temporelles sont apprises directement en utilisant des réseaux de neurones convolutionnels $3D$ [20] mais une telle stratégie nécessite l'apprentissage d'un grand nombre de paramètres.

Dans cet article, la stratégie proposée consiste à classer les STIS à l'aide d'un CNN classique $2D$, grâce à une nouvelle représentation de la série temporelle d'images intégrant simultanément des informations temporelles et spatiales du

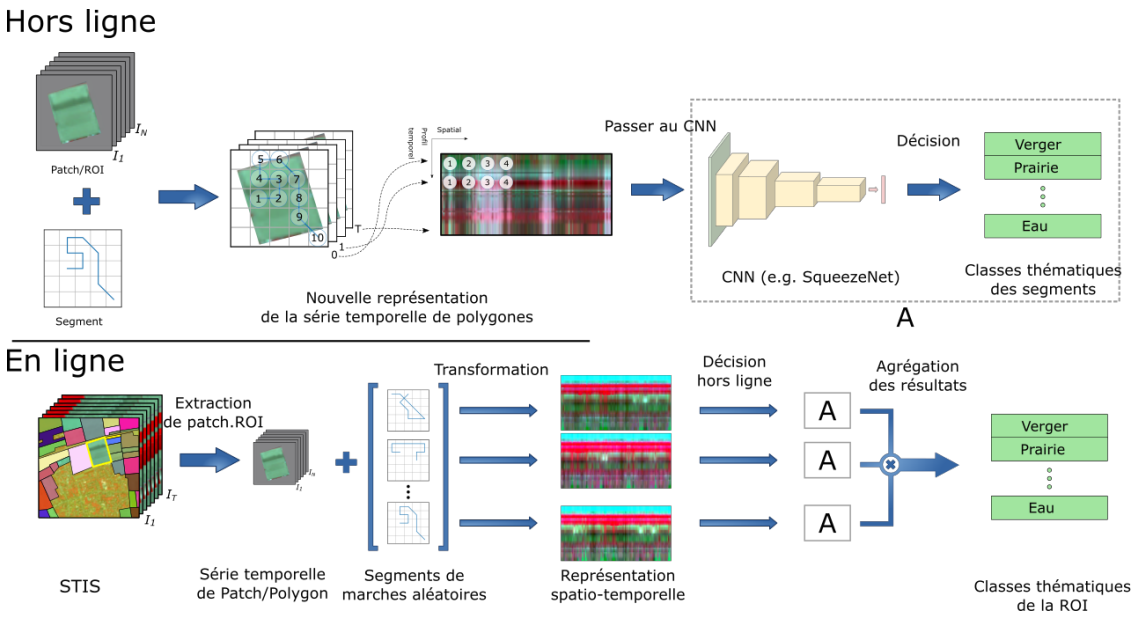


FIGURE 1 – Organigramme de la méthode proposée pour la classification des séries temporelles d'images basée sur une représentation planaire spatio-temporelle obtenue par les segments des marches aléatoires (*Random Walk*). (en haut) Étape hors-ligne (e.g. apprentissage) et (en bas) étape en-ligne (e.g. test) liées au processus de classification.

cube de données. Nous comparons notre approche avec une approche de l'état de l'art basée sur l'utilisation de convolutions $1D$ appliquées dans le domaine temporel [2] pour classer les séries temporelles de pixels.

3 Méthode proposée

La méthode proposée vise à classer les séries temporelles d'images à partir de caractéristiques spatio-temporelles. La stratégie sous-jacente est d'utiliser une architecture de réseau de neurones profond classique d'entrée $2D$ afin d'apprendre un modèle spatio-temporel à partir de données $2D + t$. La Figure 1 illustre l'organigramme de notre système, avec les phases traditionnelles hors ligne (e.g. apprentissage) et en ligne (e.g. test) d'un processus de classification. Notre système étant dédié à la classification d'objets d'intérêt présents dans des images (e.g. les parcelles agricoles), les données d'entrée initiales peuvent être une image centrée sur un objet spécifique, un patch d'image ou uniquement les pixels connectés d'une région d'intérêt (ROI), comme une parcelle représentée sous la forme d'un polygone. Dans tous les cas, nous utiliserons le terme « image » pour se référer aux données d'entrée.

Nous construisons des éléments $2D$ dans lesquels nous intégrons certaines propriétés spatiales et nous les utilisons aussi bien dans la phase d'apprentissage (dans le processus hors ligne) que dans la phase de classification d'une nouvelle STIS. Nous différons ainsi des autres approches de l'état de l'art considérant une structure $1D$ [2] ou $3D$ [20]. Nous commençons par transformer les données originales $2D+t$ en une entité planaire contenant des données spatio-temporelles acquises au cours du temps, construites à partir de segments spatiaux $1D$. Une telle représentation est ensuite considérée comme entrée d'un CNN pour obtenir une classification des segments qui sont construits hors ligne et utilisés en ligne. Le réseau est entraîné afin d'apprendre les étiquettes des segments à partir des informations spatiales et temporelles contenues dans les données.

3.1 Transformation des données

Nous expliquons d'abord comment transformer les données originales $2D + t$ en représentations $2D$ moins complexes, qui contiennent des données spatio-temporelles construites à partir de segments spatiaux $1D$.

Des pixels aux segments Les méthodes traditionnelles qui ne gèrent que des informations temporelles considèrent le domaine $2D$ comme un ensemble (ou sac) de pixels, c'est-à-dire des entités $0D$. Les pixels sont généralement caractérisés par leurs séries d'intensités colorimétriques à travers le temps. Dans notre cas, nous voulons inclure de l'information spatiale, d'où l'introduction de la notion de segments qui sont des entités spatiales $1D$. Un pixel est alors caractérisé par une portion de courbe contenue dans l'image et ayant comme origine ce pixel. Dans une portion de courbe $1D$, chaque pixel a 2 voisins, à l'exception des deux pixels extrêmes. Notre transformation diminuera alors l'information spatiale en ne gardant que 2 voisins. Si L est la longueur du segment inclus dans l'image d'entrée, et N est le nombre d'images de la série temporelle, nous construisons alors une image 2D de taille $N \times L$ comme illustré dans la partie supérieure gauche de la Figure 1.

Différentes stratégies pour définir les segments $1D$ dans l'espace $2D$ d'origine sont ici étudiées et comparées. Pour chaque stratégie, nous appliquons le processus N_p fois à partir d'une donnée d'entrée, produisant N_p segments dif-

férents, afin de garder suffisamment de voisins. Sur chaque portion de courbe, les pixels sont naturellement ordonnés. De cette façon, la complexité de la représentation spatiale des images est réduite, passant de l'image $2D$ à des portions de courbes $1D$. Nous montrons dans la suite comment les portions de courbes caractérisées par des informations temporelles conduisent à des données spatio-temporelles que nous représentons par des images $2D$.

Des segments aux représentations $2D$ Pour une série donnée composée de N images (e.g. N acquisitions temporelles), les segments sont d'abord extraits. Ils sont utilisés pour l'apprentissage d'un modèle de classification des portions de courbe associées à un pixel. Pour chaque portion de courbe nous construisons une image de taille $N \times L$. Celle-ci contient sur chaque ligne les intensités des pixels le long du chemin et en ordonnée figure l'évolution de ces intensités au cours du temps. Ces segments spatiaux $1D$ sont ainsi désormais enrichis d'informations temporelles pour construire des données spatio-temporelles $2D$.
Cela conduit à une nouvelle représentation $2D$ composée de N lignes (N images dans le STIS) dans le domaine temporel et de L colonnes (L longueur du segment considéré) dans le domaine spatial. Cette « image » peut alors être interprétée comme une représentation spatio-temporelle $2D$ d'une partie de la série temporelle de l'image $2D + t$, associée à un pixel particulier.
En appliquant le processus de transformation aux N_p portions de courbes, nous obtenons N_p représentations spatio-temporelles à partir de la série temporelle de l'image d'origine. Ces représentations seront utilisées comme entrée d'un processus d'apprentissage, les classes associées aux segments sont les classes de l'image d'entrée annotée à laquelle ils appartiennent.

3.2 Stratégies de construction de segments

Deux stratégies différentes ont été envisagées pour créer des segments dans une image :

- **Stratégie *linéaire* (notée scan).** Ici, nous considérons toutes les lignes et les colonnes de l'image d'entrée pour construire des segments $1D$. Les dimensions de l'image d'entrée limitent à la fois le nombre et la longueur des segments possibles. Pour garantir des longueurs L similaires pour chaque segment, il est nécessaire de répliquer les valeurs sur des segments trop courts (cela peut correspondre à des segments extraits des bordures de l'image). De cette façon, chaque pixel de l'image n'est considéré que deux fois dans la nouvelle représentation et, pour chaque pixel, seuls les voisins 4-connexes sont pris en compte.
- **Stratégie basée sur les *marches aléatoires* (notée RW).** Une marche aléatoire [21] est un processus mathématique basé sur un système itératif aléatoire. Chaque itération est une étape avec des propriétés Markoviennes. Ici, les marches aléatoires sont utilisées pour générer un chemin aléatoire dans le plan $2D$ de l'images, un tel chemin ayant une longueur L est noté $RW(L)$. Le premier point du segment est choisi aléatoirement sur l'image $2D$ et pour le point suivant, 8 directions sont possibles. Étant donné une image d'entrée, nous procédons aux initialisations des N_p chemins aléatoires. Pour chacun, une image $2D$ est alors construite, où les lignes correspondent aux valeurs des pixels du chemin extraites dans les différentes images de la série. La chronologie est liée au numéro de ligne. Au centre de la partie en-ligne de la Figure 1 sont illustrées les représentations spatio-temporelles de trois segments différents construits à partir d'une série temporelle d'images.

3.3 Modèle CNN (architecture)

Les réseaux neuronaux convolutifs (CNN) font référence à une famille d'algorithmes d'apprentissage profond. Les systèmes sont composés de deux parties. La première est conçue pour l'extraction de caractéristiques, elle possède des couches de neurones qui calculent les convolutions des précédentes. Les neurones de chaque couche sont activés par des fonctions non linéaires (e.g. sigmoïde, ReLU) afin de conserver les caractéristiques les plus représentatives. On retrouve également des couches de *max-pooling* entre les couches convolutives pour réduire progressivement la quantité des entrées et le nombre de paramètres à calculer pour définir le réseau, et donc aussi contrôler le sur-apprentissage. La deuxième partie peut être un classifieur. Généralement, c'est une couche entièrement connectée qui fournit un vecteur de probabilité, avec une fonction *softmax* pour prédire l'étiquette de classe des données d'entrée.
Nous avons choisi le modèle SqueezeNet [22] mais tout autre CNN $2D$ peut être utilisé. SqueezeNet a des propriétés intéressantes, peu de paramètres et le même niveau de précision que le modèle AlexNet sur l'ensemble des images de la base ImageNet. L'entraînement du modèle peut alors être plus rapide si l'on peut utiliser un processus de transfert. Le modèle SqueezeNet introduit un nouveau module appelé *Fire* composé d'une couche de compression utilisant des filtres de convolution 1×1 suivie d'une couche d'expansion qui contient un mélange de filtres de convolution 1×1 et 3×3. Nous avons utilisé l'implémentation PyTorch de SqueezeNet [1]. Le modèle CNN est entraîné avec les représentations spatio-temporelles $2D$ obtenues à partir de chaque courbe extraite des séries temporelles d'images d'entrée de l'ensemble d'apprentissage.

3.4 Prise de décision au niveau du polygone

Comme déjà mentionné, nos données d'entrée sont des polygones représentant des objets d'intérêt dans les STIS. À chaque donnée d'entrée, nous associons un ensemble de N_p chemins, N_p est donc un paramètre de la méthode. À chaque chemin est associée une représentation spatio-

1. https://github.com/pytorch/vision/blob/master/torchvision/models/squeezenet.py

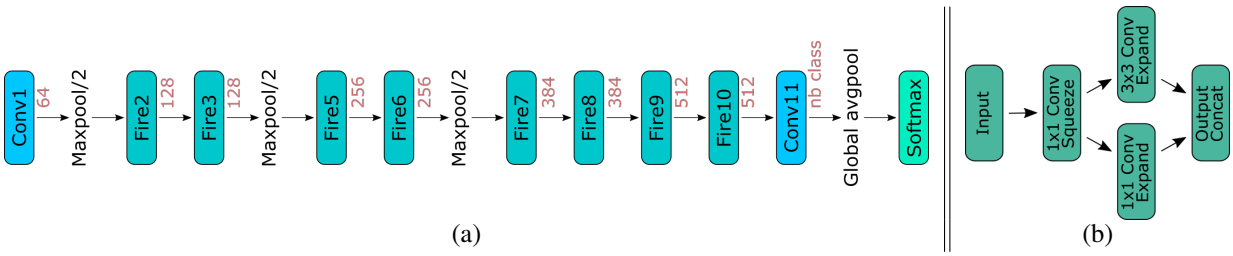


FIGURE 2 – Architecture du CNN. (a) Le modèle SqueezeNet [22] ; (b) Couche *Fire*.

temporelle planaire $2D$. Grâce au classifieur décrit dans la Section 3.3, une étiquette de classe est prédite pour chaque représentation spatio-temporelle $2D$ (e.g. pour chaque chemin) avec une certaine probabilité. Nous procédons à la prise de décision en considérant la moyenne des probabilités retournées par le modèle pour les N_p segments du polygone et nous affectons l'étiquette de classe avec la probabilité la plus élevée assurant une décision unique par image.

4 Étude expérimentale

L'étude expérimentale dans le domaine de la télédétection repose sur la classification de parcelles agricoles à partir d'une STIS. Le but est de discriminer des classes thématiques agricoles complexes (e.g. vergers traditionnels vs. vergers intensifs). L'identification automatique de ces classes est une tâche difficile car les vergers font l'objet de nombreuses pratiques agricoles en fonction de la saison et de la politique de gestion du territoire. Afin de différencier ces deux classes, les caractéristiques spatio-temporelles portent des informations utiles pour discriminer les pratiques agricoles.

4.1 Données

Nous disposons d'une STIS acquise par le satellite Sentinel-2. Elle contient $N = 50$ images optiques captées en 2017 sur la même zone géographique en Alsace (tuile 32ULU). La Figure 3 montre la distribution temporelle des images de cette STIS. Les images ont été corrigées et orthorectifiées par le programme français Theia [2] pour être radiométriquement comparables. Nous disposons également des masques de nuages, d'ombres et de saturation associés à chaque image. Une étape de prétraitement a été appliquée sur les images avec une interpolation linéaire sur les pixels masqués pour combler les valeurs manquantes dans la STIS.

Pour chaque image, seules trois bandes sont conservées : le proche infrarouge (Nir), le rouge (R) et le vert (G). La bande bleue (B) est considérée comme inutile dans la littérature pour discriminer différents types de cultures agricoles car plus sensible aux effets atmosphériques. Toutes ces bandes ont une résolution spatiale de 10 mètres.

Les données de référence utilisées sont extraites du RPG [3], qui contient la délimitation des parcelles agricoles (dans notre contexte les vergers). Quelques exemples de polygones sont représentés dans la Figure 1. Ces polygones ont été corrigés manuellement par photo-interprétation pour assurer une bonne délimitation des parcelles. Les données de référence utilisées dans notre expérience sont les étiquettes sémantiques de ces polygones (vergers traditionnels ou intensifs). Ces polygones conduisent chacun à une nouvelle série temporelle de polygones, notée *Polygon Image Time Series* (PITS).

2. https://theia.cnes.fr/
3. http://professionnels.ign.fr/rpg

FIGURE 3 – Distribution des images de la STIS (2017).

TABLE 1 – Résumé des données. (première col.) Nombre initial de polygones par classe ; (deux dernières col.) Nombre de segments spatio-temporels en fonction de la stratégie de construction des segments.

Classes	# polygones	# Rep. spatio-temp. pour *scan*	# Rep. spatio-temp. pour *RW*
Vergers int.	100	3084	3000
Vergers trad.	100	3059	3000
Total	200	6143	6000

4.2 Préparation des données

Partant des données initiales, des PITS sont formés, puis N_p segments sont extraits de la ROI. En analysant au préalable les tailles des ROI, nous avons fixé $N_p = 30$ pour la stratégie RW. Pour la stratégie *scan*, le nombre de segments $1D$ possibles dépend de la taille de la ROI. Dans la partie hors ligne de la Figure 1, nous illustrons le processus de transformation de PITS avec $RW(10)$. Le Tableau 1 présente le nombre d'instances de polygones par classe et le nombre de segments construits à partir de ces données selon la stratégie choisie de construction des segments.

Par la suite, nous étudions l'impact de la longueur L des segments. Cela permet d'évaluer l'impact de l'ajout d'informations spatiales pour apprendre des caractéristiques spatio-temporelles au lieu de ne considérer des pixels seulement (des entités $0D$), comme c'est le cas dans la plupart des approches de l'état de l'art. Les longueurs L utilisées sont de 10, 50, 100 et 224. Le plus grand dépend de la taille d'entrée maximale du modèle CNN *SqueezeNet*

utilisé. Pour la stratégie *scan*, la longueur des segments est limitée a 10 car cela dépend des tailles des ROI. Lors de la construction de l'image $2D$ de taille 224×224 à partir des segments, si les segments sont inférieurs à 224, nous les centrons horizontalement et le reste des colonnes est fixé à une valeur nulle. Le Tableau 1 indique le nombre réel de segments.

Pour la dimension temporelle (axe vertical), nous proposons deux stratégies. La première consiste à centrer verticalement les informations d'origine des N images d'entrée ($N = 50$). Les lignes supérieures et inférieures restantes sont fixées à une valeur nulle. La seconde stratégie consiste à compléter la STIS de manière à modéliser 224 acquisitions d'images (potentiellement synthétiques). Elles sont définies en appliquant une interpolation linéaire temporelle à partir de la STIS initiale. Nous supposons que l'évolution temporelle entre deux dates consécutives est monotone et linéaire. L'interpolation se fait alors en considérant que nous n'avons que 224 acquisitions dans l'année donc une acquisition toutes les 39 heures. Pour les dates précédant la première image de la série initiale (début de l'année 2017), nous affectons les informations temporelles de la première date dans la STIS, cela correspond à l'hiver où la végétation est au repos. Pour les dernières dates (fin de l'année des 2017), nous affectons les dernières informations temporelles connues dans la STIS initiale.

La normalisation des données est une transformation linéaire basée sur les valeurs maximale et minimale de l'ensemble de données après que les valeurs soient limitées par les 2% (ou 98 %) percentiles, comme proposé dans [2].

4.3 Protocole d'apprentissage / validation

Les expériences sont validées en utilisant une validation croisée à cinq sous-ensembles (5 *folds*). Pour chaque sous-ensemble, nous divisons l'ensemble de données en trois sous-ensembles, au niveau polygone, avec des tailles de 60%, 20% et 20% représentant respectivement les ensembles d'apprentissage, de validation et de test. Le modèle CNN est ensuite appris et évalué cinq fois. Au final, nous rapportons le taux de reconnaissance globale (TR) des cinq sous-ensembles et nous indiquons l'écart-type (STD).

Le modèle est appris à l'aide de l'optimiseur *Adam* avec un taux d'apprentissage de 10^{-6} et les valeurs par défaut des autres paramètres ($\beta_1 = 0.9$, $\beta_2 = 0.999$ et $\epsilon = 10^{-8}$) avec une taille de lot de 8. Nous limitons le nombre d'époques à 2000, suivant une technique d'arrêt précoce avec un nombre de patience de 100. Les expériences sont effectuées sur une machine avec une garte graphique Nvidia GPU GTX 1050 Ti (4 Go).

Si le nombre de parcelles agricoles est limité, le nombre de chemins construits n'a pas le même ordre de grandeur et permet de réaliser un apprentissage raisonnable. Néanmoins, nous avons considéré deux stratégies pour l'apprentissage de la labellisation des chemins. Dans la première stratégie (*from scratch*), le modèle est entraîné à partir d'une initialisation aléatoire des poids du réseau et dans la seconde (*fine-tuning*) le réseau est initialisé avec les poids obtenus sur IMAGENET, puis affiné avec nos données.

5 Résultats et discussions

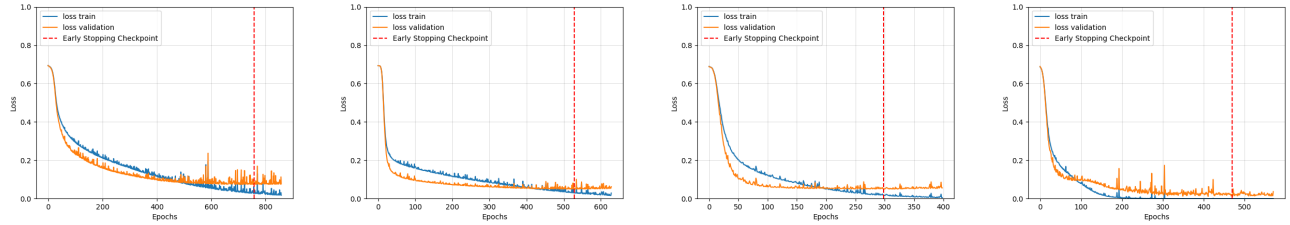
Les représentations spatio-temporelles $2D$ proposées sont utilisées pour apprendre les poids de l'architecture du CNN choisi. Pour la stratégie *scan*, nous utilisons simplement la longueur de segment $L = 10$ car nous sommes limités par les dimensions des ROI. Au niveau du segment, la Figure 4 illustre les courbes de perte obtenues lorsque le modèle est entraîné à partir d'une initialisation aléatoire (*from scratch*) avec les différentes longueurs des segments RW, respectivement 10, 50, 100 et 224. On observe que l'entraînement se fait dans de meilleures conditions avec les différentes longueurs de marches aléatoires. Dans la courbe de perte de *RW(10)*, on observe de fortes oscillations ce qui n'est pas le cas dans d'autres. Cela est potentiellement dû au manque d'informations dans les images fournies au CNN (beaucoup de valeurs nulles dans l'image d'entrée), et chaque fois lors de l'augmentation de la longueur L, la perte en validation (courbe orange) diminue conduisant à de meilleurs taux d'apprentissage.

Les résultats de la classification (précision globale) avec la STIS originale sont rapportés dans le Tableau 2. Tous les scores sont dans la même plage sauf le *scan(10)*. On peut penser qu'avec les différentes longueurs de chemins aléatoires, nous avons pu conserver assez d'informations spatiales permettant de discriminer les deux classes considérées (vergers traditionnels et intensifs). Nous remarquons qu'avec le *fine-tuning*, tous les scores sont augmentés, avec *RW(224)* en première position.

Les résultats obtenus sont comparés à ceux obtenus avec la méthode TempCNN [2] [4]. TempCNN est dédié à la classification des séries temporelles, où les convolutions sont appliquées dans le domaine temporel (convolutions $1D$). Les tailles de filtres sont fixées selon le critère donné dans [2] : avec une taille de noyau de 5 si l'on considère les dates originales, et de 11 si l'on considère les dates interpolées. À des fins de comparaison, nous avons entraîné et validé le modèle TempCNN en utilisant les mêmes données et le même protocole de validation que celui utilisé pour notre modèle. Ce modèle TempCNN est proposé dans [2] avec différentes architectures (profondeurs du réseau), conduisant à différents nombres de filtres que nous avons testés.

Le Tableau 3 rapporte les résultats obtenus avec la méthode TempCNN. Les meilleurs scores ont été obtenus avec 256 filtres. Le meilleur score obtenu avec notre méthode (lorsque nous entraînons le réseau *from scratch*) est légèrement meilleur que ceux obtenus avec TempCNN. Cependant, lorsque nous utilisons *fine-tuning*, nous les surpassons. Cela met en évidence, pour notre contexte applicatif, l'avantage de considérer un modèle classique de CNN $2D$ pour classer des images $2D + t$ combinées avec nos représentations spatio-temporelles, en particulier dans un cas où les données sont peu nombreuses.

4. https://github.com/charlotte-pel/temporalCNN



(a) (b) (c) (d)

FIGURE 4 – Courbes de *loss* de la phase d'entraînement, en fonction de la longueur L des chemins des marches aléatoires : (a) $L = 10$; (b) $L = 50$; (c) $L = 100$ and (d) $L = 224$.

TABLE 2 – Résultats de classification (taux de reconnaissance global – TR et écart-type – STD) obtenus avec la représentation spatio-temporelle (avec l'information temporelle originale).

	From scratch		Fine-tuning	
Longueurs des segments	**TR**	**STD**	**TR**	**STD**
scan(10)	73.00	9.27	80.50	5.09
RW(10)	85.50	5.56	90.50	5.78
RW(50)	80.00	7.27	92.00	2.91
RW(100)	84.50	5.33	**94.00**	**2.00**
RW(224)	**86.00**	**5.61**	92.50	4.18

TABLE 3 – Résultats de classification (taux de reconnaissance global – TR et écart-type – STD) avec les différentes architectures de TempCNN [2] (avec l'information temporelle originale et un noyau de taille 5)

Nb de filtres	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
TR	78.81	77.38	81.66	78.45	**85.37**	81.73	84.80
STD	6.08	6.51	4.59	4.79	**3.44**	5.75	6.48

TABLE 4 – Résultats de classification (taux de reconnaissance global – TR et écart-type – STD) obtenus avec la représentation spatio-temporelle (avec l'information temporelle interpolée).

	From scratch		Fine-tuning	
Longueurs des segments	**TR**	**STD**	**TR**	**STD**
scan(10)	78.50	6.44	83.00	1.87
RW(10)	90.00	4.18	**93.00**	**4.30**
RW(50)	90.50	1.87	**93.00**	**2.44**
RW(100)	**93.50**	**2.00**	**93.00**	**2.44**
RW(224)	91.50	1.22	91.00	2.00

TABLE 5 – Résultats de classification (taux de reconnaissance global – TR et écart-type – STD) avec les différentes architectures de TempCNN [2] (avec l'information temporelle interpolée et un noyau de taille de 11)

Nb de filtres	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
TR	78.96	81.40	83.96	81.86	85.93	84.23	**87.21**
STD	7.34	6.32	7.14	5.18	8.03	6.23	**8.28**

Le Tableau 4 présente quant à lui les résultats de la classification lorsque l'on considère les images spatio-temporelles avec la stratégie d'interpolation temporelle. Nous remarquons que tous les scores sont augmentés par rapport à ceux avec moins d'informations temporelles (Tableau 2). Cela s'explique par la distribution temporelle non régulière des images originales. Ainsi, avec l'interpolation linéaire, nous rendons la distribution temporelle régulière pour obtenir 224 dates. *scan(10)* est toujours moins efficace que les stratégies qui sont basées sur *RW*. Tous les scores obtenus sont du même ordre avec *RW(100)* en première position que ce soit avec ou sans *fine-tuning*.

Le Tableau 5 rapporte les scores lors de la classification avec TempCNN [2] basé sur des données seulement temporelles. La précision globale est légèrement augmentée par rapport à la précédente (Tableau 3) et le meilleur résultat est obtenu avec 1024 filtres. Les scores obtenus avec notre méthode sont plus élevés (avec et sans *fine-tuning*) qu'avec TempCNN ce qui confirme l'intérêt d'une approche prenant en compte des caractéristiques spatio-temporelles.

6 Conclusion

Dans cet article, nous présentons une nouvelle méthode de classification de séries temporelles d'images basée sur une représentation spatio-temporelle. Cette dernière vise à réduire la structure de données $2D + t$ vers du $2D$ sans trop perdre de relations spatiales entre pixels et en tenant compte des relations temporelles. Ensuite, ces nouvelles images de représentations des pixels via la notion de chemins, sont utilisées pour alimenter un CNN classique dans un objectif de classification. Avec la représentation proposée, les convolutions $2D$ conduisent à une extraction de caractéristiques spatio-temporelles. Les filtres entraînés ont des poids liés à l'évolution temporelle et d'autres liés à l'évolution spatiale, enfin, la combinaison des deux porte des informations sur l'évolution spatio-temporelle. En considérant les convolutions $2D$ sur ce type d'images, nous pouvons également bénéficier d'un modèle

pré-entraîné, par exemple sur la base ImageNet dans un problème de classification similaire. Une telle initialisation des poids du CNN est moins applicable pour les études $1D$ car il est plus difficile de bénéficier de grands ensembles de données publiques, à l'échelle d'ImageNet et de réseaux pré-entraînés.

Références

[1] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes. Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sensing*, 9(1) :95–108, 2017.

[2] C. Pelletier, G.I. Webb, and F. Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11(5) :523–534, 2019.

[3] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring : A review. *International Journal of Remote Sensing*, pages 1565–1596, 2004.

[4] L. Bruzzone and D.F. Prieto. Automatic analysis of the difference image for unsupervised change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 38(3) :1171–1182, 2000.

[5] J. R. Jensen. Urban change detection mapping using Landsat digital data. *Cartography and Geographic Information Science*, 8(21) :127–147, 1981.

[6] R.D. Johnson and E.S. Kasischke. Change vector analysis : A technique for the multispectral monitoring of land cover and condition. *International Journal of Remote Sensing*, 19(16) :411–426, 1998.

[7] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor. Detecting trend and seasonal changes in satellite image time series. *Remote Sensing of Environment*, 114(1) :106–115, 2010.

[8] C. Senf, P.J. Leitao, D. Pflugmacher, S. Van der Linden, and P. Hostert. Mapping land cover in complex mediterranean landscapes using landsat : Improved classification accuracies from integrating multiseasonal and synthetic imagery. *Remote Sensing of Environment*, 156 :527–536, 2015.

[9] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off : a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3) :606–660, 2017.

[10] L. Andres, W.A. Salas, and D. Skole. Fourier analysis of multi-temporal AVHRR data applied to a land cover classification. *International Journal of Remote Sensing*, 15(5) :1115–1121, 1994.

[11] P. Ravikumar and V. S. Devi. Weighted feature-based classification of time series data. In *CIDM, Procs.*, pages 222–228, 2014.

[12] F. Petitjean, C. Kurtz, N. Passat, and P. Gançarski. Spatio-temporal reasoning for the classification of satellite image time series. *Pattern Recognition Letters*, 33(13) :1805–1815, 2012.

[13] M. Chelali, C. Kurtz, A. Puissant, and N. Vincent. Urban land cover analysis from satellite image time series based on temporal stability. In *JURSE, Procs.*, pages 1–4, 2019.

[14] F. Petitjean, J. Inglada, and P. Gançarski. Satellite image time series analysis under time warping. *IEEE Transactions on Geoscience and Remote Sensing*, 50(8) :3081–3095, 2012.

[15] B. Huang, K. Lu, N. Audebert, A. Khalel, Y. Tarabalka, J.M. Malof, and A. Boulch. Large-scale semantic classification : Outcome of the first year of inria aerial image labeling benchmark. In *IGARSS, Procs.*, pages 6947–6950, 2018.

[16] M. Russwurm and M. Korner. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multispectral satellite images. In *EarthVision@CVPR, Procs.*, pages 1496–1504, 2017.

[17] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(10) :1685–1689, 2017.

[18] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.A. Muller. Deep learning for time series classification : A review. *Data Mining and Knowledge Discovery*, 33(4) :917–963, 2019.

[19] N. Di Mauro, A. Vergari, T. M. A. Basile, F. G. Ventola, and F. Esposito. End-to-end learning of deep spatio-temporal representations for satellite image time series classification. In *DC@PKDD/ECML, Procs.*, pages 1–8, 2017.

[20] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV, Procs.*, pages 4489–4497, 2015.

[21] L. Grady. Multilabel random walker image segmentation using prior models. In *CVPR, Procs.*, pages 763–770, 2005.

[22] F.N. Iandola, M.W. Moskewicz, K. Ashraf, S. Han, W.J. Dally, and K. Keutzer. Squeezenet : AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *Computing Research Repository*, abs/1602.07360, 2016.