ConvEntion: Classification des séries chronologiques d'images astronomiques à l'aide d'attention convolutive

Anass BAIROUK[1], Marc CHAUMONT[1, 2], Dominique FOUCHEZ[3], Jérome PASQUET[4, 5], Frédéric COMBY[1]

[1]LIRMM, Univ. Montpellier, CNRS
161 rue Ada, 34392 Montpellier Cedex 05, France

[2]Univ. Nîmes
Place Gabriel Péri, 30000 Nîmes Cedex 01, France

[3]CPPM, Univ. Aix-Marseille, CNRS
163, avenue de Luminy, 13288 Marseille Cedex 09, France

[4]Groupe AMIS - Univ. Montpellier 3,
Route de Mende, 34090 Montpellier, France

[5]UMR TETIS, AgroParisTech, CIRAD, INRAE
500, rue Jean-François Breton, 34093 Montpellier, France

Anass.Bairouk@lirmm.fr, Marc.Chaumont@lirmm.fr
Dominique.Fouchez@cppm.in2p3.fr, Jerome.Pasquet@univ-montp3.fr,
Frederic.Comby@lirmm.fr

Résumé – L'utilisation de séries temporelles d'images astronomiques suscite un intérêt grandissant dans la communauté scientifique. Par ailleurs, avec la massification des données, il est nécessaire de proposer des solutions d'analyse automatique. Dans cet article, nous proposons une nouvelle approche basée sur l'apprentissage profond pour classer différents types d'objets célestes en utilisant les séquences d'images issues des télescopes. Nous appelons notre approche ConvEntion (abréviation de *CONVolutional attENTION*). Elle est basée sur l'utilisation conjointe de convolutions et de transformeurs. Ceci constitue une innovation dans le domaine du traitement des séries temporelles d'images. Sur un sous-ensemble de données issues de la base SDDS nous améliorons la précision de 7 % par rapport aux approches de l'état de l'art utilisant des séries temporelles d'images.

Abstract – We propose a novel approach based on deep learning for classifying different types of space objects directly using images. We name our approach ConvEntion which stands for CONVlutional attENTION. Our approach is based on Convolutions and Transformers, such a proposition is new for the treatment of astronomical image time series. Our solution can be applied to different types of image datasets with any number of bands. Besides, we integrated saptio-temporal features all along the model which yield great results with an increase of accuracy of 7% compared to state of the art approaches that uses image time series.

1 Introduction

La communauté scientifique en astronomie est confrontée à un défi considérable depuis quelques années car les télescopes deviennent de plus en plus puissants et peuvent observer un volume toujours plus grand de l'univers observé ce qui génère une quantité massive de données. Il est donc nécessaire de proposer des solutions de classification automatique des objets célestes.

Dans cet article nous proposons un nouveau transformeur basé sur une architecture d'apprentissage en profondeur pour classer les séries temporelles d'images (STI) astronomiques. Contrairement à la majorité des travaux qui séparent l'extraction des caractéristiques spatiales et temporelles, nous combinons les deux étapes en une seule. Nous proposons également une solution pour le problème d'observations manquantes. Dans la section 2, nous passons en revue les travaux liés à la classification de STI. Dans la section 3, nous présentons notre architecture nommée *ConvEntion*. Dans la section 4, nous présentons le jeu de données utilisé, et nous comparons les résultats obtenus avec ceux des propositions de l'état-de-l'art. Enfin, nous concluons en section 6.

2 État de l'art

Traditionnellement, le classement d'une série d'images (vignettes) centrées sur un objet céleste passe par une phase de prétraitement appelée photométrie. Le flux (i.e. l'intensité lumineuse) de l'ensemble des vignettes est calculé pour obtenir une série temporelle de scalaires (la valeur du flux) pour chaque objet observé dans différentes bandes spectrales. Les astrophysiciens nomment ces séries temporelles les courbes de lumière.

De nombreuses méthodes ont été développées pour effectuer la classification par courbes de lumière. Parmi elles on trouve [1] qui utilise un réseau de neurones récurrent (RNN), [2] qui repose sur une architecture neuronale appelée PELICAN, ou bien [3] qui génère une carte 2D de chaleur qui est transmise à un réseau de neurones convolutif.

Les travaux récents éliminent le passage par les courbes de lumière et proposent d'utiliser directement les images. [4] et [5] utilisent des RNNs après avoir passé les images à travers un CNN. Notons que ces deux articles ont montré des résultats prometteurs pour les STI astronomiques. Plus généralement, si l'on s'intéresse à la littérature récente de l'apprentissage traitant des STI, trois solutions émergent : celles basées sur les RNN, les 3D-CNN ou les Transformeurs.

Les approches basées RNNs sont nombreuses, et peuvent être divisées en deux catégories. La première gère les caractéristiques spatiales séparément des caractéristiques temporelles comme dans [4, 5]. Un CNN gère les caractéristiques spatiales pour les transmettre ultérieurement au RNN (un LSTM ou un GRU). La deuxième catégorie intègre quant à elle la convolution à l'intérieur de la cellule RNN comme dans (ConvLSTM) [7]. Les expériences indiquent que le réseau ConvLSTM bat régulièrement le LSTM entièrement connecté (FC-LSTM). Une extension proposée dans [8] utilise un nouveau type de RNN plus robuste appelé ConvSTAR avec moins de paramètres.

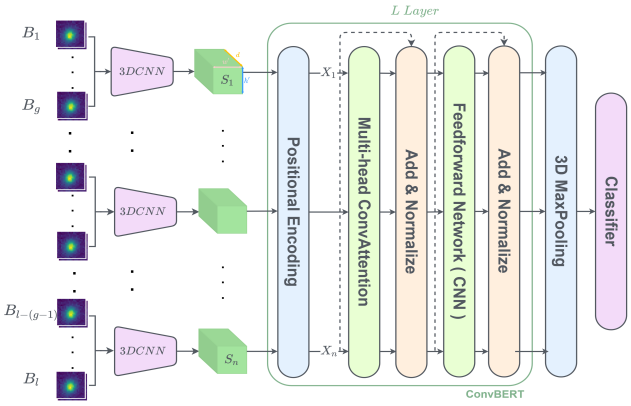

FIGURE 1 – L'architecture générale

3 Proposition

Dans cette section, nous proposons notre réseau basé sur une combinaison de convolution et d'auto-attentions. Ce modèle gère entre autre les particularités rencontrées en astrophysique telles que la rareté des données, les observations manquantes et le déséquilibre des classes. Notre architecture est présentée à la figure 1. Dans la sous-section 3.1, nous présentons le reformatage des séries temporelles d'images. Dans la sous-section 3.2, nous présentons le 3DCNN, dans la section 3.3 nous présentons l'encodage positionnel et enfin dans la sous-section nous présentons notre BERT (Bidirectional Encoder Representations from Transformers) convolutif 3.4.

3.1 Reformatage de la STI

Un réseau est alimenté avec une série d'images I_t où $I_t \in \mathbb{R}^{h\times w\times 5}$, avec h et w la hauteur et la largeur de l'image, 5 le nombre de bandes (les bandes sont (u, g, r, i, z)), et t est le temps (jour d'acquisition). En astrophysique, certaines bandes sont manquantes. Afin d'éviter de donner au modèle des images avec des bandes vides, nous "divisons" les images, et considérons chaque bande $B_k \in \mathbb{R}^{h\times w}$ comme une image (k est l'indice dans la série de bandes). Les bandes manquantes ne sont donc pas présentes. Afin de conserver l'information sur le numéro de bande, c'est-à-dire l'identifiant de la bande (id) qui appartient à $\{1, .., 5\}$, nous proposons de *représenter* le nombre id par une matrice de dimension $h \times w$, et de la concaténer avec sa bande associée, B_k, avant de le transmettre à notre architecture.

Le codage d'un $id \in \{1, .., 5\}$ est réalisé grâce à une couche dense qui prend en entrée le nombre id et retourne une matrice $E_{id} \in \mathbb{R}^{h\times w}$. Ainsi la série, $B \in \mathbb{R}^{l\times h\times w\times 1}$, de bandes, est transformée en une série d'images composées de deux canaux : les bandes, B_k, et l'encodage de leur id respectif, E_{id}. La série résultante est notée $B' \in \mathbb{R}^{l\times h\times w\times 2}$ avec l le nombre de bandes.

3.2 3D-CNN

Les mécanismes d'auto-attention des transformeurs classique ont une complexité quadratique en fonction de la longueur de la séquence. La complexité est encore plus grande pour nous car nous utilisons des convolutions et des tenseurs 3D à l'intérieur du mécanisme d'attention. Certaines solutions ont été proposées pour réduire cette complexité comme [12] qui utilise une matrice de faible rang pour approximer le mécanisme d'auto-attention, ou [13] qui utilise des transformeurs avec une attention à rang complet. De notre côté, nous proposons de réduire la longueur de la séquence avant de la transmettre au transformeur.

Pour cela, nous utilisons un réseau de neurones à convolution 3D (*3D-CNN*) qui transforme la série $B' \in \mathbb{R}^{l\times h\times w\times 2}$ en $S \in \mathbb{R}^{n\times h'\times w'\times d}$. Le 3D-CNN traitera la série B' par groupe de g images tel que la taille de la séquence S est réduite à $n = l/g$ ainsi que les dimensions des images puisque $h' < h$ et $w' < w$.

3.3 Ajout de l'information de position dans la séquence issue du 3D-CNN

La séquence S produite par le 3D-CNN est ensuite passée à notre BERT (*Convolutional Bidirectional Encoder Representations from Transformers*) convolutif. À notre connaissance, c'est la première fois qu'un BERT convolutif est proposé.

La séquence S composée de n images $S_i \in \mathbb{R}^{h'\times w'\times d}$ issues du 3D-CNN doit être traité par le transformeur. Or, les transformeurs ne maintiennent pas l'information sur l'ordre des images. Il est donc usuel d'ajouter, à la séquence d'images, l'information de position, c'est à dire l'index $i \in \{1, ..., n\}$ de chaque image (ainsi que le numéro du canal) pour que le transformeur prenne en compte l'ordre des images dans sa prédiction. L'astuce consiste à représenter chaque index i (plus le

numéro de canal) sous forme d'une image $P_i \in \mathbb{R}^{h' \times w' \times d}$, puis à sommer terme à terme chaque image S_i à chaque image P_i pour former une nouvelle séquence $X \in \mathbb{R}^{n \times h' \times w' \times d}$

$$\forall i \in \{1, \ldots, n\}\ X_i = S_i + P_i \tag{1}$$

La représentation d'un index i (plus le numéro de canal) sous forme d'une image P_i s'appelle le codage positionnel (positional encoding) et est réalisé en reprenant la formule issue du papier fondateur [9] :

$$P_i^{(k)} = \begin{cases} sin(i/10000^{k/d}), & si\ k\ est\ pair, \\ cos(i/10000^{k/d}), & si\ k\ est\ impair. \end{cases} \tag{2}$$

avec $i \in \{1, ..., n\}$ et $k \in \{1, ..., d\}$ l'indice sur le numéro de bande.

Notons que cette opération de codage positionnel est réalisée avant chaque couche d'attention convolutionelle multi-tête ("multi-head convolutional self-attention").

3.4 BERT convolutionel

Nous avons repris l'encodeur du transformeur de [11] pour créer notre BERT convolutionnel. La séquence d'entrée est la séquence d'images $X \in \mathbb{R}^{n \times h' \times w' \times d}$. Chaque image X_i avec $i \in \{1, .., n\}$ passe à travers une couche de convolution. On obtient alors 3 séquences distinctes, chacune de taille $\mathbb{R}^{n \times h' \times w' \times d'}$, nommées séquence *requêtes*, séquence *cartes de clés*, séquence *valeurs* et notées respectivement Q, K, V. Le nombre de canaux d' est égal à d/h avec h le nombre de têtes d'attention.

Chaque paire d'images $(Q_i, K_j) \in \mathbb{R}^{h' \times w' \times d' \times 2}$, avec $i, j \in \{1, .., n\}$, est transmise à un sous-réseau, noté M_θ permettant de générer une *carte d'attention*, $H_{(i,j)}$, de dimension $\mathbb{R}^{h \times w}$, associée aux deux images tel que :

$$H_{(i,j)} = M_\theta(Q_i, K_j) \qquad \forall i, j \in \{1, .., n\}. \tag{3}$$

$$H_i = SoftMax(H_{(i)}) \quad ou \quad H_i \in \mathbb{R}^{h' \times w' \times n} \tag{4}$$

Le réseau M_θ prend une image requêtes Q_i et une image cartes de clé K_i, effectue une somme élément par élément, puis applique une couche de convolution afin d'obtenir la carte d'attention $H_{(i,j)}$. On applique ensuite une opération softmax le long de la troisième dimension de taille n .

Les n^2 cartes d'attentions servent à indiquer l'importance de chaque image de la séquence valeurs V, relativement à chacune des autres images. Ainsi, pour chaque position $i \in \{1, .., n\}$ nous effectuons une somme pondérée des n images valeurs V_j ; la pondération passe par les cartes d'attention relatives à la position i et la position j tel que nous obtenons les images pondérées, V_i', suivantes :

$$\forall i \in \{1, .., n\}, V_i' = \sum_{j=1}^{n} H_{(i,j)} \otimes V_j, \tag{5}$$

avec $\otimes$ l'opérateur de multiplication élément par élément.

Nous obtenons ainsi une séquence $V' \in \mathbb{R}^{n \times h' \times w' \times d'}$. Enfin, nous concaténons toutes les séquences V_i' issues des différentes têtes d'attention.

En toute fin de réseau, nous trouvons une couche de "3D Max-pooling" suivi du classifieur qui contient deux couches denses avec une dropout et de l'activation.

4 Ensemble de données et expériences

4.1 Description de la base de données

Nous avons construit notre base de données en utilisant le Sloan Digital Sky Survey (SDSS) [6] qui est une base d'images astrophysiques bien établie. Cette base contient des courbes de lumière multi-bandes (ugriz) ainsi que des STI. L'étiquetage spectroscopique (vérité terrain) est également disponible pour certaines observations. Nous simplifions le jeu de données en 4 types : AGNs, Variables, SNIa, SNAutre (Tableau 1). Notez que les supernovae de type Ia (SNIa) présentent un intérêt particulier car les scientifiques les utilisent pour déterminer leur distance à notre galaxie.

Nom de la classe	Nombre
AGN	906
SNIa	1988
Variable	3225
SNAutre	2130

TABLE 1 – Classes et nombres d'objets par classe.

4.2 Détails d'implémentation

Les supernovae ne sont pas toutes "confirmées" par spectroscopie, ce qui signifie que celles qui ne sont pas confirmées peuvent être mal classées. Afin que notre modèle généralise sur les données confirmées par spectroscopie, nous réalisons l'apprentissage en deux étapes, et divisons les données en deux ensembles. Le premier contient uniquement les données spectroscopiquement *non-confirmées*, et le second contient les données spectroscopiquement *confirmées*. Nous entraînons d'abord le modèle avec les données non-confirmées, puis nous affinons le modèle sur les données confirmées. Nous avons utilisé l'augmentation des données pour le suréchantillonnage de notre base de données. Pour cela, nous modifions simplement l'ensemble de données afin de supprimer ce déséquilibre en augmentant le nombre de classes minoritaires et en diminuant le nombre de classes majoritaires jusqu'à obtenir un ensemble de données équilibré. Toutes les architectures présentées dans cet article suivent ce même processus et sont implémentées en PyTorch.

Nous avons entraîné plus de 30 modèles afin de déterminer les meilleurs hyperparamètres pour notre architecture qui contient 1,6 million de paramètres. La taille des groupes utilisés par le 3DCNN est $g = 3$. La longueur de la séquence S entrant dans BERT est $n = 33$. Le nombre de couches convolutives de BERT est de 2 et le nombre de têtes d'attention $h = 4$. Nous avons utilisé l'optimiseur Adam avec un taux d'apprentissage de 10^{-3}, une perte de type cross-entropy, un dropout de 0.3, et des batchs de 128 séquences. Nous stoppons l'apprentissage au bout de 100 époques. Nous avons entraîné tous les modèles avec 4 GPU GeForce RTX 2080 Ti. Nous avons testé le modèle sur des données confirmées par spectroscopie que le modèle n'a jamais vues. Nous avons utilisé 10% de l'ensemble des données confirmées pour le test.

4.3 Expériences

TABLE 2 – Comparaison de plusieurs architectures.

Modèle	Bandes	BDD	TC%
ConvEntion (notre)	ugriz	Images	**79.83**
ConvEntion (notre)	g	Images	76.89
CNN+GRU [5]	g	Images	63.67
CNN+LSTM [4]	g	Images	63.00
SuperNNova (Bayes) [1]	ugriz	CL	65.54
SITS-BERT [14]	ugriz	CL	67.43
SCONE (CNN) [3]	ugriz	CL	62.57
SuperNNova (RNN) [1]	ugriz	CL	56.30

CL : Courbes de lumière, TC : Taux de classification

Le tableau 2 résume les résultats obtenus en utilisant les STI où les courbes de lumière avec différentes approches de l'état de l'art. Notre modèle *ConvEntion* obtient la meilleure précision avec 79,83 %, soit 7 % de plus que les meilleurs résultats sur les images de [5], et 12% de plus que le meilleur modèle utilisant les courbes de lumière. Cela confirme l'intérêt à utiliser les images plutôt que les courbes de lumière. En effet les images contiennent plus d'informations que les valeur de flux présentes dans une courbe de lumière. ConvEntion obtient de meilleures performances par rapport aux autres modèles basés sur les images comme [4]. Les transformeurs traitent une séquence dans son entièreté, contrairement aux RNN qui ne capturent pas très bien les dépendances temporelles longues. Enfin, l'apprentissage de notre transformeurs est plus rapide (même nombre de paramètres). L'entrainement de notre modèle a pris seulement 3 heures alors que ceux des autres modèles basés images ont pris 5 heures sur une unique carte de GPU.

5 Conclusion

Dans ce travail, nous proposons une méthode pour la classification de séries temporelles d'images astronomiques nommée *ConvEntion*. Elle est entièrement basée sur la combinaison d'un réseau convolutif et d'un transformeur. De plus, nous avons proposé un moyen de gérer les observations manquantes et le déséquilibre des données. Enfin, nos expériences sont réalisées dans des conditions réalistes en utilisant des données spectro confirmées.

Notre travail confirme que les images contiennent plus d'informations que les courbes de lumière et que de meilleures performances peuvent être obtenues en utilisant des images. Dans le futur, nous prévoyons d'étendre ConvEntion en utilisant l'apprentissage auto-supervisé.

Acknowledgment

Ce travail a été réalisé grâce au soutien de l'ANR DEEPDIP projet (ANR-19-CE31-0023). Cette publication utilise les données du Sloan Digital Sky Survey (SDSS). Le financement de SDSS-III a été assuré par la Fondation Alfred P. Sloan.

Références

[1] Moller, A, and T, Boissiere. "SuperNNova : an open-source framework for Bayesian, neural network-based supernova classification". Monthly Notices of the Royal Astronomical Society 491, no.3 (2019)

[2] J. Pasquet, J. Pasquet, M. Chaumont, and D. Fouchez. "PELICAN : deeP architecturE for the LIght Curve ANalysis".Astronomy & Astrophysics 627 (2019)

[3] Helen Qu, Masao Sako, Anais Möller, and Cyrille Doux 2021. SCONE : Supernova Classification with a Convolutional Neural Network. The Astronomical Journal, 162(2), p.67.

[4] Carrasco-Davis et al. "Deep Learning for Image Sequence Classification of Astronomical Events".Publications of the Astronomical Society of the Pacific 131, no.1004 (2019)

[5] Gomez, Catalina, Mauricio, Neira, Marcela, Hernandez Hoyos, Pablo, Arbelaez, and Jaime E, Forero-Romero. "Classifying image sequences of astronomical transients with deep neural networks". Monthly Notices of the Royal Astronomical Society 499, no.3 (2020)

[6] Jon A. Holtzman et al. "The Sloan digital sky survey-ii : photometry and supernova Ia light curves from BTHE 2005 data".The Astronomical Journal 136, no.6 (2008)

[7] Shi, Xingjian, et al. "Convolutional LSTM network : A machine learning approach for precipitation nowcasting." Advances in neural information processing systems 28 (2015).

[8] Turkoglu, Mehmet Ozgur, et al. "Crop mapping from image time series : Deep learning with multi-scale label hierarchies." Remote Sensing of Environment 264 (2021)

[9] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[10] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale." The International Conference on Learning Representations (2021).

[11] Liu, Zhouyong, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Shilei Sun, Chunguo Li, and Luxi Yang. "Convtransformer : A convolutional transformer network for video frame synthesis." arXiv preprint arXiv :2011.10185 (2020).

[12] Wang, S., Li, B.Z., Khabsa, M., Fang, H. and Ma, H., 2020. Linformer : Self-attention with linear complexity. arXiv preprint arXiv :2006.04768.

[13] Krzysztof Choromanski et al "Rethinking Attention with Performers." International Conference on Learning Representations (2021).

[14] Y. Yuan and L. Lin, "Self-Supervised Pretraining of Transformers for Satellite Image Time Series Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2021.