Accepted to Astronomy & Astrophysics
March 13, 2023

ConvEntion: Astronomical Image Time Series Classification Using Convolutional attEntion

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

[1] LIRMM, University of Montpellier, France
e-mail: {anass.bairouk, frederic.comby, marc.chaumont}@lirmm.fr
[2] Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France
e-mail: {fouchez, bautista}@cppm.in2p3.fr
[3] University of Nimes, France
[4] Groupe AMIS, Paul Valéry University Montpellier 3, France
e-mail: jerome.pasquet@univ-montp3.fr
[5] UMR TETIS, INRAE/CIRAD/CNRS

March 13, 2023

ABSTRACT

Aims. Astronomical Image Time Series has received increasing attention in recent years. Indeed, many surveys for the follow-up of transient objects are in progress or under construction such as the Vera Rubin Observatory Legacy Survey for Space and Time (LSST) which will produce huge amounts of these time series. The associated scientific topics are numerous, from the study of objects in our galaxy to the observation of the most distant supernovae to measure the expansion of the universe. With the large amount of data available, the need for robust automatic tools to detect and classify celestial objects is growing like never before. It opens a big opportunity for deep learning to excel in the domain of astronomy.

Methods. Our making hypothesis is that astronomical images contain more information than light curves. So, in this paper, 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 CONVolutional attENTION. Our approach is based on Convolutions and Transformers, which is new for the treatment of astronomical image time series. Our solution integrates Spatio-temporal features and can be applied to various types of image datasets with any number of bands.

Results. In this work, we solved various problems that the datasets suffer from and present new results on classification using astronomical image time series with an increase of accuracy of 13% compared to state-of-the-art approaches that use image time series and a 12% increase compared to approaches that use light curves.

Key words. Transformer, ConvEntion, Astronomical Image Time Series, Convolutional Attention, Classification, Supernovae, 3D Convolution Network

1. Introduction

The scientific community in astronomy is facing a considerable challenge in the last few years as the tools for observing the universe are improving. Telescopes are becoming more powerful and can observe a huge part of the universe which generates a massive amount of data. Processing and analyzing these data is very demanding in computation and human resources. With the promises of The Vera Rubin Observatory Legacy Survey for Space and Time (LSST) (Ivezić et al. 2019), it is about to change the field by uncovering 10 to 100 times more astronomical sources that fluctuate in the night sky than we have ever seen before. Some of these sources will be entirely new. It is estimated to alert 10 million new objects per night. These objects should all be classified. There are many types of objects, we can mention Active Galactic Nuclei (AGN), Variables, Cepheids, RR Lyrae, and Supernovae. The last one is the most important transient object for cosmology because increasingly large samples of Type Ia supernovae (SNe Ia) are being used to measure luminosity distances as a function of redshift in order to understand the origin of the acceleration of the expansion of the universe.

Traditionally the classification of these objects goes through many processes in a complicated pipeline. First of all, a preprocessing phase, called photometry, is conducted on a series of images to extract the flux per band, each band corresponds to a passband-like color filter. The number of bands can vary depending on the survey for example SDSS survey (Holtzman et al. 2008; Sako et al. 2014; Frieman et al. 2007) has five bands, and the Catalina survey (Drake et al. 2011) has only one band. Then, it generates a time series of brightness changes over time. These time series are called the light curves. Afterward, the light curve is fed to a machine learning classifier to provide the class of the object. Among all methods developed to perform such a classification, Möller & de Boissière (2020) introduced a model called SuperNNova: a supernova photometric classification framework that uses a Recurrent Neural Network (RNN) (Rumelhart et al. 1985; Hochreiter & Schmidhuber 1997; Cho et al. 2014) to classify different types of supernovas like SNIa, SNIb, SNIIP, and many others using only light curves. The proposition yields good results because Bayesian Neural Networks (BNN) are known to be robust to overfitting, and can easily learn from small datasets. Nonetheless, BNNs are

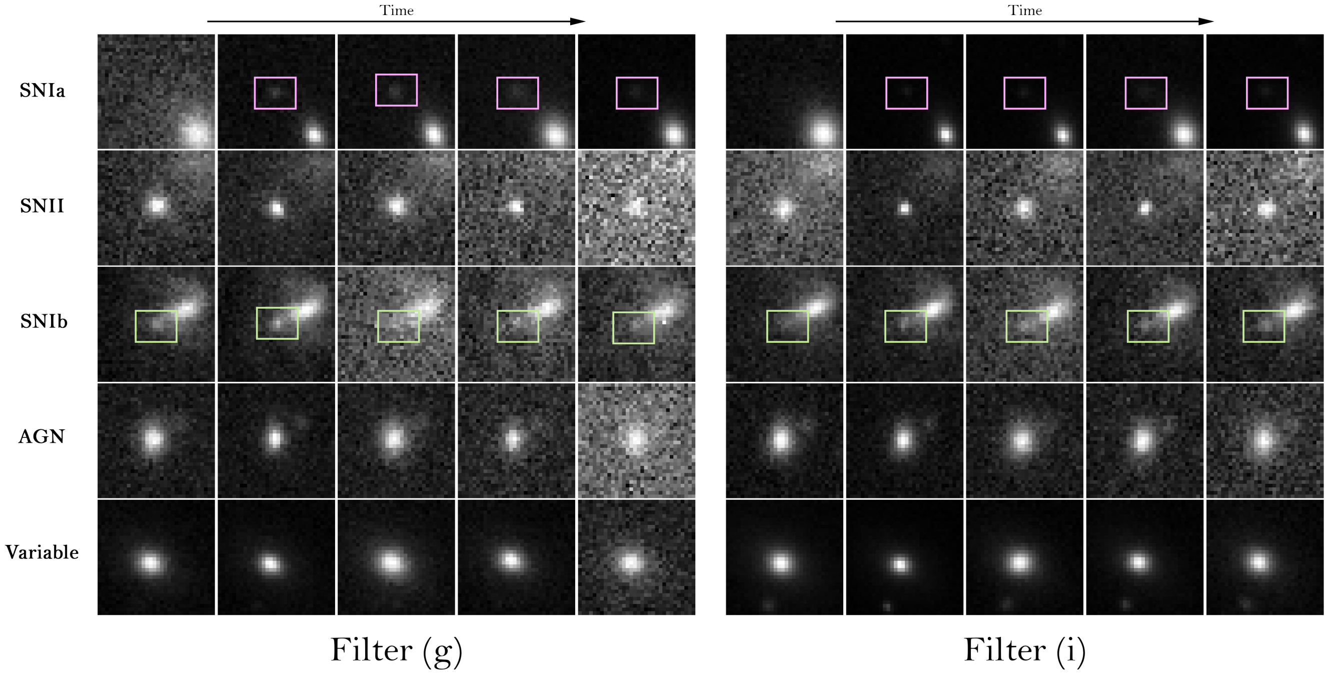

Fig. 1. A sample of some objects present in our dataset. Each image in filter g/i corresponds to a different observation with the same filter.

significantly more complex than standard neural networks and computationally expensive. Boone (2019), (winner of the photometric classification challenge PLAsTiCC (PLAsTiCC-team et al. 2018; Hložek et al. 2020)) presented a model based on Gaussian process augmentation of the light curve and then train it on boosted decision tree classifier. Pasquet et al. (2019) created a deep architecture called PELICAN that accepts only light curves and redshift as input. PELICAN can handle light curves with sparsity and irregular sampling. Others chose to add more preprocessing before training a model. Qu et al. (2021) proposed a novel approach where they generate a 2D image heatmap from light curves using 2D Gaussian process regression which they fed to convolutional neural networks to classify different types of supernovae. The approach yields great results on PLAsTiCC data with an accuracy of 99.73% on the binary classification of SNIa and non-SNIa. The methods that use light curves for classification still have some limitations. For instance in order to generate a light curve, we should align correctly the two consecutive images and we must lower the quality of one of the two images to subtract them to get the flux, which could lead to a loss of information. Some dedicated algorithms called scene modeling can mitigate such issues on blended objects but are very computer resources consuming. Most importantly, the scene information, i.e the background of the transient object, is in general not taken into account in the classification. Several recent works have proposed to eliminate the feature extraction and light curve phase and focus on classifying the objects using only images. Carrasco-Davis et al. (2019) and Gómez et al. (2020) used RNN to classify the sequences after passing the images through a CNN to extract the spatial features. They forward the output to the RNN (GRU/LSTM) to extract the temporal characteristics and classify the object where (Gómez et al. 2020) applied their model to only transient objects while Carrasco-Davis et al. (2019) classify variables and transient. These two papers showed promising results for the Astronomical Image Time Series (AITS). Therefore we followed the same path to improve the classification and also solve some challenges of AITS which have never been tackled before.

ITS classification has always been one of the challenging areas of deep learning. In addition to spatial characteristics, you also need to give importance to the temporal aspects, which makes traditional feedforward networks ineffective. Due to little research done on ITS in astronomy, we will need to import new technics from other fields of research. Most of the research in ITS classification is done in two major domains, action recognition, where the goal is to classify the type of human action (Shi et al. 2015; Ji et al. 2013), the second one is landscape classification using satellite images (Turkoglu et al. 2021). These two fields have covered many of the essential methods to handle ITS. RNN-Based approaches use recurrent neural networks to manage the aspect of time in the classification. These approaches have two main categories. The first one handles the spatial features separately from the temporal features. Carrasco-Davis et al. (2019) and Gómez et al. (2020) used precisely this method where the CNN handles the spatial characteristics to pass it later to the RNN, which might be LSTM (Hochreiter & Schmidhuber 1997) or GRU (Cho et al. 2014). The second category combines convolution inside the RNN cell, thus maintaining the spatial structure of the input which leads to extracting spatial-temporal features in the sequence. This method was first introduced by Shi et al. (2015). They demonstrated how to create an end-to-end trainable model using the convolutional LSTM (ConvLSTM). Experiments indicate that their ConvLSTM network regularly beats fully connected LSTM (FC-LSTM) in capturing Spatio-temporal correlations. Using satellite images, Turkoglu et al. (2021) proposed a new type of RNN called ConvSTAR, which has fewer parameters than the LSTM and GRU. Another way of achieving the classification of ITS is by using convolution neural networks. Ji et al. (2013) created a new 3D CNN model for action recognition. This model pulls features from spatial and temporal dimensions collecting motion information contained in several consecutive frames. Meanwhile, some of the latest developments have abandoned convolutions and RNNs to replace them with only Transformers. Liu et al. (2022) and Yan et al. (2022) proposed an improved supervised Transformer for image classification. On the other hand, Zhou et al. (2022) and Bao et al. (2022) proposed more complex Transformers that are self-supervised.

In this work, we develop a new deep learning Transformer-based architecture to classify AITS. Unlike other works that separate spatial and temporal feature extraction, we combine these two steps by performing Spatio-temporal feature extraction in one step. It improves the capacity of the network to recognize the objects. We also propose a solution for the missing observations problem, which showed a significant improvement in the accuracy of the model. To illustrate the performances of our model, we tested it with actual data from the SDSS survey (Holtzman et al. 2008; Sako et al. 2014; Frieman et al. 2007). In section 2, we describe the dataset that we used in our work. Section 3 introduces our architecture ConvEntion. We will go over the role of each component of the model. In section 4, we present the results of our work with some statistics about the performance and some comparisons with other architectures used for image time series classification. Finally, in section 5 we conclude and present perspectives about this work.

2. Dataset

2.1. Database description

The Sloan Digital Sky Survey (SDSS) (Holtzman et al. 2008; Frieman et al. 2007) is a very ambitious and successful large-scale survey program using a dedicated 2.5-meter telescope at Apache Point Observatory, New Mexico, equipped with photometric and spectroscopic instruments that have released images, spectra and catalog information for several hundred million celestial objects. The dataset used in this paper has been collected during the SDSS Supernova Survey (Sako et al. 2014), one of three components (along with the Legacy and SEGUE surveys) of SDSS-II, a 3-year extension of the original SDSS that operated from July 2005 to July 2008. The Supernova Survey was a time-domain survey, involving repeat imaging of the same region of the sky every other night, weather permitting.

The images are obtained through five wide-band filters (Fukugita et al. 1996) named u', g', r', i' and z', simplified in u, g, r, i and z in the following, which corresponds to an effective mid-point wavelength of u (365nm), g (475nm), r (658nm), i (806nm) and z (900nm). The survey region observed repeatedly over 3 years is a 2.5-degree-wide stripe centered on the celestial equator in the Southern Galactic Cap that has been imaged numerous times in the last twenty years, allowing the construction of a big image database for the discovery of new celestial objects. Most of the sources, which included Galactic Variable Stars, Active Galactic Nuclei (AGN), Supernovae (SNe), and other astronomical transients, have been processed to generate multi-band (ugriz) light curves. The imaging survey is reinforced by an extensive spectroscopic follow-up program that uses spectroscopic diagnostics to identify SNe and measure their redshifts. Light curves have been evaluated during the survey to provide an initial photometric type of the SNe, and a selected sample of sources is targeted for spectroscopic observations.

In order to investigate the classification from images rather than light curves, we acquired the images from the public SDSS dataset through their platform. Our dataset contains many types of supernovas (see table 1 and (Sako et al. 2014)). "Unknown" are mainly very sparse or poorly measured transient candidates, "Variables" have signals spanning over two seasons, and AGN has a spectral signature. The three other classes are supernovae of type Ia, Ib/c, and II. Among supernovae, the typing is performed from spectroscopy or from the lightcurve using different machine learning techniques (see Sako et al. 2014). We have grouped the non-Ia supernovas because we focus, in this study, only on the Ia type for their interest in cosmology as standard candles and also because of the small number of non-Ia with spectral signatures. The very small class of three SLSN bright objects has been added to the non-Ia supernovae. Figure 1 shows an example of astronomical image time taken from the SDSS dataset.

Object name	Count
AGN	906
SNIa	499
SNOther	89
Unknown	2009
Variable	3225
SNOther_PT	2041
SNIa_PT	1448

Table 1. Number of objects per class in the SDSS dataset. PT: Photometrically Typed which means that the SNs are not spectroscopically verified

2.2. Challenges

Most of the astronomical dataset suffers many problems that should be dealt with before feeding it to the classification algorithm. Among difficulties contributing to the challenging nature of AITS, one can mention class imbalance as shown in the table 1 of our dataset, we can clearly see that the classes we have are not balanced where the number of samples for variables is much bigger than SNIa. This imbalance impacts significantly machine learning models due to their higher prior probability, machine learning models tend to over-classify the larger class(es). As a result, instances belonging to the smaller class(es) are more likely to be misclassified than those belonging to the larger class(es). Another problem that impacts the model is the missing bands. Indeed, each time an image is acquired in an AITS it is captured through one filter among a set of up to five or more channels. So, an image of a celestial object can be taken in many channels but not necessarily at the same time. Which results in missing bands for a given time of observation (See figure 3). It is well known that the missing data impacts negatively the performance of the model if it is not dealt with. Gill et al. (2007) stated that an increasingly missing percentage of training data resulted in an increased testing error. Which requires us to propose a solution to mitigate the impact of missing data.

3. Methods

In this section, we propose a neural network based on a combination of convolution and self-attentions. The goal of the model is to handle the challenges that we mentioned previously like class

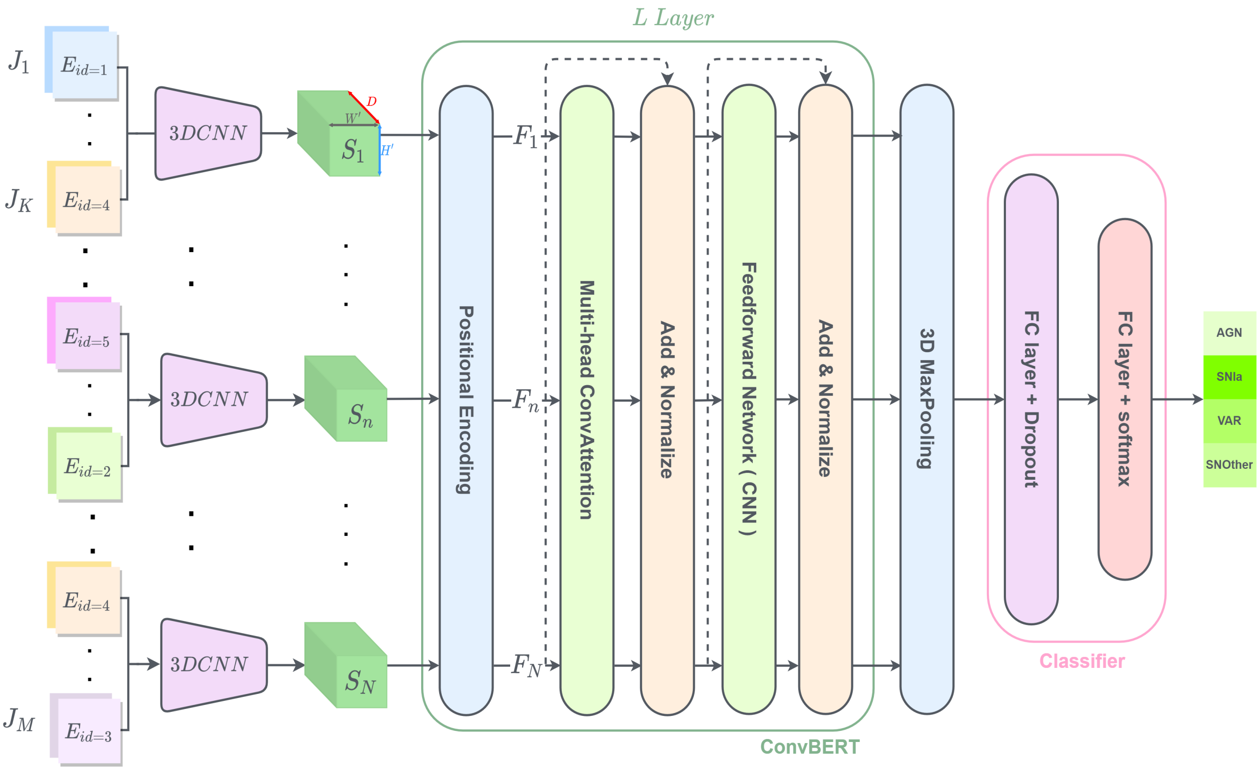

Fig. 2. The general architecture of the ConvEntion network. The image time series are first rearranged to embed the band information. Then each 3DCNN is fed with a sub-sequence of K inputs of the time series $J(\in \mathbb{R}^{M\times H\times W\times 2}$ for M elements of images of size HxW) to create the new downsized sequence $S(\in \mathbb{R}^{N\times H'\times W'\times D})$. S will be fed to the positional encoder in order to add the information about the position which outputs $F(\in \mathbb{R}^{N\times H'\times W'\times D})$. Then F is passed to ConvBERT which has L layers. The 3D Max-Pooling is used to downsize the output of ConvBERT for the classifier

imbalance, data sparsity, and missing observations. Figure 2 represent the general architecture of the ConvEntion model. The model takes as input the sequence of images that have been rearranged to embed the band information (See section 3.1 and figure 4). The sequence first passes through a 3DCNN to downsize its length. It allows for the reduction of the computation complexity of the model and also captures the local characteristics of the objects. The newly constructed sequence by the 3DCNN is fed to a convolutional BERT which will extract spatio-temporal features with high-level representation from the input. Finally, we pass the output of the convolutional BERT, which is a projection of our input into a high-level representation subspace, through a 3D Max-pooling to down-sample it, then to the final classifier to make the prediction. In the following subsections, we will explain each component in depth.

3.1. Data modeling

Throughout the document, vectors will be noted in bold capital letters, sizes in capital letters, and indices in lowercase.

To start with the **missing data problem**, a network dedicated to image time series is usually fed a sequence of images $\mathbf{I} \in \mathbb{R}^{H\times W\times 5}$ where H and W are respectively the height and width of the image and 5 is the number of channels representing the bands (u, g, r, i, z). However, we know, as explained earlier, that some bands are missing in the dataset. To fix this issue, instead of giving the model images with empty channels, hence introducing a bias to the network, we decided to separate the channels as individual images ($\mathbf{X} \in \mathbb{R}^{H\times W}$) simply skipping the empty channels. As a consequence, the information about the type of filter, which holds a crucial value for the network to accurately discriminate between objects, is also eliminated. Indeed, in an image with different channels, the order of the channels usually represents the type of the filter (See figure 3).

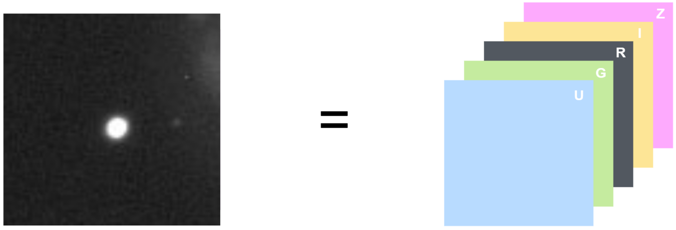

Fig. 3. Each image has five filters (u, g, r, i, z), The black channel represents the missing observation

In order to preserve this valuable information we should add the band type to the new 2D images $\mathbf{X}$. Knowing that the information about the type of the filter is a categorical feature, thus we will need to adapt it to the model 2D input representation. To do so we propose to use an embedding layer to encode the channel type before passing the input to the model. For each band (u, g, r, i, z), we assign a unique number $id \in \{1, 2, 3, 4, 5\}$. Then, an embedding layer $BandEmbed$ converts the band type id, which is a categorical features, into 2D dense representation E_{id} with $E_{id} \in \mathbb{R}^{H\times W}$ (See figure 4):

$$E_{id} = BandEmbed(id). \tag{1}$$

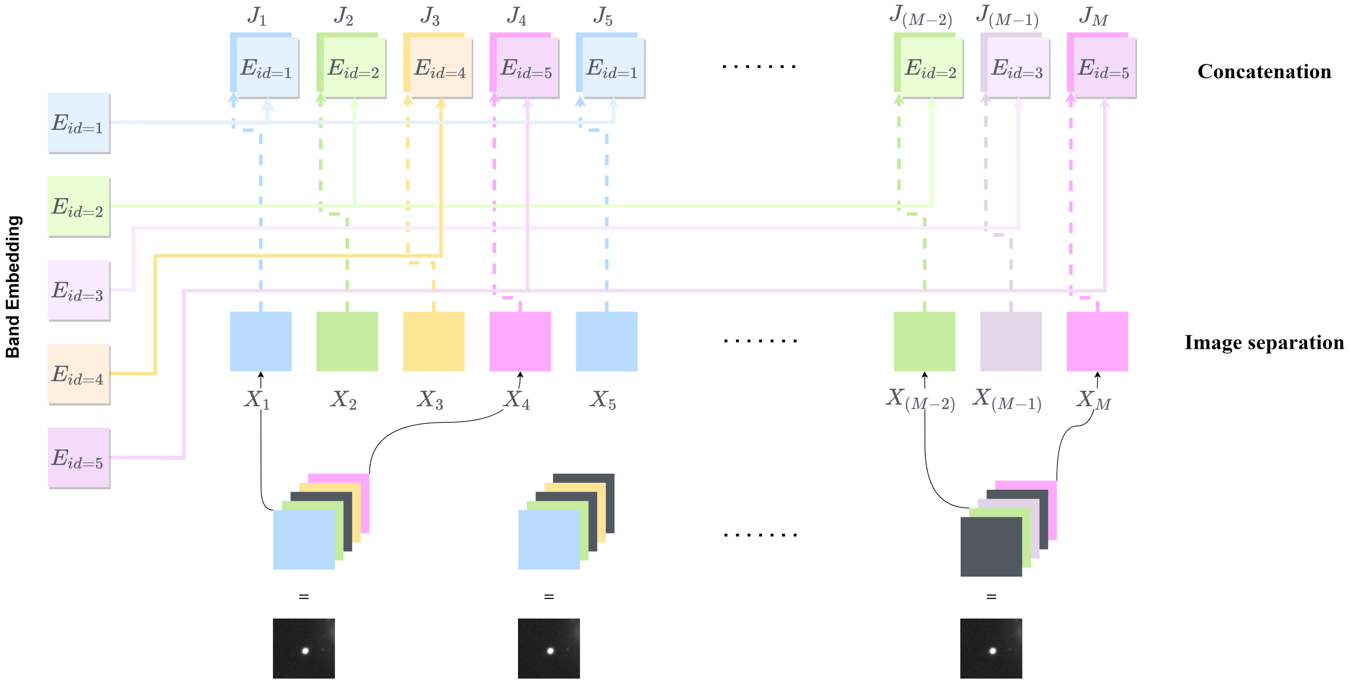

Fig. 4. Illustration of the handling of missing information by separating the bands. The empty channels are dropped, then we concatenate each image with a 2D representation of the band used to capture the image. The band embedding contains five band representations. The black channel represents the missing observation

The embedding layer is a fully connected layer that is reshaped to a 2D representation. The weights of *BandEmbed* are learnable. After getting the band embedding, we concatenate it with the new image to get our new input $J \in \mathbb{R}^{M \times H \times W \times 2}$ that contains the band information where M is the length of the sequence:

$$J_m = Concat(X_m, E_{id}), \qquad m \in \{1, .., M\}. \tag{2}$$

The problem of **class imbalance** is one of the major challenges for any machine learning project. Some tried to solve this problem by adding a new loss function to mitigate the impact of the class imbalance. For example Lin et al. (2017) proposed a loss function called “Focal loss” which applies a modulating term to the cross-entropy loss in order to focus the learning on hard misclassified examples. However, this approach tends to produce a vanishing gradient during backpropagation (Hossain et al. 2021). Other solutions propose the use of oversampling like SMOTE (Chawla et al. 2002). The authors proposed an approach where they synthesize new samples of the minority class. However, this solution was proposed mainly for tabular data. Knowing that our data are images that contain a way higher number of features than tabular data, it appears obvious that using SMOTE may not be optimal in our case. Dablain et al. (2021) introduced a solution based on SMOTE dedicated to images called DeepSMOTE. It aims at generating new images for the minority class. Once again, this approach is unsuitable in our case as our dataset is not composed of images but of a sequence of images and it will be too expensive to generate a whole new sequence. So, instead of generating a new one, we have used data augmentation and Weighted Random Sampling(WRS) (**?**) on our database. We oversample the dataset which translates to simply altering the dataset to remove such an imbalance by increasing the number of minority classes and undersampling the data by decreasing the majority classes until we have reached a balanced dataset. In our case, the WRS was applied on a batch level. We generate balanced batches based on the probability of a sample being selected. We weight each sample according to the inverse frequency of its label’s occurrence, and then sample mini-batches from a multinomial distribution based on these weights. This means that samples with high weights are sampled more often for each mini-batch. The same sample can be reused in other mini-batches of the same epoch to increase the minority class but with a data augmentation applied to it. Different methods of data augmentation were used: for example random drop of some steps from the whole sequence to create a new one, or sequence rotation, horizontal and vertical flip, and sequence shifting where we construct a smaller sequence from the original one which has a bigger length than the input length of ConvEntion. In our implementation we recall the dataset at every epoch, the transforms operation (augmentation) is executed, and then get different augmented data. Using this oversampling approach has drastically improved the performance of the model. We used the function *WeightedRandomSampler* from PyTorch (**?**) as an implementation of WRS.

3.2. 3D Convolution Network:

In several deep learning applications, large Transformer models have demonstrated fantastic success in obtaining state-of-the-art results. However, because the original Transformer’s self-attention mechanism consumes $O(M^2)$ time and space with respect to the sequence length M, training the model for a long sequence is so expensive, it causes the problem called "Attention Bottleneck" (Wang et al. 2020; Choromanski et al. 2021). The problem is more severe for us because we use convolutions and 3D tensors inside the attention mechanism (The attention map is of size $H \times W$ so the complexity of the attention will be $O(M^2 \times H \times W)$). Thus, our model will be prohibitively expensive to train. In the last few years, there were many propo-

Layer	Layer Parameters
Conv3d + BN3d	$11 \times 11 \times 3 \times 64, 64$
Conv3d + BN3d	$5 \times 5 \times 3 \times 128, 128$
Conv3d + BN3d	$3 \times 3 \times 3 \times 64, 64$
Conv3d + BN3d	$3 \times 3 \times 3 \times 64, 64$

Table 2. 3D CNN architecture where Conv3D is a 3D convolutional element and BN3d is a 3D batch normalization element.

sitions to solve this issue. Wang et al. (2020) demonstrated that a low-rank matrix could approximate the self-attention mechanism. They suggest a new self-attention method that minimizes total self-attention complexity. Choromanski et al. (2021) presented a novel Transformer architecture that uses linear space and time complexity to estimate regular (softmax) full-rank-attention Transformers with proven accuracy. However, all these propositions stay irrelevant to our case because we are not using the standard self-attention mechanism. The convolutions make it an arduous task. So the solution we preferred to go with is to reduce the length of the sequence before feeding it to the Transformer block. Reducing the sequence must be done without losing relevant information. So we propose to use a 3D convolution neural network (3D CNN). 3D CNN is an improved type version of CNN first proposed by Tran et al. (2014) where it applies a three-dimensional filter to the dataset and the filter moves in three directions to calculate the low-level feature representations. Their output shape is a three-dimensional volume space. We applied $3DCNN$ where we input the sequence **J** to get the reduced new sequence **S** following this equation:

$$S_n = \text{3DCNN}(J_{(n-1)*K+1}, .., J_{n*K}), \quad n \in \{1, .., N\}. \tag{3}$$

Let M be the length of the series J, we will feed K inputs of J to the 3DCNN to generate one entry S for our Transformer. So, in the end, the new sequence S will be $S \in \mathbb{R}^{N \times H' \times W' \times D}$ where $N = M/K$, D is the number of channels and H' and W' are the new height and width. By using the 3DCNN, we reduced the length of the sequence by a factor of K which also reduced the complexity of the model. The 3DCNN does not just reduce the length of the input sequence, it also captures local Spatio-temporal low-level features. The 3DCNN captures these particulate features due to its focus on the local characteristics (space and time) of the sequence, while the Transformer focuses on the global characteristics. On the whole, we have reduced the computation without losing essential information that is important for classification. Table 2 summarize the architecture used inside the 3DCNN.

3.3. Convolutional BERT

After getting the new output S of the 3DCNN, it is time to feed it to what we call Convolutional BERT which stands for Convolutional Bidirectional Encoder Representations from Transformers. Transformer and self-attention have become one of the main models that revolutionize deep learning in the last few years, especially in neural language processing (NLP). Self-attention (Bahdanau et al. 2014), also known as intra-attention, is an attention mechanism that connects different positions in a single sequence to compute a representation of the sequence. "Attention" refers to the fact that, in real life, when viewing a video or listening to a song, we frequently pay more attention to certain details while paying less attention to others based on the importance of the details. Deep Learning uses a similar flow for its attention mechanism, giving particular parts of the data more focus as it is processed. Our intention form using this mechanism is for the model to focus more on the changes happening in the image sequence to better discriminate between astronomical objects. Self-attention layers are the foundation of the Transformer block design. Transformers were first introduced by Vaswani et al. (2017) where they did present a model-based attention dispensing with recurrence and convolutions entirely. They inspired many others that used the concept of Transformers to achieve even better results. For example, in BERT (Devlin et al. 2019) the authors used only the encoder block by stacking many of them. Even though Transformers were widely used in NLP in the last two years, people started implementing these blocks in other domains like image classification. Dosovitskiy et al. (2021) presented a model free from convolutions by using only a Transformer to classify images. Garnot et al. (2019) also suggested that they are able to extract temporal characteristics using a custom neural architecture based on self-attention instead of recurrent networks. Their use was not limited to image classification; action recognition was also investigated like in Sharir et al. (2021) where the authors used a Transformer-based approach inspired by Dosovitskiy et al. (2021) work. Liu et al. (2021) did propose a new Transformer where they added convolution to the attention mechanisms making it able to apply convolutions while extracting the temporal features.

3.3.1. Positional encoding

Because Transformers have no recurrence throughout the thumbnail sequence, some information about each thumbnail's relative or absolute position must be injected into the feature map obtained by the 3DCNN in order to inform the model about the order in the sequence. Similar to the original Transformer paper (Vaswani et al. 2017), we use positional encoding at each layer in the encoder to achieve this. The only difference is that our positional encoding is a 3D tensor where $P \in \mathbb{R}^{N \times H' \times W' \times D}$. Because the positional encoding and the new feature maps have the same dimension, they can be added together. We use sine and cosine functions to encode the position (Vaswani et al. 2017):

$$P_{(n,2i)} = sin(n/10000^{2i/D}), \tag{4}$$

$$P_{(n,2i+1)} = cos(n/10000^{2i/D}), \tag{5}$$

where n denotes the position in the sequence of length N, i is the channel dimension. D represent the total number of channel gotten by the 3DCNN. The sinusoidal positional encoding is chosen to make it easy for the model to learn to attend to relative positions. To get the new input for the convolutional BERT we conduct element-wise addition between the positional encoding and the feature maps obtained from 3DCNN to obtain the new tensor $F \in \mathbb{R}^{N \times H' \times W' \times D}$:

$$F_n = S_n + P_n, \qquad n \in \{1, .., N\}. \tag{6}$$

In this study, we only used information about the position of the image in a sequence. While the observation date could be

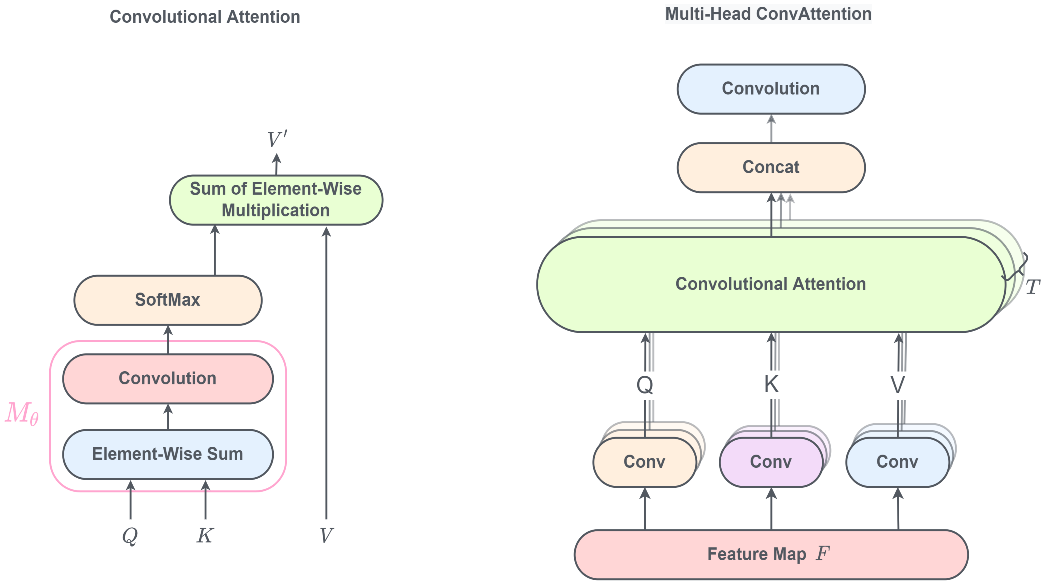

Fig. 5. (left) The convolutional attention. (right) The multi-head convolutional attention. To obtain the query, key and value maps we applied a convolution layer on the feature map obtained from 3DCNN

used as an alternative to the position, it would require adjusting the positional encoding function. Our experiments on the SDSS dataset did not reveal any improvement in the model when using the observation date as opposed to just using the position. This can be understood because we do the training and the test with the same observation sequence and the network can therefore learn this sequence. On the other hand, not incorporating any information regarding the order of the sequence greatly degraded the performance of the model. As a result, we ultimately chose to use only the position in our model, see section 4.2 for more discussion.

The newly obtained sequence F is fed to Multi-Head convolutional attention, which is an improved self-attention that has convolution. Then the Multi-Head convolutional attention is followed by the second component which is a tiny Feed-Forward Network (FFN) that has convolutions applied to every attention map. Its primary purpose is to transform the attention map into a form acceptable by the next convolutional BERT layer, the FFN consists of two convolutional layers with ReLU activation in between.

3.3.2. MultiHead Convolutional Self Attention

For the MultiHead Convolutional SelfAttention, we used the model proposed by Liu et al. (2021) with a few modifications where we replaced the last linear layer with a convolution layer. We believe that convolution in self-attention is better than the dot product between the query and the key because the convolution will accurately calculate the similarity, especially when we have 3D feature maps. A query map and a set of a pair key, value maps are encoded to an output using convolutional self-attention, where the query map, key maps, value maps, and output are all 3D tensors. Figure 5 represent the general architecture of the multi-head ConvAttention.

We use a convolution layer to generate the attention model's query, value, and key. The input to the attention model is $F \in \mathbb{R}^{N\times H'\times W'\times D}$. We pass each map through a convolution layer to get $\{Q, K, V\} \in \mathbb{R}^{N\times H'\times W'\times D'}$ where $D' = D/T$ and T represent the number of attention heads. Then we apply a subnetwork M_θ on the query and the key maps, which consists of an element-wise sum of the query and the key maps followed by another convolution layer to generate our attention map $H_{(n,m)} \in \mathbb{R}^{H'\times W'\times 1}$:

$$H_{(n,m)} = M_\theta(Q_n, K_m), \qquad n, m \in \{1, .., N\}. \tag{7}$$

After getting all the map attentions $H_n = \{H_{(n,1)}, H_{(n,2)}, ..., H_{(n,N)}\}$ where $H_n \in \mathbb{R}^{H'\times W'\times N}$, we apply a softmax operation along the third dimension of size N. Then we conduct an element-wise product between the attention map and the value map following this equation:

$$V'_n = \sum_{m=1}^{N} SoftMax(H_n)_{(n,m)} V_m. \tag{8}$$

We concatenate the new value representation V'_n obtained from the different attention heads. Multi-head attention is used to attend to input from various representation subspaces jointly:

$$MultiHead(Q, K, V) = Concat(V'_{n_1}, ..., V'_{n_T}). \tag{9}$$

Finally, we apply a convolution layer for merging the output of the Multi-head and obtaining a high-level representation that groups all the heads. At the end of the network, we pass the encoded sequence to 3D max-pooling and finally to the classifier to make a prediction.

3.4. *Evaluation metrics*

Accuracy is the probability that an object will be correctly classified. It is defined as the sum of the true positives plus true negatives divided by the total number of individuals tested.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

where TP, TN, FP, and FN are respectively True Positive, True Negative, False Positive, and False Negative.

The F1 score is a classification accuracy metric that combines precision and recall. F1 is a suitable measure of models tested with imbalanced datasets.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

4. Experiments

4.1. Implementation details

The supernovae in our data are not all spectroscopically confirmed which means that the ones not confirmed might contain some misclassified objects due to errors from the photometric typing. The model may not generalize due to this data bias. To ensure that our model will generalize only on spectroscopically confirmed data, we split up the training process into two steps. We have divided the data into two datasets. The first one contains only the photometrically typed data, and the second contains spectroscopically confirmed data. We trained the model at first with the photometrically typed data then we used transfer learning to fine-tune the model on only spectroscopically confirmed data, the table 5 summarizes the partition of the data. The models are trained using cross-validation of five folds and three ensembles in each fold. All the architectures presented in this paper follow this same process and are implemented using PyTorch (**?**).

We performed an extensive hyperparameter tuning of over 20 models to specify the best hyperparameters for our architecture that contains 1.3 Million parameters. We conducted hyperparameter optimization using only non-confirmed dataset with different parameters such as sequence length M, learning rate lr, 3DCCN sub-sequence length K, classifier layers size, number of ConvBERT layers L, number of Multi-head ConvAttention T, batch size, and dropout. We used Adam optimizer (Kingma & Ba 2017) with the value of learning rate 10^{-3}, we trained the model with cross-entropy loss and a dropout of 0.3. Hyperparameter tuning involves the number of images K that feed the 3DCNN and the maximum length of the sequence. The best values were $K = 3$ and $M = 99$ which means the number of sequences for the convolutional BERT is $N = 33$. The batch size was 128 sequences which we ran over 100 epochs. We chose the number of convolutional BERT layers to be $L = 2$ and the number of attention heads $T = 4$. Also, the images were normalized bandwise as each band has different characteristics. We used only 4 classes (AGN, SNIa, Variable, SNOther) to train all the models. The class unknown has not been considered in the study. It corresponds to noisy or very sparse data. It can easily be tagged from sparsity or noise in the image metrics and we do not expect any improvement in the classification if such objects are added to the training. We trained all models with 4 GPUs GeForce RTX 2080 Ti, Each model takes about 3 hours to complete training. The implementation will be released upon publication on the following link `https://github.com/DaBihy/ConvEntion`.

4.2. Results

This section provides studies on SDSS comparing the accuracy and F1 score of our proposed solution with other works. Table 3 summarises the result of **different models from different deep learning areas** to diversify our benchmark as it contains RNN architectures (SuperNNova, LSTM), CNN-based models like SCONE, Hybrid models that have CNN and RNN like (Carrasco-Davis et al. 2019) and (Gómez et al. 2020), and finally a Transformer-based model. Also, we compared the result using two types of datasets, first the image dataset and second the same dataset object but with the light curves, the goal is to highlight the advantage of using images instead of light curves. Moreover, the different works mentioned in the table 3 were initially proposed for different datasets with different classes and training protocols. Hence the results do not reflect the quality of these works on other datasets. The goal of the comparison is to give visibility into the performance of our model from a deep-learning standpoint, and the importance of using image time series from an astronomy perspective.

Overall our model ConvEntion obtains the highest accuracy of 79.83% and F1 score of 70.62%, 13 points higher in accuracy than the best results on images by (Gómez et al. 2020), and 12 points higher in accuracy than the best model using light curves. This confirms the advantage of using images over light curves. This advantage can be explained by the fact that the image contains more information than a single value of flux in a light curve. Hence a model can learn robustly with the existence of more high-level feature maps. Also, ConvEntion performed better compared to the other image-based models like Carrasco-Davis et al. (2019). Additionally, Transformers give a remarkable computational advantage because Transformers avoids recursion and allows parallel computation hence reducing the training time. Our model took only 3 hours to train compared to other image-based models which took 5 hours of training on our GPUs. Our model achieved better results using fewer parameters compared to the other models trained on image sequences. The main benefit of using a Transformer is that it reduces the drop in performance due to long dependencies. Transformers do not rely on past hidden states to capture dependencies with previous features like RNNs. They instead process a sequence as a whole. Therefore there is no risk to lose past information. Also, the integration of Spatio-temporal feature extraction helped in getting a better high-level representation of the sequence in comparison to separating the spatial features from the temporal ones. The two types of features have correlations that may help the model to better discriminate between objects. We can also highlight the importance of separating the band to mitigate the impact of missing observation. Our model performed well in comparison to Gómez et al. (2020) that uses multiple bands, which shows that separating the bands and adding band embedding works better than feeding the network with empty bands.

The authors of the paper Carrasco-Davis et al. (2019) trained their model on a dataset that has only a "g" band and they mentioned that the model can be adapted to classify the image sequence combining information using multiple bands. So for the sake of comparison, we trained the image models with all the bands "ugriz" at first and then with only one band "g". Our model achieved an accuracy so 76.89% and 63.20% in the F1 score using one band "g" which dropped 7% in comparison to using multiple bands. Meanwhile, Carrasco-Davis et al. (2019) achieved 63% in accuracy and 60% in F1 score. This shows that our model is more efficient when using multiple bands. This also

Model	Bands	Type of data	Accuracy	F1 Score	Num params
ConvEntion (Ours)	ugriz	Images	**79.83**	**70.62**	1.253M
CNN+GRU (Gómez et al. 2020)	ugriz	Images	66.39	63.22	1.993M
ConvEntion (Ours)	g	Images	76.89	63.20	1.253M
CNN+GRU (Gómez et al. 2020)	g	Images	63.67	61.00	1.992M
CNN+LSTM (Carrasco-Davis et al. 2019)	ugriz	Images	64.08	60.65	2.190M
CNN+LSTM (Carrasco-Davis et al. 2019)	g	Images	63.00	60.00	2.189M
SuperNNova (Bayes) (Möller & de Boissière 2020)	ugriz	Light curves	65.54	55.40	-
SITS-BERT (Yuan & Lin 2021)	ugriz	Light curves	67.43	51.60	0.596M
SCONE (CNN) (Qu et al. 2021)	ugriz	Light curves	62.57	50.43	22.2K
SuperNNova (RNN) (Möller & de Boissière 2020)	ugriz	Light curves	56.30	42.60	-
LSTM	ugriz	Light curves	55.24	40.33	60K

Table 3. Performance comparison in terms of average F1 Score and the average of the Accuracy of 5 folds of cross-validation

. This table includes only experiments on a dataset with 4 classes.

Model	Bands	Accuracy	F1 Score
ConvEntion (Ours)	ugriz	**83.90**	**75.77**
ConvEntion (Ours)	g	79.47	72.38
CNN+GRU (Gómez et al. 2020)	g	74.84	68.95
CNN+LSTM (Carrasco-Davis et al. 2019)	g	73.94	67.29

Table 4. Performance comparison in terms of average F1 Score and the average of the Accuracy of 5 folds of cross-validation. This table includes only experiments on a dataset with 3 classes.

highlights the impact of band separation to mitigate the impact of the missing observations.

Figure 6 illustrates the obtained confusion matrix by ConvEntion and it shows that the model has well classified the supernovas. Most of the misclassified SNIa are associated with SNOther and vice versa, which is not a serious error. This is even an expected behavior, especially since all types of supernovas share a lot of similarities which may confuse the model. Additionally, with a small dataset like ours, it is normal to have such behavior because the model does not have enough samples to totally discriminate between objects. Meanwhile, Variables were the best classified class in our dataset with a bit of confusion with the AGN, this misclassification between AGN and variable can be explained by the class imbalance in our dataset knowing that the number of variables is higher than the other classes.

The table 4 summarizes the results of different models trained only on 3 classes (AGN, SN, Variable) where classes SNIa and SNOther were combined into a single class. The goal of this experiment is to see the behavior of our model on discriminating between transient and non-transient objects. We got the best results with an accuracy of 83.90% with an F1 Score of 75.77%. The model was able to classify the SN accurately with a score of 86% as shown in figure 7

The model is able to effectively process a given survey without any loss in performance and without the requirement of providing it with the time information for each image. However,

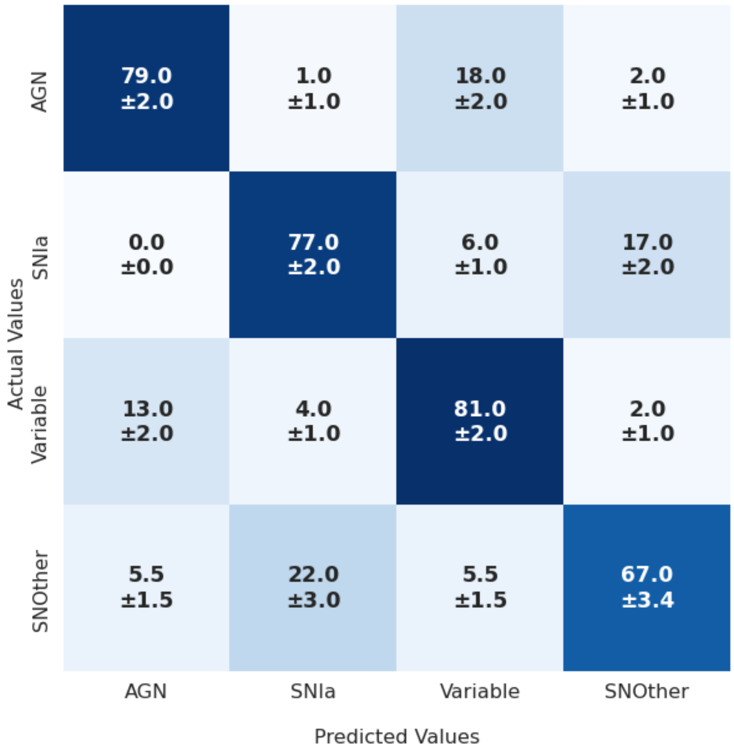

Fig. 6. Confusion matrix showing the average accuracy and standard deviation of the predictions generated by ConvEntion over cross-validation of five folds on test data.

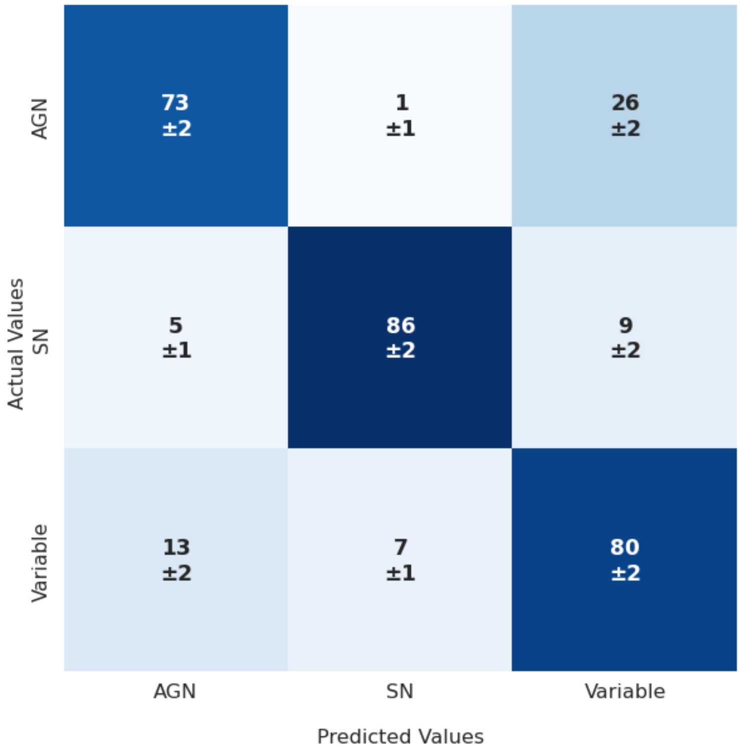

Fig. 7. Confusion matrix of 3 classes showing the average accuracy and standard deviation of the predictions generated by ConvEntion over cross-validation of five folds on test data.

when there is a covariate shift, or a mismatch, between the training set and the test set (such as using a different dataset with a different observation sequence), incorporating the time information can improve the results. This experimental finding will be further studied and reported in future work using other datasets.

5. Conclusion

In this work, we have presented a method for efficient astronomical image time series classification that is entirely based on the combination of convolutional networks and Transform-

Class	Train	FineTune	Test
AGN	362	362	182
SNIa	1448	400	99
Variable	1290	1290	645
SNOther	2041	72	17

Table 5. Count of every object in a dataset of each step in training protocol. Train contains only photometrically typed data, FineTune and Test contain only spectroscopically confirmed data

ers. Inspired by action recognition and satellite image time series classification we proposed a model ConvEntion that utilizes convolutions and Transformers jointly to capture complex spatio-temporal dependencies between distinct steps, leading to accurate predictions based on different observations of an object. The accuracy of our model is better with a high margin of 13% in comparison to state-of-the-art methods using image data and even better compared to approaches using light curves.

Our model achieved good results on the SDSS dataset while being faster thanks to using fewer parameters and the parallel computation, making it a good candidate for latency-sensitive applications like real-time thumbnail classifier of astronomical events. Meanwhile, our benchmark is a clear evidence of the importance of images in the domain of astronomy. Indeed, the images contain more information than the normal light curves, even if they present more difficulties. In the future, we plan to scale up ConvEntion using self-supervised learning and investigate whether the model can generalize even better. With the large amount of unlabeled data in astronomy, we believe that the next step to advance AITS classification is with creating self-supervised models.

Acknowledgements. This work has been carried out thanks to the support of the DEEPDIP ANR project (ANR-19-CE31-0023). This work makes use of Sloan Digital Sky Survey (SDSS) data. Funding for SDSS-III has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, and the U.S. Department of Energy Office of Science. The SDSS-III web site is `http://www.sdss3.org/`. SDSS-III is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS-III Collaboration including the University of Arizona, the Brazilian Participation Group, Brookhaven National Laboratory, Carnegie Mellon University, University of Florida, the French Participation Group, the German Participation Group, Harvard University, the Instituto de Astrofisica de Canarias, the Michigan State/Notre Dame/JINA Participation Group, Johns Hopkins University, Lawrence Berkeley National Laboratory, Max Planck Institute for Astrophysics, Max Planck Institute for Extraterrestrial Physics, New Mexico State University, New York University, Ohio State University, Pennsylvania State University, University of Portsmouth, Princeton University, the Spanish Participation Group, University of Tokyo, University of Utah, Vanderbilt University, University of Virginia, University of Washington, and Yale University.

References

Bahdanau, D., Cho, K., & Bengio, Y. 2014, Neural Machine Translation by Jointly Learning to Align and Translate

Bao, H., Dong, L., Piao, S., & Wei, F. 2022, BEiT: BERT Pre-Training of Image Transformers

Boone, K. 2019, The Astronomical Journal, 158, 257

Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., et al. 2019, Publications of the Astronomical Society of the Pacific, 131, 108006

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, Journal of Artificial Intelligence Research, 16, 321–357

Cho, K., van Merrienboer, B., Gulcehre, C., et al. 2014, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

Choromanski, K. M., Likhosherstov, V., Dohan, D., et al. 2021, in International Conference on Learning Representations

Dablain, D., Krawczyk, B., & Chawla, N. V. 2021, DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. 2019, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2021, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Drake, A. J., Djorgovski, S. G., Mahabal, A., et al. 2011, Proceedings of the International Astronomical Union, 7, 306

Frieman, J. A., Bassett, B., Becker, A., et al. 2007, The Astronomical Journal, 135, 338–347

Fukugita, M., Ichikawa, T., Gunn, J. E., et al. 1996, AJ, 111, 1748

Garnot, V. S. F., Landrieu, L., Giordano, S., & Chehata, N. 2019, Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-Attention

Gill, K., Asefa, T., Kaheil, Y., & McKee, M. 2007, Effect of missing data on performance of learning algorithms for hydrologic predictions: Implications to an imputation technique

Gómez, C., Neira, M., Hernández Hoyos, M., Arbeláez, P., & Forero-Romero, J. E. 2020, Monthly Notices of the Royal Astronomical Society, 499, 3130–3138

Hložek, R., Ponder, K. A., Malz, A. I., et al. 2020, Results of the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC)

Hochreiter, S. & Schmidhuber, J. 1997, Neural Computation, 9, 1735

Holtzman, J. A., Marriner, J., Kessler, R., et al. 2008, The Astronomical Journal, 136, 2306–2320

Hossain, M. S., Betts, J. M., & Paplinski, A. P. 2021, Neurocomputing, 462, 69

Ivezić, Ž., Kahn, S. M., Tyson, J., et al. 2019, Astrophys.J., 873, 111

Ji, S., Xu, W., Yang, M., & Yu, K. 2013, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, 221

Kingma, D. P. & Ba, J. 2017, Adam: A Method for Stochastic Optimization

Lin, T.-Y., Goyal, P., Girshick, R. B., He, K., & Dollár, P. 2017, CoRR, abs/1708.02002

Liu, Z., Luo, S., Li, W., et al. 2021, ConvTransformer: A Convolutional Transformer Network for Video Frame Synthesis

Liu, Z., Ning, J., Cao, Y., et al. 2022, Video Swin Transformer

Möller, A. & de Boissière, T. 2020, Monthly Notices of the Royal Astronomical Society, 491, 4277–4293

Pasquet, J., Pasquet, J., Chaumont, M., & Fouchez, D. 2019, Astronomy & Astrophysics, 627, A21

PLAsTiCC-team, au2, T. A. J., Bahmanyar, A., et al. 2018, The Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC): Data set

Qu, H., Sako, M., Möller, A., & Doux, C. 2021, The Astronomical Journal, 162, 67

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1985, Learning internal representations by error propagation, Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science

Sako, M., Bassett, B., Becker, A. C., et al. 2014, Publications of the Astronomical Society of the Pacific, 130, 064002

Sharir, G., Noy, A., & Zelnik-Manor, L. 2021, An Image is Worth 16x16 Words, What is a Video Worth?

Shi, X., Chen, Z., Wang, H., et al. 2015, Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. 2014, Learning Spatiotemporal Features with 3D Convolutional Networks

Turkoglu, M. O., D'Aronco, S., Perich, G., et al. 2021, Crop mapping from image time series: deep learning with multi-scale label hierarchies

Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, Attention is All you Need

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. 2020, Linformer: Self-Attention with Linear Complexity

Yan, S., Xiong, X., Arnab, A., et al. 2022, Multiview Transformers for Video Recognition

Yuan, Y. & Lin, L. 2021, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 474

Zhou, J., Wei, C., Wang, H., et al. 2022, iBOT: Image BERT Pre-Training with Online Tokenizer