Draft version August 11, 2020
Typeset using LaTeX **twocolumn** style in AASTeX63

Alert Classification for the ALeRCE Broker System: The Real-time Stamp Classifier

R. Carrasco-Davis [1,2,*] E. Reyes [2,1,*] C. Valenzuela [3,1] F. Förster [3,1,4] P. A. Estévez [2,1,3] G. Pignata [5,1] F. E. Bauer [6,1,7] I. Reyes [1,3,2] P. Sánchez-Sáez [1,6,8] G. Cabrera-Vives [9,1] S. Eyheramendy [8,1] M. Catelan [6,1] J. Arredondo,[1] E. Castillo-Navarrete,[3,1] D. Rodríguez-Mancini,[1,9] D. Ruz-Mieres [3,1,10] A. Moya [1,3] L. Sabatini-Gacitúa,[1,3] C. Sepúlveda-Cobo,[1,3] A. A. Mahabal,[11,12] J. Silva-Farfn,[4] E. Camacho-Iiguez,[6] and L. Galbany[13]

[1] *Millennium Institute of Astrophysics (MAS), Nuncio Monseñor Sótero Sanz 100, Providencia, Santiago, Chile*
[2] *Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, Santiago 8320000, Chile*
[3] *Center for Mathematical Modeling, Universidad de Chile, Beauchef 851, North building, 7th floor, Santiago 8320000, Chile*
[4] *Departamento de Astronomía, Universidad de Chile, Casilla 36D, Santiago, Chile*
[5] *Department of Physical Science, Universidad Andres Bello, Av. Republica 230, Santiago 8370146, Chile*
[6] *Instituto de Astrofísica and Centro de Astroingeniería, Facultad de Física, Pontificia Universidad Católica de Chile, Casilla 306, Santiago 22, Chile*
[7] *Space Science Institute, 4750 Walnut Street, Suite 205, Boulder, Colorado 80301, USA*
[8] *Faculty of Engineering and Sciences, Universidad Adolfo Ibañez, Diagonal Las Torres 2700, Peñalolén, Santiago, Chile*
[9] *Department of Computer Science, Universidad de Concepcin, Edmundo Larenas 219, Concepcin, Chile*
[10] *Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK*
[11] *Cahill Center for Astrophysics, California Institute of Technology, 1200 E. California Boulevard, Pasadena, CA 91125, USA*
[12] *Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA 91125, USA*
[13] *Departamento de Física Teórica y del Cosmos, Universidad de Granada, E-18071 Granada, Spain*

ABSTRACT

We present a real-time stamp classifier of astronomical events for the ALeRCE (Automatic Learning for the Rapid Classification of Events) broker. The classifier is based on a convolutional neural network with an architecture designed to exploit rotational invariance of the images, and trained on alerts ingested from the Zwicky Transient Facility (ZTF). Using only the *science, reference* and *difference* images of the first detection as inputs, along with the metadata of the alert as features, the classifier is able to correctly classify alerts from active galactic nuclei, supernovae (SNe), variable stars, asteroids and bogus classes, with high accuracy (∼94%) in a balanced test set. In order to find and analyze SN candidates selected by our classifier from the ZTF alert stream, we designed and deployed a visualization tool called SN Hunter, where relevant information about each possible SN is displayed for the experts to choose among candidates to report to the Transient Name Server database. We have reported 3060 SN candidates to date (9.2 candidates per day on average), of which 394 have been confirmed spectroscopically. Our ability to report objects using only a single detection means that 92% of the reported SNe occurred within one day after the first detection. ALeRCE has only reported candidates not otherwise detected or selected by other groups, therefore adding new early transients to the bulk of objects available for early follow-up. Our work represents an important milestone toward rapid alert classifications with the next generation of large etendue telescopes, such as the Vera C. Rubin Observatory's Legacy Survey of Space and Time.

Keywords: Supernovae — Alert Broker Visualization Tools — Deep Learning

1. INTRODUCTION

The amount of data generated by modern survey telescopes cannot be directly handled by humans. Therefore, automatic data analysis methods are necessary to fully exploit their scientific return. A particularly challenging problem is the real-time classification of tran-

Corresponding author: Rodrigo Carrasco-Davis
rodrigo.carrasco.davis@gmail.com
rodrigo.carrasco.d@ing.uchile.cl

* These authors contributed equally to this work

sient events. Nevertheless, the possibility to generate a quick probabilistic evaluation of which type of transient has been discovered is crucial to perform the most suitable follow-up observation, and by extension obtain the best constraints on its physics. In this work we focus on the early detection of supernovae (SNe) by quickly discerning between SNe and various other confusing classes of astronomical objects. Photometric and spectroscopic observations carried out soon after the explosion are fundamental to put constraints on the progenitor systems and explosion physics.

In the case of thermonuclear explosions (Type Ia SNe), early observations probe the outermost part of the ejecta, where it is possible to detect the material present at the surface of the progenitor (e.g., Nugent et al. 2011), evaluate the degree of mixing induced by different explosion models (e.g., Piro & Morozova 2016; Jiang et al. 2017; Noebauer et al. 2017), and estimate the size of the companion star (e.g., Kasen 2010).

For core collapse (CC) SNe, observations carried out soon after the explosion allow to constrain the radius of the progenitor star, its outer structure and the degree of ^{56}Ni mixing (e.g., Tominaga et al. 2011; Piro & Nakar 2013), but also the immediate SN environment, providing a critical diagnostic for the elusive final evolutionary history of the progenitor and/or the progenitor system configuration (e.g., Moriya et al. 2011; Gal-Yam et al. 2014; Groh 2014; Khazov et al. 2016; Tanaka et al. 2016; Yaron et al. 2017; Morozova et al. 2017; Frster et al. 2018).

We propose a method to quickly classify alerts among five different classes, four of which are astrophysical, to then use the predictions to find and report SNe. This work has been developed in the framework of ALeRCE[1] (Automatic Learning for the Rapid Classification of Events; Frster et al. 2020). The ALeRCE system is able to read, annotate, classify and redistribute the data from large survey telescopes. Such efforts are commonly called *Astronomical Broker Systems* (other examples include, e.g., ANTARES, Narayan et al. 2018; Lasair, Smith et al. 2019). Currently, ALeRCE is processing the alert stream generated by the Zwicky Transient Facility (ZTF; Bellm et al. 2018) and its main goal is to reliably classify data of non-moving objects, and make these classifications available to the scientific community.

For the purpose of classifying astronomical objects or transients, one way to discriminate among them is by computing features from the light curve of each object (e.g., Richards et al. 2011; Pichara et al. 2016; Martnez-Palomera et al. 2018; Boone 2019; SnchezSez et al. 2020), using the light curve directly as input to a classifier (e.g., Mahabal et al. 2017; Naul et al. 2018; Muthukrishna et al. 2019; Becker et al. 2020). In the case of an alert stream scenario such as for ZTF (whereby no forced photometry of past images is as yet provided), the light curve is built by estimating the flux from the difference image for all alerts triggered at the same coordinates.

Our model is dubbed the *"stamp classifier"*, since it only uses the first alert of an astronomical object. ALeRCE also developed a *light curve classifier* (SnchezSez et al. 2020) based on light curves with ≥ 6 detections in g or ≥ 6 detections in r ZTF bands. The *light curve classifier* is able to discriminate among a richer taxonomy of astronomical objects. Both the stamp and light curve classifiers are currently running through the ALeRCE frontend (Frster et al. 2020).

Our proposed *stamp classifier* is based on a convolutional neural network (CNN) architecture that uses only the information available in the first alert of an astronomical object, which includes the images of the objects plus metadata regarding some of the object properties, observation conditions and information from other catalogs. The stamp classifier uses the first alert to discriminate between active galactic nuclei (AGN), supernovae (SNe), variable stars (VS), asteroids and bogus alerts. The architecture was designed to exploit the rotational invariance of astronomical images. The classifier was trained using an entropy regularizer that avoids the assignment of high probability to a single class, yielding softer output probabilities that give extra information to experts, useful for further analysis of candidates. To the best of our knowledge, this is the first classifier that discriminates among five classes using a single alert, allowing a rapid, reliable characterization of the data stream to trigger immediate follow-up. Previous work on stamp classification has focused instead on the classification of real objects vs. bogus detections (e.g., Goldstein et al. 2015; Cabrera-Vives et al. 2017; Reyes et al. 2018; Duev et al. 2019; Turpin et al. 2020), galaxy morphologies (e.g., Dieleman et al. 2015; Prez-Carrasco et al. 2019; Barchi et al. 2020), or time domain classification (Carrasco-Davis et al. 2019; Gmez et al. 2020).

An associated contribution to the stamp classifier is the implementation of a visualization tool called *SN Hunter*[2], which allows experts to explore SN candidates to further filter alerts, and choose objects to request follow-up. This visualization tool is deployed online and

[1] https://alerce.online/

[2] https://snhunter.alerce.online/

provides a snapshot of the current ZTF data stream within minutes of receiving new alerts.

This work is structured as follows: In Section 2 we describe the data used to train the proposed neural network model, as well as a brief description of each class and how we gathered labeled data. In Section 3 we describe the data pre-processing, the neural network architecture, the entropy regularizer added to the optimization function and the experiments run to find the best architecture for the problem at hand. In Section 4 we report and discuss our results in terms of the classification task, and also in terms of SNe detection. In Section 5, we describe the SN Hunter visualization tool and the visual criteria used by human experts to choose good candidates to report to the Transient Name Server (TNS)[3], along with an analysis of reported and confirmed SNe by ALeRCE using the proposed methodology since June 2019. We finally draw our conclusions and describe future work in Section 6.

2. DATA

An alert within the ZTF stream is defined as a source in the sky that produces a signal five standard deviations higher than the background noise (a five-σ magnitude limit; Masci et al. 2018), and which passes a real bogus filter designed by the ZTF collaboration (Mahabal et al. 2019). When an alert is triggered, an *alert packet* is generated with all the relevant information about the source that triggered the alert (Bellm et al. 2018). The alert packet contains three images called stamps, which are cropped at 63 pixels on a side (1 pixel = 1 arcsec) from the original image and centered on the position of the source. In addition, the alert packet contains metadata related to the source, the observation conditions of the exposure and other useful information (Masci et al. 2018). An example of the three stamps within an alert packet is shown in Figure 1. The stamp in Figure 1a is called the *science image* and corresponds to the most recent measurement of the source. The stamp depicted in Figure 1b is the *reference image*, which is fixed for a given region and bandpass. It is usually based on images taken at the beginning of the survey and it is built by averaging multiple images to improve its signal-to-noise ratio. The stamp shown in Figure 1c is the *difference image*, between the science and reference images (Masci et al. 2018), which shows the change in flux between those frames, removing other sources with constant brightness.

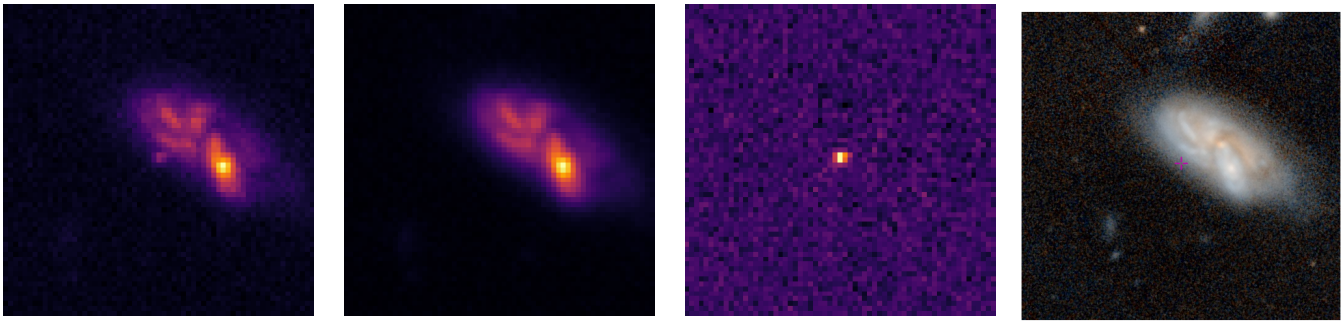
[3] https://wis-tns.weizmann.ac.il/

Each alert packet represents only 2 samples in time, the reference and science image exposures, and often is insufficient to correctly classify objects over the full taxonomy of different variable stars, transients or stochastic sources as in SnchezSez et al. (2020).

However, our hypothesis is that it is feasible to use the information included in a single alert packet to separate objects into several broad classes, namely AGN, SNe, VS, asteroids and bogus alerts. Each class presents distinctive characteristics within the image triplet of the first detection alert (see Figure 2), which could be automatically learned by a CNN. In addition to the images, information in the metadata in the alert packet along with some derived features from the metadata are important to discriminate among the mentioned classes. The metadata used for the classification task is listed in Table 1, and the distribution of values for each feature per class is shown in Figure A1 in Appendix A. Some of the distinctive characteristics for each class are the following:

- **AGN**: Being stochastically variable objects, an alert generated by an AGN should have flux from the source in both the *reference* and *science* stamps. Considering this feature alone, it is difficult to discriminate AGNs from other variable sources. Nevertheless, AGNs should lie at the centers of their host galaxies (based on dynamical friction arguments), or appear as (quasi-)stellar objects, in relatively lower stellar density fields. Thus, a change in flux will appear as a variable source, which may lie at the center of a galaxy, or even when the galaxy is not visible they tend to be in lower stellar density fields. In these cases, the alert is likely to be triggered by an AGN. In addition, AGNs are commonly found outside the Galactic plane, as shown in Appendix A. Other important features are `sgscores`, which tend to have values closer to 0, since AGNs occur in extragalactic sources, and `distpsnr`, which should have low `distpsnr1` values since the nearest source should be the AGN itself, combined with large `distpsnr2` and `distpsnr3` values due to the lower source density outside of the Galactic plane. The `classtar` is also useful as more weakly accreting AGN candidates tend to be classified as galaxy-shaped sources by the SExtractor classifier (Bertin & Arnouts 1996).

- **Supernovae (SNe)**: An alert generated by a SN should appear as a change in flux where no unresolved sources were present. These transients tend to appear near their host galaxies, and their location should be consistent with the underlying host stellar population distribution (e.g., a SN will have a higher probability



(a) Science (b) Reference (c) Difference (d) Colored Image

Figure 1. Example g-band images from a ZTF alert packet, in this case from a type Ia SN (ZTF19abmolyr) classified by our method. (a) The science image is the latest measurement of a source. (b) The reference image is usually a higher signal-to-noise measurement taken from an earlier epoch. (c) The third stamp is the *difference* between the reference and science images. (d) For context, we also show the *gri* color image from PanSTARRS, which is not part of the alert packet nor used in the current stamp classifier. Each image stamp is 63 pixels×63 pixels, where 1 pixel = 1 arcsec.

of arising from a location aligned with the disk than perpendicular to it). As such, most SN detections exhibit a visible host galaxy in both the *science* and *reference* stamps, with the flux from the SN arising only in the *science* and *difference* images. SN candidates tend to appear outside the Galactic plane, and so the `sgscores`, `distpsnr`, and Galactic latitude features have similar distributions to AGN candidates. However, there are other features that might help to classify SN candidates correctly. For instance, `chinr` and `sharpnr` present a different distribution for the SN class, compared to the other classes (see Appendix A). Furthermore, `isdiffpos` value should always be 1 for new SN candidates.

- **Variable Stars (VS)**: The flux coming from variable stars usually appears in both the *reference* and *science* stamps. With ZTF's sensitivity, variable stars can be detected within the Milky Way or the Local Group, and thus the alert will typically not be associated with a visible host galaxy in the stamp, but rather with other point-like sources. In addition, such alerts will have a higher probability of residing at lower Galactic latitudes and in crowded fields with multiple point sources within the stamps, given the high concentration of stars in the disk and bulge of our Galaxy. Therefore, VS candidates present a distribution of higher `sgscores`, lower `distpsnr` and Galactic latitude closer to 0 compared to AGN and SN candidates (see Figure A1).

- **Asteroids**: Alerts from moving Solar-system objects will appear only one time at a given position, and thus will show flux only in the *science* and *difference* images. Depending on their distance and speed, they may appear elongated in the direction of motion. In addition, such alerts should have a higher probability of residing at lower ecliptic latitudes as shown in Figure A1. Also, new asteroid candidates should always have an `isdiffpos` feature equal to 1.

- **Bogus alerts**: Camera and telescope optics effects, such as saturated pixels at the centers of bright sources, bad columns, hot pixels, astrometric misalignment in the subtraction to compute the *difference* image, unbaffled internal reflections, etc., can produce bogus alerts with no interesting real source. Bogus alerts are characterized by the presence of NaN pixels due to saturation, single or multiple bright pixels with little or no spatial extension (i.e., smaller than the telescope point spread function (PSF) and nightly seeing), or lines with high or low pixel values that extend over a large portion of the stamp (hot or cold columns/rows). We are currently working to include satellites in this class. However, they may share some image traits with asteroids, but are not confined to the ecliptic.

We built a training set of ZTF alerts using the labeled set from SnchezSez et al. (2020), which is a results of cross-matching with other catalogs, such as the ASAS-SN catalogue of variable stars (Jayasinghe et al. 2018, 2019a,b, 2020), the Roma-BZCAT Multi-Frequency Catalog of Blazars (Massaro et al. 2015), the Million Quasars Catalog (version June 2019, Flesch 2015, 2019), the New Catalog of Type 1 AGNs (Oh2015; Oh et al. 2015), the Catalina Surveys Variable Star Catalogs (Drake et al. 2014, 2017), the LINEAR catalog of periodic light curves (Palaversa et al. 2013), *Gaia* Data Release 2 (Mowlavi et al. 2018; Rimoldini et al. 2019), the SIMBAD database (Wenger et al. 2000), and spectroscopically classified SNe from the TNS database. The

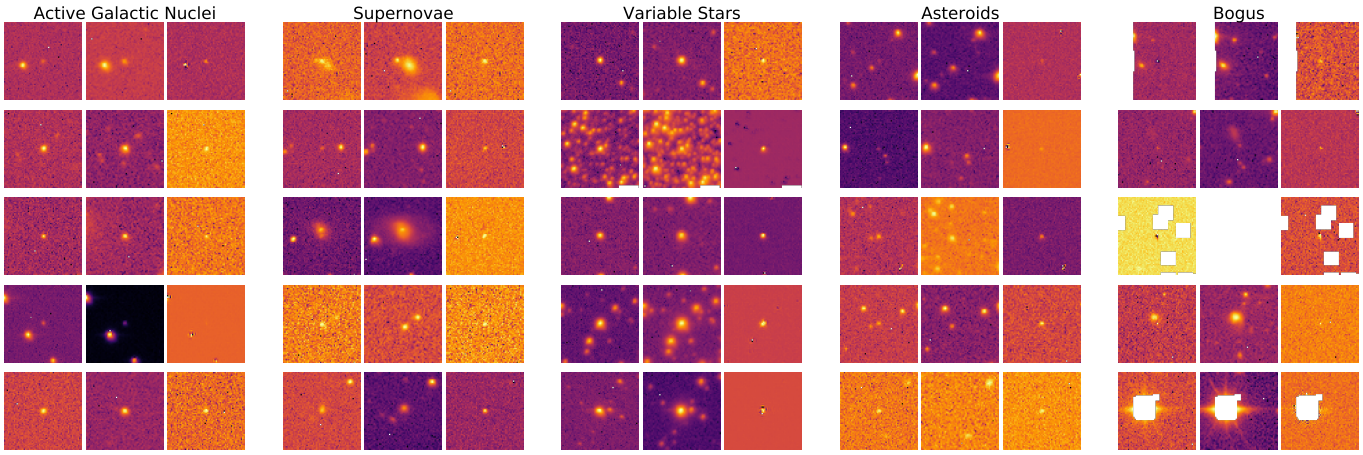


Figure 2. Examples of the five classes that are to be discriminated by using only the first detection. For each class, the triplet of images in each row are *science, reference* and *difference* images from left to right. Each row corresponds to a different candidate.

asteroid subset was built by selecting the alerts that were near a Solar-system object, requiring that the `ssdistnr` field in the alert metadata exists. Each sample corresponds to the triplet of *science, reference*, and *difference* images of the first detection. The number of samples of AGN, SN, VS, asteroid, and bogus are 14,966 (29%), 1620 (3%), 14,996 (29%), 9899 (19%), and 10,763 (20%), respectively, with a total of 52,244 examples (undersampling the labeled set from SnchezSez et al. (2020) for for better balance between classes since 3% SNe would not exactly be considered balanced compared to the rest). The bogus class was built in two steps: we first used 1980 bogus examples reported by ZTF (based on human inspection) and ran an initial iteration of the proposed classifier detailed in Section 3.2. Then, another 8783 bogus samples were labeled by our team of experts using the SN Hunter and added to the training set by manually inspecting the samples predicted by an early version of the model as SNe. The main aim of the stamp classifier is the fast detection of SNe, therefore the training set consists only of first alerts, which allows us to estimate probabilities of objects as soon as we receive the alert.

3. METHODOLOGY

3.1. *Data Pre-Processing*

The standard shape for each stamp within an alert is 63×63 pixels; 650 non-square shaped stamps were removed from the dataset. After removing misshaped stamps, we obtained 14742 (29%) AGN, 1596 (3%) SN, 14723 (29%) VS, 9799 (19%) asteroids and 10734 (20%) bogus alerts, with a total of 51,594 examples. Some pixels have NaN values due to pixel saturation, bad columns or stamps from the edges of the camera; all NaN pixels were replaced by a value of 0, giving information about NaNs content within the stamp to the classifier. Preliminary tests showed that smaller images for training lead to better results, therefore we cropped all the stamps at the center getting 21×21 pixels images; this size was selected by the hyperparameter random search discussed in Section 3.5. Better results with a small stamp size may be explained by the fact that smaller stamps means a dimensionality reduction with respect to the original image size in the input of the CNN, and this may be easier to handle by the model. Further analysis of the optimal stamp size for the classification task at hand must be carried out since it might be important for the design of future alert stream based surveys. Each stamp was normalized independently between 0 and 1 by subtracting the minimum pixel value in the image and dividing by the maximum of the result. Finally, a 3-channel cube is assembled as input to the classifier, built by stacking the resulting *science, reference* and *difference* images as separate channels, resulting in a $21 \times 21 \times 3$ image. The metadata are clipped differently for each feature following the values in Table A1, then each feature is normalized by subtracting the mean value of the training set and dividing by the standard deviation.

3.2. *Classifier Architecture*

The classification model is a CNN based on the real-bogus classifier proposed by Reyes et al. (2018), which is an improvement over Deep-HiTS (Cabrera-Vives et al. 2017) by adding rotational invariance to the CNN and analyzing the predictions of the model using Layer-wise Relevance Propagation (LRP; Bach et al. 2015). The specific CNN architecture used in this work is shown in Table 2. In these previous works, metadata were not included for classification.

Table 1. List of metadata of the alert used as features by the classifier. The definitions are from the ZTF avro schemas[a].

Feature	Description [units]
`sgscore{1, 2, 3}`	Star/Galaxy score of the {first, second, third} closest source from PanSTARRS1 catalog 0 ≤ `sgscore` ≤ 1 where a value closer to 1 implies higher likelihood of being a star, -999 when there is no source.
`distpsnr {1, 2, 3}`	Distance of the {first, second, third} closest source from PanSTARRS1 catalog, if one exists within 30 arcsec, -999 if there is no source [arcsec].
`isdiffpos`	t (converted to 1) if the candidate is from positive (science minus reference) subtraction; f (converted to 0) if the candidate is from negative (reference minus science) subtraction.
`fwhm`	Full Width Half Max assuming a Gaussian core of the alert candidate in the science image from SExtractor (Bertin & Arnouts 1996) [pixels].
`magpsf`	magnitude from PSF-fit photometry of the alert candidate in the difference image. [mag].
`sigmapsf`	1-sigma uncertainty in `magpsf` [mag].
`ra`, `dec`	Right ascension and declination of candidate; J2000 [deg].
`diffmaglim`	5-sigma mag limit in difference image based on PSF-fit photometry [mag].
`classtar`	Star/Galaxy classification score of the alert candidate in the science image, from SExtractor.
`ndethist`	Number of spatially-coincident detections falling within 1.5 arcsec going back to beginning of survey; only detections that fell on the same field and readout-channel ID where the input candidate was observed are counted. All raw detections down to a photometric Signal/Noise≈3 are included.
`ncovhist`	Number of times input candidate position fell on any field and readout-channel going back to beginning of survey.
`chinr`, `sharpnr`	DAOPhot (Stetson 1987) chi, sharp parameters of nearest source in reference image PSF-catalog within 30 arcsec.
Ecliptic coordinates	ecliptic latitude and longitude computed from the `ra`, `dec` coordinates of the candidate [deg].
Galactic coordinates	Galactic latitude and longitude computed from the `ra`, `dec` coordinates of the candidate [deg].
approx non-detections	`ncovhist` minus `ndethist`. Approximate number of observation in the position of the candidate, with a signal lower than Signal/Noise≈3.

[a] https://zwickytransientfacility.github.io/ztf-avro-alert/

The input of the neural network has a shape of $21 \times 21 \times 3$ as explained in Section 3.1. Following the architecture of Reyes et al. (2018), a zero padding is applied to the input, to then augment the batch with rotated versions of itself as described in Section 3.3. For the convolutional layers, the parameters shown in Table 2 are the filter dimensions and number of output channels. All convolutional layers, except for the first one, have zero padding (filling the edges of the images with zeros) that preserves the input shape after the convolution. Moroever, all the convolutional layers and fully connected layers have a Recified Linear Unit (ReLU; Nair & Hinton 2010) activation function (except for the last fully connected one that has a softmax output). The output of the last max-pooling layer, which reduces the dimensionality of the image by selecting the most intense values of non-overlapping windows of 2×2 pixels, is re-arranged (flattened) to a single dimension array for each sample in the batch, to feed the fully connected layers. Then, the rotation concatenation step takes place, stacking the fully connected output representation of the rotated versions of each sample, and passing through the cyclic pooling layer, where an average is applied in the stacked dimension. The metadata features are first processed by a batch normalization layer that learns an optimal bias and scale to normalize the data. The normalized features are concatenated to the output of the cyclic pooling. The concatenated representation passes through two fully connected layers. Finally, a softmax function is applied to the output of the last fully connected layer to obtain the estimated probabilities for each of the five classes. A glossary about CNNs and its training is presented in Appendix B.

3.3. *Rotational Invariance*

Astronomical objects present within a stamp usually have a random orientation. It has been shown that imposing rotational invariance to a classifier improves its accuracy for some classification problems (e.g., Dieleman et al. 2015, 2016; Cabrera-Vives et al. 2017; Reyes et al. 2018). In this work, rotational invariance is achieved by feeding the neural network with 90°, 180° and 270° rotated versions of the original input batch x. Defining r as a 90° rotation operation, then the samples within the extended batch will be $B(x) = [x, rx, r^2x, r^3x]$ after applying the rotations. At the last step of the architecture before the softmax output layer, a cyclic pooling operation is performed, which is basically an average pooling over the representation of the dense layer for each rotated example. A scheme of the procedure described in this section is shown in Figure 3.

3.4. *Entropy Regularization*

When the CNN model is trained using cross-entropy as the loss function to be minimized, the classification confidence of the model is very high, resulting in a distribution of output probabilities with saturated values of 0s and 1s without populating the values in between, even for wrong classifications. In this case there is no insight of certainty (relative probabilities between classes)

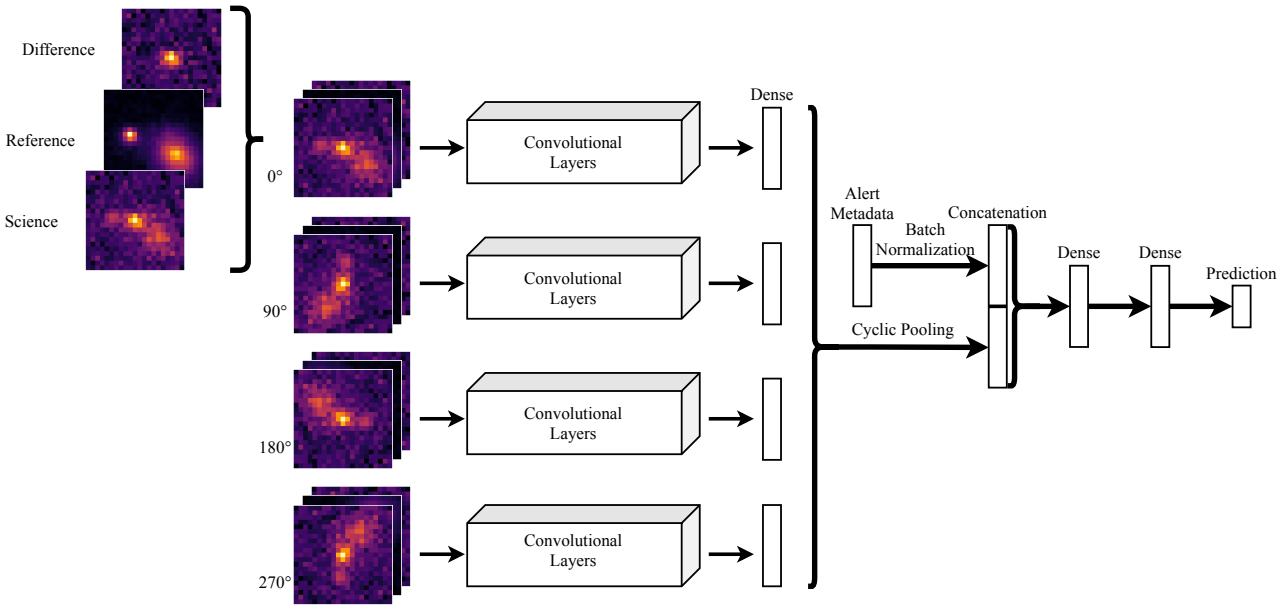


Figure 3. CNN enhanced with rotational invariance. The box *Convolutional Layers* refers to those described in Table 2, from the first convolutional layer to the last pooling layer. For each sample, the *science, reference* and *difference* images are concatenated in the channel dimension, obtaining an image input of dimension $21 \times 21 \times 3$. For each sample within the sampled batch, rotated versions are generated as described in Section 3.2 and fed to the CNN. After the first dense layer, the Cyclic pooling is performed. The metadata features pass through a batch normalization layer, the output of which is concatenated with the cyclic pooling output. Then, the concatenation goes through 2 fully connected layers, and finally a softmax function is applied to estimate the output probabilities.

of the prediction because most estimated probabilities for each class were either 0 or 1. In order to provide more granularity to the astronomers who revise SN candidates reported by the model to later request follow-up observing time, we added the entropy of the predicted probabilities of the models as a regularization term, to be maximized during training (Pereyra et al. 2017). By maximizing the entropy of the output probabilities, we penalize predictions with high confidence, in order to get better insight in cases where the stamps seem equally likely to belong to more than one class. The loss function $\mathcal{L}$ per sample is as follows:

$$\mathcal{L} = \underbrace{-\sum_{c=1}^{N} y_c \log\left(\hat{y}_c\right)}_{\text{cross-entropy}} + \underbrace{\beta \sum_{c=1}^{N} \hat{y}_c \log\left(\hat{y}_c\right)}_{\text{entropy regularization}} , \qquad (1)$$

where N is the number of classes, y_c is the one-hot encoding label (a value of 1 in the corresponding index of class, and 0 for the rest) indexed by c, $\hat{y}_c$ is the model prediction for class c, and β controls the regularization term in the loss function. Further explanation on the role of the loss function in the training process of a neural network is explained in Appendix B.

3.5. *Experiments*

A hyperparameter search was done by randomly sampling 133 combinations of the parameters shown in Table 3. For each combination of hyperparameters, we trained 5 networks with different initial random weights. The initial maximum number of iterations (presenting a single batch per iteration) was 30,000, evaluating the loss in the validation set every 10 iterations to save the best model thus far. After the first 20,000 iterations, if a lower loss is found on the validation set, 10,000 more iterations are performed. The validation and testing subsets were sampled randomly only once, taking 100 samples per class from the whole dataset, obtaining 500 samples for each of the mentioned subsets. The remaining samples were used in the training set. For each training iteration, the batch was built to contain roughly the same number of samples per class. We used Adam (Kingma & Ba 2017) as the updating rule for the network parameters during training, with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. Further details on the updating rules of a neural network an the Adam optimizer are described in Appendix B.

4. RESULTS

In this section, we first describe our results in terms of the classification task for the five classes. Then, we change our focus to the detection of SN candidates, since our main interest in this work is to discover extremely young transient candidates to be observed with follow-up resources. Further applications of this early classifi-

Table 2. Convolutional neural network architecture.

Layer	Layer Parameters	Output Size
Input	-	$21 \times 21 \times 3$
Zero padding	-	$27 \times 27 \times 3$
Rotation augmentation	-	$27 \times 27 \times 3$
Convolution	$4 \times 4, 32$	$24 \times 24 \times 32$
Convolution	$3 \times 3, 32$	$24 \times 24 \times 32$
Max-pooling	2×2, stride 2	$12 \times 12 \times 32$
Convolution	$3 \times 3, 64$	$12 \times 12 \times 64$
Convolution	$3 \times 3, 64$	$12 \times 12 \times 64$
Convolution	$3 \times 3, 64$	$12 \times 12 \times 64$
Max-pooling	2×2, stride 2,	$6 \times 6 \times 64$
Flatten	-	2304
Fully connected	2304×64	64
Rotation concatenation	-	4×64
Cyclic pooling	-	64
Concat with BN[a] features	-	$64 + 26$
Fully connected with dropout	90×64	64
Fully connected	64×64	64
Output softmax	64×5	5 (n° classes)

[a] BN stands for batch normalization

Table 3. Hyperparameter random search values.

Hyperparameter	Random Search Values
Learning rate	$5e$-3, $1e$-3, $5e$-4, $1e$-4, $5e$-5
Regularization parameter (β)	$0, 0.3, 0.5, 0.8, 1.0$
Batch size	$16, 32, 64$
Image size	$21, 41, 63$
Dropout rate	$0.2, 0.5, 0.8$
CNN kernel size	$3, 5, 7$

cation system might include rapid detection of extreme variability in AGN or tracking solar system objects.

The following results correspond to the best model in the search for hyperparameters, which adopts a batch size of 64 samples, learning rate of $1e$-3, dropout rate of 0.5, CNN kernel size of 5, image size of 21×21 pixels, and regularization parameter of $\beta = 0.5$. Appendix C contains the details on how this model was selected. We use accuracy[4] to compare models since the validation and test sets are balanced; achieving 0.95 ± 0.005 in the validation set and 0.941 ± 0.004 in the test set. Figure 4 shows the confusion matrix for the test set consisting of using five realizations of the mentioned model. With our five class model, we recover $87 \pm 1\%$ of the SNe, with only $5 \pm 2\%$ of false positives. For completeness, we also report the confusion matrix of the stamp classifier when no metadata features are included in the fully connected layers (see Figure A3), which has a test-set accuracy of 0.883 ± 0.006, recovering $80 \pm 2\%$ of the SNe in the test set, with $10 \pm 4\%$ of false positives.

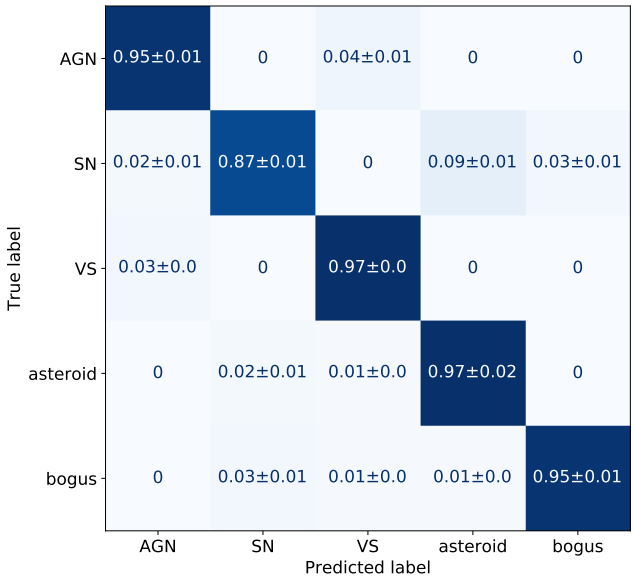


Figure 4. Average confusion matrix for the test set using 5 different realizations of the stamp classifier.

By inspecting the predictions made by our model for each SN sample in the test set, we found that the results are in agreement with our initial expectations regarding the class discrimination described in Section 2, and the characteristics presented within the three stamps for each sample. Figure 5 shows SNe examples from TNS that have been corretly classified by our model, where in most cases a host galaxy is present, which is a good indicator of an alert triggered by a SN. In the examples shown in Figures 5c and 5d, the second most likely class is AGN, due to the spatial coincidence of the transient with the center of the host galaxy.

In Figure 6, incorrectly classified examples are shown. The examples in Figures 6a, 6b and 6c are SNe from TNS classified as Asteroids by our model. The absence

[4] accuracy $= \frac{\text{N° correct classifications}}{\text{Total N° of samples}}$

of a host galaxy in these cases reduces the probability of an alert to be triggered by a SN. In the samples shown in Figures 6d, the confusion occurs between SN and bogus alerts. In this case, the confusion is likely due to the small size of the point spread function (PSF) for some observations, which leads to confusion between true variables and hot pixels or cosmic rays (these often appear as single or a few adjacent bright pixels on the image).

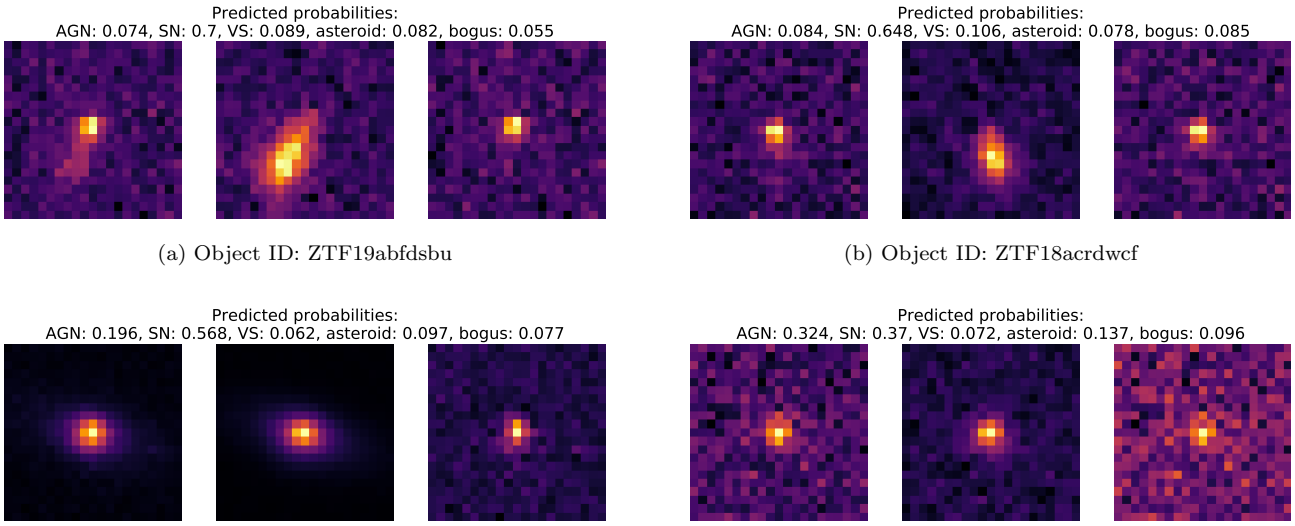
It is worth highlighting that the results of our model are achieved by using **the first alert only**. According to the confusion matrix, the most probable missclassification for SN candidates are asteroid and bogus classes. This confusion between SN, asteroids, and bogus could be fixed by looking at the second alert of the same object. If the second alert exists, it is safe to discard the bogus and asteroid classes, since it is extremely unlikely that the same bogus error or a moving object will appear in the exact same location in consecutive images, unless the alert is near a bright star that produces pixel errors due to saturation.

An example of the effect of the regularization term is depicted in Figure 7. Considerable differences in the distribution of the predicted probability for each class can be observed by varying β between 0 and 1, since both terms in eq. 1 are expected values of log probabilities. In the case of $\beta = 0$, the predictions are mostly saturated around 0 or 1 for the SN, VS, Asteroids and Bogus alert classes, creating difficulties to identify stamps that seem equally likely to belong to more than one class, because every sample is mapped to similar levels of high certainty. As the value of β increases, the saturation of predicted values decreases, spreading the predicted probability distributions and emphasizing the different levels of certainty between predictions of different samples. The order of predicted probabilities for each sample does not change significantly by varying β, achieving 99% of accuracy in the test set by checking whether the correct label lies in the highest two predicted probabilities for different β. The use of regularization to find noticeable differences in the predicted probabilities could be helpful to an expert for evaluating the output of the classifier, gaining better insight into how reliable the classifications are.

As a consistency check, we predicted the classes of unlabeled candidates using the stamp classifier, in order to compare their spatial distribution to the expected spatial locations for each class as mentioned in Section 2. To gather the unlabeled candidates, we queried objects using the ALeRCE API[5] by selecting 390,498 first alerts of different objects, chosen to be uniformly distributed over the full sky coverage of ZTF, where 325,582 of the alerts come from objects with > 1 alert (SNe, AGN, and VS) and 64,916 come from objects with only 1 alert to have a better representation of asteroids and bogus candidates. Figure 8 shows the spatial distribution of the predictions made by our model over the unlabeled data. As expected, the spatial distribution for extragalactic sources (SNe and AGNs) has a lower density of predicted candidates near the Galactic plane. On the contrary, the spatial distribution of VS candidates is more concentrated toward the Galactic plane. In the case of asteroids, these are found near the ecliptic. It is also possible to see a slight trend of predicted SNe near the ecliptic due to the confusion with asteroids class when there is no apparent host galaxy in the stamp, as shown in Figure 6. In addition, we included the distribution for the same unlabeled data classified by the CNN without including features, shown in Figure A4 in Appendix D. It is noticeable the presence of predicted extragalactic objects (SNe and AGNs) within the Galactic plane, and a higher density of predicted asteroids far from the ecliptic. Even though the images alone have important information to classify the five classes, the features are essential to improve the accuracy of the classifier in the labeled dataset as shown in the Appendix D, in addition to obtain the expected spatial distribution for each class.

We use the stamp classifier on a daily basis to filter suitable SN candidates to report for follow-up. The filtered candidates are inspected by experts (F. Förster, F. E. Bauer, G. Pignata, J. Silva-Farfn, E. Camacho-Iiguez, L. Galbany) to choose the most reliable candidates to report among the ones indicated by the classifier. Therefore, it is important to control the false positive ratio and the amount of classified SNe events. To understand this trade-off, we computed the Receiver Operating Characteristic (ROC) curves depicted in Figure 9. To build the ROC curve, we converted the classification problem into a detection problem by making a binary classification between SN vs. the rest of the classes (AGN, VS, asteroids and bogus alerts). Using the predicted probabilities in the test set of each alert being a SN, we varied the threshold value (minimum probability) necessary to assign the SN class to an alert and change the operation regime of the model. By choosing a high SN probability threshold, the false positive ratio can be reduced in order to decrease the number of false candidates in the

[5] https://alerceapi.readthedocs.io/en/latest/



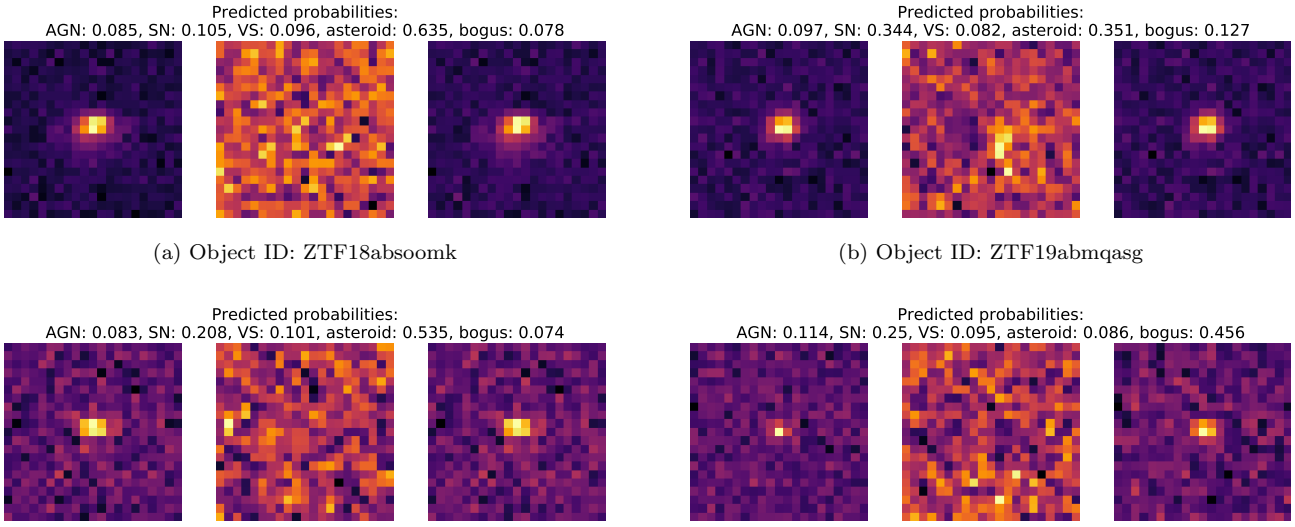
(a) Object ID: ZTF19abfdsbu

(b) Object ID: ZTF18acrdwcf

(c) Object ID: ZTF18abuhzfc

(d) Object ID: ZTF19abrirdm

Figure 5. Correctly classified SN examples, with their respective predicted probabilities according to the proposed model. Panels (a) and (b) show typical examples of well-classified SNe, where the presence of a host galaxy within the stamps increases the chances of a SN alert being triggered. Panels (c) and (d) show small confusions between SN and AGN, due to the spatial coincidence of the transient with the center of the host galaxy.



(a) Object ID: ZTF18absoomk

(b) Object ID: ZTF19abmqasg

(c) Object ID: ZTF19aazlsfj

(d) Object ID: ZTF19abpbvsk

Figure 6. Incorrectly classified SN examples, with their respective predicted probabilities by the proposed model. In Panels (a), (b) and (c), the SNe are classified as asteroids. The SN in panel (d) is classified as a bogus alert, which might be caused by the small size of the PSF, confusing the classifier with a hot pixel or a cosmic ray, which usually occupies a very narrow portion of the stamp at the center. In all cases, the absence of a clear host-galaxy within the stamps reduces the probability of a SN alert being triggered.

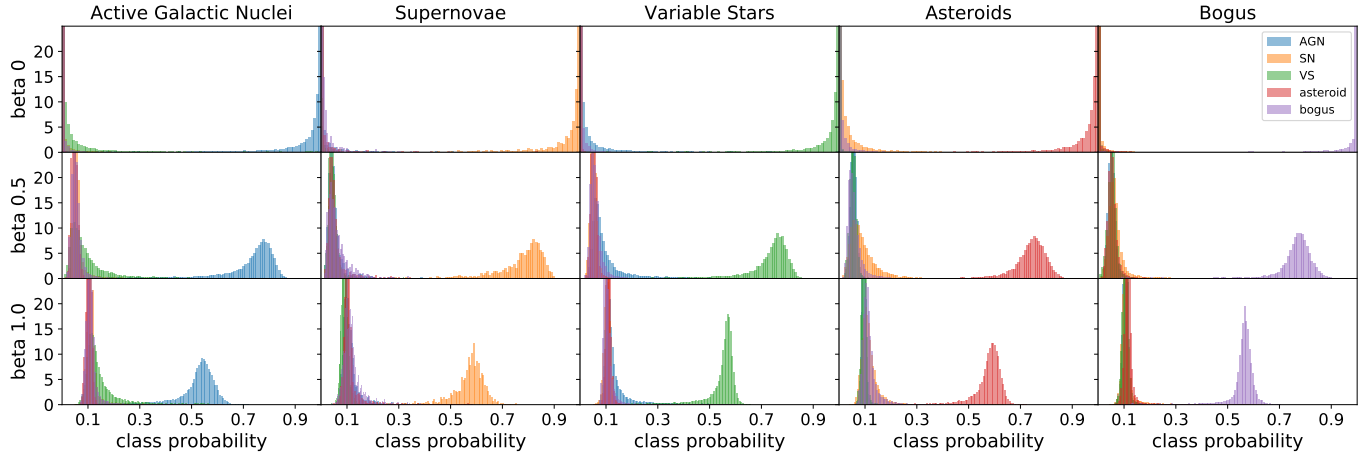


Figure 7. Probability distribution for each of the classes in the training set, for different values of the regularization constant $\beta = \{0, 0.5, 1.0\}$. For the model without regularization ($\beta = 0$ shown on the top plot), the probability distribution saturates to 1 or 0. Increasing β to 0.5 or 1.0 decreases the saturation and spreads the distribution of predictions made by the model (mid and bottom plots).

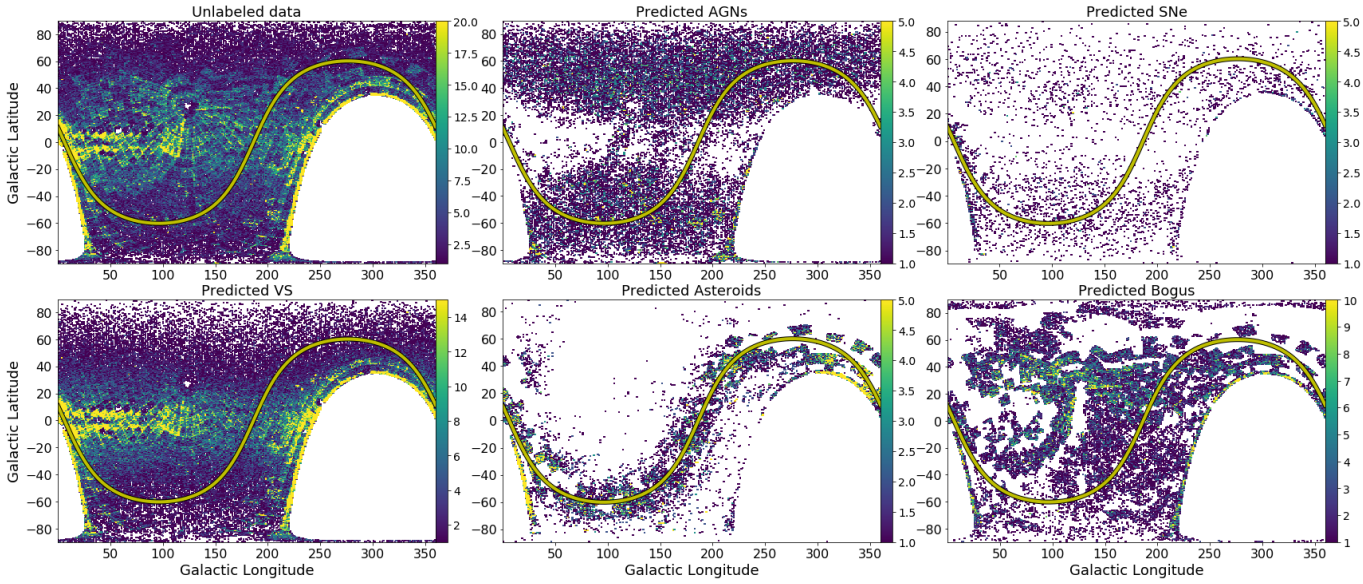


Figure 8. Spatial distribution for the unlabeled data, and distribution of predictions per class. The colorbar indicates the density of points. The ecliptic is shown with a yellow line with black edges. The distributions are shown as a 2d histogram of density of alerts. Extragalactic sources (SNe and AGNs) are found outside the Galactic plane. On the contrary, VS are concentrated in the Galactic plane. Asteroids are near the ecliptic.

list for inspection by experts, while keeping a high true positive ratio. For instance, for a SN probability threshold of 0.1, 0.2 and 0.5, the false positive ratio is 0.87, 0.03 and 0.01 respectively, while the true positive ratio is 0.94, 0.92 and 0.83, respectively. Our model is suitable to be used to process large volumes of alerts, when limited resources for manual inspection and confirmation by means of follow-up observations are available.

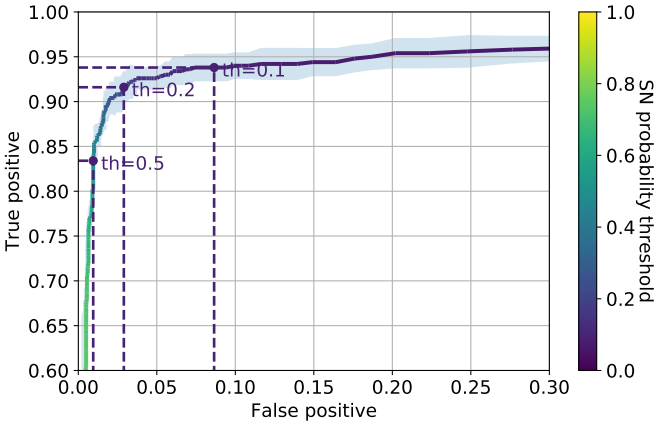


Figure 9. ROC curve with SN detection threshold. The colorbar shows the threshold that a sample's predicted SN probability must surpass in order to be assigned as a member of the SN class. For a SN probability threshold of 0.1, 0.2 and 0.5, the false positive ratio is 0.87, 0.03 and 0.01, respectively, while the true positive ratio is 0.94, 0.92 and 0.83, respectively.

5. MODEL DEPLOYMENT AND SN HUNTER

The SN Hunter (https://snhunter.alerce.online) is a visualization tool that allows the user to inspect SN candidates classified by the model in real time, in order to select good targets for follow-up observations. The interface of the SN Hunter is shown in Fig. 10. At the left of the interface, a celestial map shows the position of each candidate with a circle, where the size of the circle is proportional to the class probability assigned by our model, with the map centered on the right ascension (`ra`) and declination (`dec`) coordinates of the alert. The Milky Way plane is highlighted by the regions with lighter shades of purple. The green curve in the map represents the ecliptic, where SN candidate alerts are more likely to be triggered by asteroids instead of real SNe. The right side of the interface provides a table where the highest probability SN candidates are listed. The table shows the ZTF Object ID which uniquely identifies each astronomical alert, the discovery date specifying day, month, year and time where the first alert was triggered, the corresponding SN probability (score) from the stamp classifier, and the number of available alerts in the r and g bands (#Obs) since the discovery date. The list can be sorted by object, discovery date, score or number of alerts. The total number of high probability candidates shown in the table and maximum age of the candidates can be modified by the user. By clicking on a given candidate row, a new visualization panel is deployed as shown in Figure 11, with detailed information for the selected candidate.

The visualization panel contains some metadata on the left side (see Figure 11), including the ZTF Object ID (which is a clickable link to open the objects as a new tab within the ALeRCE online main frontend, https://alerce.online/), `ra` and `dec` coordinates of the alert, the filter in which the first detection was made, the PSF magnitude and the observation date. Below comes the PanSTARRS cross-match information, containing the Object ID, distance to the first closest known object, and the `classtar` score of the first closest known object, where a score closer to 1 implies higher likelihood of it being a star. The buttons below this information, from left to right, correspond to queries with the ALeRCE frontend, the NASA Extragalactic Database (NED[6]), TNS, and the Simbad Astronomical Database (Wenger et al. 2000) around the position of the candidate. Finally, the full metadata associated with the first alert of the SN candidate is linked below these buttons. The middle panel of Figure 11 contains an interactive color image from PanSTARRS DR1 (Chambers et al. 2019), centered around the source using Aladin (Bonnarel et al. 2000; Boch & Fernique 2014); this image is also available in the main frontend of ALeRCE. The right panel of Figure 11 provides the science, reference and difference stamps of the first detection. It is also possible to sign in with a user account and label candidates as either a possible SN or bogus by clicking the corresponding buttons below the image stamps. These can be used to build up larger training sets, as well as select candidates for the Target and Observation Managers (TOMs).

We implemented the CNN stamp classifier using TensorFlow 1.14 (Martn Abadi et al. 2015) and deployed it to classify the streaming alerts from ZTF's Kafka server[7]. The timespan between a ZTF exposure and its first arrival as an alert from the stream is 14.6 ± 4.5 minutes. Once the alert is received by ALeRCE, it takes a few seconds for the candidate to be listed in the Super-

[6] The NASA/IPAC Extragalactic Database (NED) is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. https://ned.ipac.caltech.edu/

[7] https://kafka.apache.org/

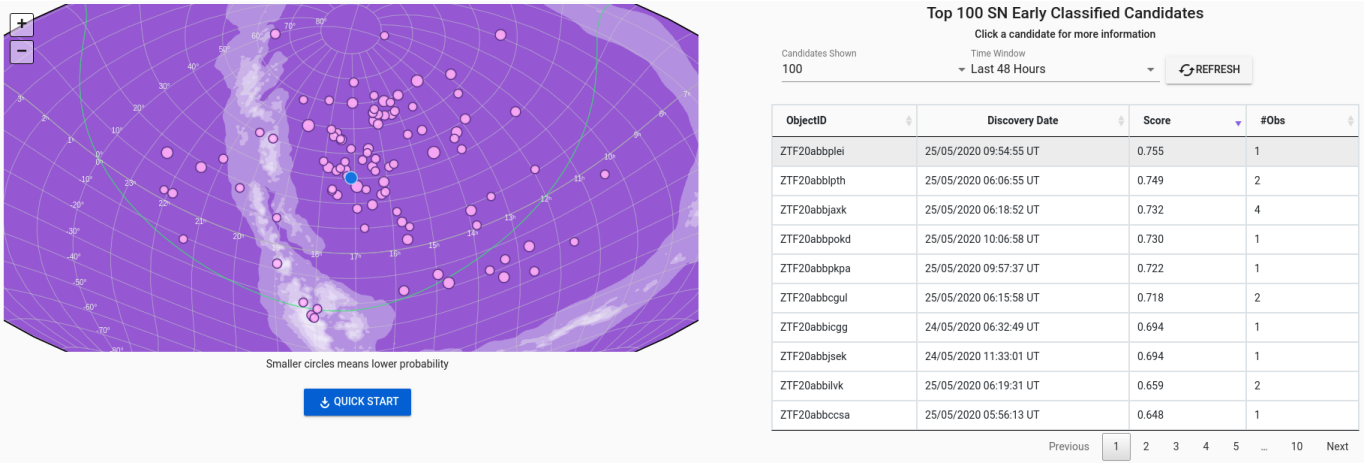


Figure 10. SN Hunter, a tool for the visualization of SNe candidates. On the left side, the location of each candidate in sky coordinates with respect to the Galactic plane and the ecliptic are depicted. On the right side, a selection of the top candidates is listed, initially ordered by SN probability score from the stamp classifier. The list of candidates can be sorted by other parameters, and updated/refreshed to include newly ingested alerts.

nova Hunter tool for expert inspection. Further details about the complete processing pipeline are described in Frster et al. (2020).

5.1. *Additional Visual Selection Criteria*

We note that the SN candidate sample presented in this and the following subsections resulted from an older version of the Stamp Classifier which relied only the three images within the first alert and did not use features for SN classification. Moreover, some of the filtering steps we applied manually are no longer necessary now that features are included (we note these below). As shown in Appendix D, even without the metadata features, the classifier provides reasonably high accuracy (only 6% worse than the model with features). Regardless of whether features are included or not, we found it critical to visually inspect the predicted SNe candidates in order to weed out misclassifications and submit more reliable candidates to TNS.

There are some common characteristics among the higher confidence SN candidates. As mentioned in Section 2, most confirmed SNe are located on top or near an extended galaxy. If there is no galaxy within the stamps, then it is more likely that is a variable star or asteroid, when the candidate is located near the ecliptic or the Milky Way, respectively, or bogus. In some cases, it is difficult to tell if the nearest source to the alert in the science image is an extended galaxy or star; for these, a search of archival catalogs and/or an assessment of the spectral energy distribution can further aid classification. Therefore, the star galaxy score from PanSTARRS in Figure 11 should be closer to 0, indicating that the extended source is more likely to be an extended galaxy. Real SN should have a positive flux in the difference image, so we removed candidates that have negative flux in the latter, by checking if the field `isdiffpos` value in the metadata is false, this is automatically done in the current pipeline. It is also important to check that the object is visible in the difference image.

Another relevant feature is the shape of the candidate, which should be similar to other stars with fuzzy edges and generally symmetric in shape. If the shape of the candidate is sharp (pixelized) or very localized, it might be a cosmic ray or a defect of the CCD camera. Alternatively, if it is elongated, it could be an asteroid or a satellite (often seen as a streak or multiple small dashes due to rotational reflections). After doing all of these checks, if the candidate is not convincing enough, then it is helpful to look at the next detections when available and search for the characteristics mentioned, which can be done using the ALeRCE frontend by clicking the ALeRCE button in the SN Hunter tool and querying directly that specific candidate's data. The 100 highest probability SNe candidates each day are manually inspected by astronomers of the ALeRCE team, and all of them must be in agreement before a candidate is reported to TNS.

5.2. *Reported and Confirmed Supernovae*

From June 26th 2019 to June 21th 2020, we have reported 3060 new SN candidates to TNS, increasing this number by 9.2 SNe per day on average, of which 399 have been observed spectroscopically. Table 4 shows the number of candidates for each confirmed class, of which 359 were confirmed as SNe.

In Figure 12, we show the cumulative distribution of candidates reported to TNS from June 26th 2019 to June 21th 2020. The cumulative distribution is sepa-

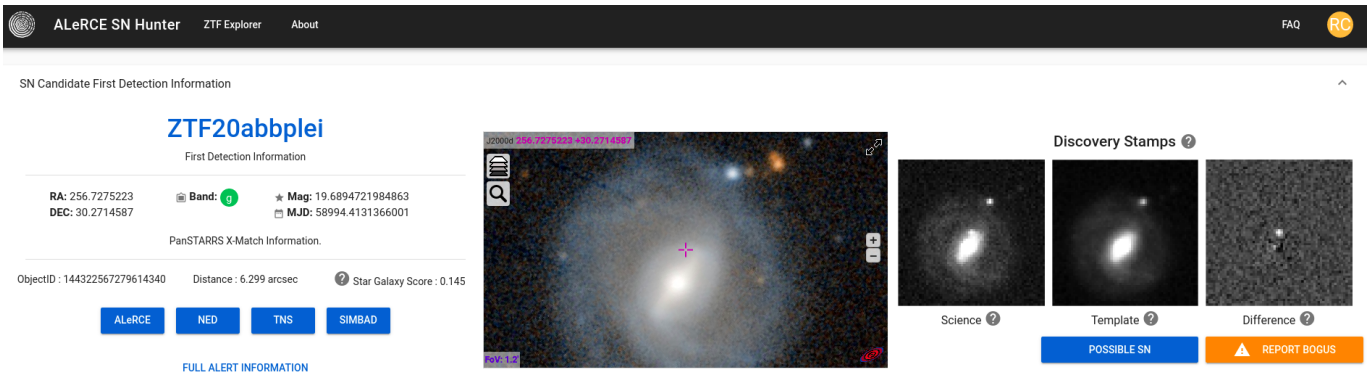


Figure 11. Candidate visualization in the Supernova Hunter tool. On the left side, the SN candidate ID is shown as a clickable link to the ALeRCE frontend, with relevant metadata such as `ra`, `dec`, magnitude, date, etc. At the bottom there are links to other sources of information, including ALeRCE, NED, TNS, and the Simbad Astronomical Database. In the middle of the figure there is a colored image from Aladin. On the right side, the stamps of the first detection are shown, along with buttons for reporting the candidate as eventual bogus or as a possible SN.

Table 4. Spectroscopically observed candidates discovered by ALeRCE, with a total of 359 SNe, 5 non-SN objects (2 galaxies, 1 TDE, 1 variable star and 1 AGN).

Confirmed class	Spectroscopically observed candidates
SN Ia	265
SN II	63
SN IIn	13
SN Ic	10
SN Ib	9
SN IIP	9
SN Ia-91T-like	7
SN Ic-BL	6
SN IIb	4
SN	2
Galaxy	2
SN Ia-91bg-like	2
SN Iax[02cx-like]	1
SN Ia-pec	1
SN Ib/c	1
SN Ibn	1
TDE	1
Variable star	1
AGN	1

rated into two parts, namely the alerts with more than one detection to date (orange) and the alerts with a single detection to date (blue). We can consider the percentage of candidates with more than one detection to be a lower bound of real non-moving astronomical objects; we define this as "purity", since multiple associated detections are a clear sign of a real non-moving astronomical object rather than a moving object or bogus alert. Candidates with only single detections to date could be due to several reasons: moving objects, bogus alerts, relatively short transients that were only above the detection threshold for a short period of time, and objects that are in locations which have not been visited again by the public ZTF Survey since the object detection.

We find that ≈70% of our reported candidates are detected with multiple alerts, while ≈30% have only one detection. Currently, we are achieving an ≈80% of purity estimated with a moving average. Among the 2187 SNe candidates with multiple alerts, 399 have been observed spectroscopically. Table 4 lists the number of candidates for each confirmed class, of which 394 were confirmed as SNe. Four of the remaining five are misclassifications: one variable star; one tidal disruption event (TDE); one AGN, and one galaxy. The final object, ZTF19aciiuta, is classified by TNS as a galaxy, but is likely be a SN (as the object has more than one detection, with a SN shaped light curve, and was reported by other groups to TNS). Even though TDEs are not SNe, follow-up of these events is still of significant interest, due to their relative scarcity (van Velzen et al. 2020). In summary, taking into consideration the conservative final candidate selection done by the team of astronomers to perform spectroscopic confirmation, our reported and confirmed candidates have around 1% contamination by non-SNe objects.

For comparison purposes, we gathered the objects reported by both ALeRCE and AMPEL (Alert Management, Photometry and Evaluation of Lightcurves, which is an internal ZTF classification effort; Nordin et al. 2019) to TNS, and compared the reporting times within 3 days after the first detection. Figure 13 shows the cumulative histogram of reporting times to TNS for

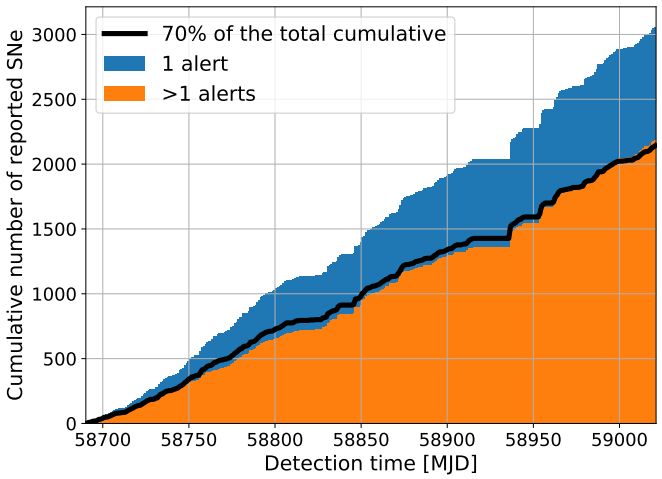


Figure 12. Cumulative number of reported SNe since we started reporting on June 26th 2019 to June 21th 2020. The average rate of reporting is 9.2 candidates per day. Roughly ≈70% of our reported candidates are detected with multiple alerts (purity), implied they are true SNe, while ≈30% have only one detection and thus less certainty. Currently, we are closer to ≈80% of purity.

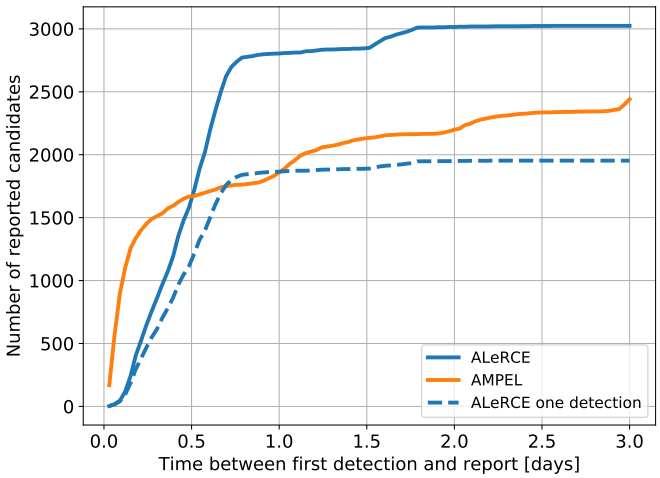


Figure 13. Cumulative distribution of time between the first detection and the reporting time from TNS, for candidates reported by ALeRCE and AMPEL. The full distributions are shown with solid lines, and the distributions of reporting time for candidates with a single detection are shown with segmented lines.

ALeRCE and AMPEL, along with the cases where reports were done by ALeRCE before having the second detection in the public stream (one detection). Approximately 64% of the candidates reported by ALeRCE were based on a single detection. An important difference between both systems is the visual inspection by experts in the reporting process to TNS. According to Nordin et al. (2019), AMPEL reports candidates automatically using their "TNS channel", which produces more reported candidates than our system, within 12 hours after the first detection. As described in this work, our system's final stage so far relies on human inspection, checking and reporting, which occurs within 10 to 24 hours after the first detection, without reporting transients already reported by AMPEL (only two cases were reported after AMPEL). Therefore, we report new candidates to TNS within a day after the first detection. Besides, since ALeRCE is mostly reporting candidates with a single detection, 92% (2804) of the reports were sent within one day after the first detection, compared to 27% (1860) from AMPEL.

Figure 14 shows the distribution of times between first or second detection and last non-detection for candidates reported by ALeRCE. Based on the data shown in Figure 14, the average time between the last non-detection and the first detection is 3.8 days, and 8.5 days for the second detection. Reporting candidates only after the second detection would result in a delay of 4.7 days on average, which represents a potentially critical timespan to measure the spectra at early stages of the transient event, as required in order to achieve the science goals described in Section 1. As mentioned before, ALeRCE currently does not report candidates that were previously reported by other groups, therefore our candidates reported using a single detection increase the bulk of objects available for early follow-up of transients that were not found by other groups. We will report already reported candidates in the near future, since this adds the additional information that the candidates passed our visual inspection test. In addition, the work presented is a starting point towards our goal of developing an automatic reporting systems of the most highly confident subset of SN candidates.

6. CONCLUSION AND FUTURE WORK

As part of the ALeRCE Broker processing pipeline, we identified characteristics of the images and metadata within the ZTF alert stream that allow us to discriminate among SN, AGN, VS, asteroids and bogus alerts. In order to solve this classification problem automatically and quickly identify the best SN candidates to perform follow-up, we trained a CNN. This classifier uses **the first detection only**, i.e., its inputs are the science, reference and difference images, and part of the metadata of the first detection alert. In addition, our CNN architecture is invariant to rotations within the stamps, and was trained using an entropy regularized loss function. The latter is useful to improve human readability in predicted probabilities per class, in terms of certainty assigned to each sample, so an expert can gain better

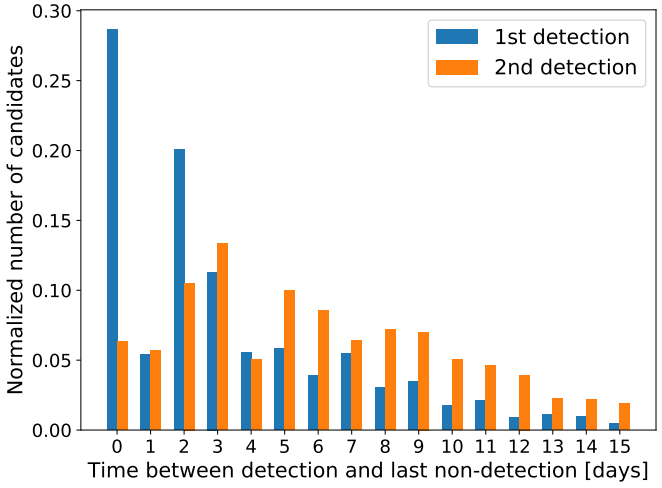


Figure 14. Histogram of time between first (second) detection and the last non-detection of the ALeRCE reported candidates.

insights into the actual nature of the transients when inspecting SN candidates.

Among all five classes that our CNN can classify, it achieves an accuracy of 0.941 ± 0.004 on a balanced test set, while in the SN class reaches a true positive rate and a false positive rate of $87 \pm 1\%$ and $5 \pm 2\%$, respectively. By manually inspecting the classification of each sample, we found that the incorrectly classified objects are in concordance with our hypotheses regarding separability of classes using only the first detection images. Moreover, the CNN model successfully classify the alerts in the labeled set by using the images only, but when applied to unlabeled data we find some flaws by inspecting the spatial distribution of each predicted class, for instance, a concentration of extragalactic sources within the Galactic plane, and a higher density of asteroids far from the ecliptic. By giving the alert metadata as additional features to the classifier, we find that the spatial distributions of the events are in agreement with the expectations, according to their tentative nature. More specifically, extragalactic classes (SNe and AGNs) are found outside the Galactic plane, VS have a higher density of predicted objects within the Galactic plane, and asteroids are found around the ecliptic.

The proposed CNN classifier is deployed and its predictions are publicly available in an especially designed visualization tool for inspection of candidates with high SN class probability, called the SN Hunter (https://snhunter.alerce.online/). The predictions are also available in the ALeRCE main frontend. This tool shows relevant information about the SN candidates in order to facilitate their analysis, and we used it to report SN candidates on TNS for follow-up. We also presented a visual inspection methodology that relies on the information presented in the Supernova Hunter tool, such as the probability assigned by our classifier to the SN class, alert metadata, position in the sky with respect to the Galactic plane and the ecliptic, number of detections, etc. By manually inspecting candidates using the SN Hunter tool, from June 26th 2019 to June 21th 2020, our team has reported 3060 candidates for follow-up, out of which 399 were tested, and 394 were spectroscopically confirmed as SN. Besides, the interface allows experts to manually label bogus and SN candidates alike, which helps improve our training set.

As many as 92% of the candidates reported to TNS by ALeRCE were reported within one day after the first detection, and 64% of the candidates were reported using a single detection, where 70% have multiple detections, confirming extragalactic nature. Since ALeRCE does not report objects that had been previously reported, these results correspond to the transients that were not detected or chosen by other groups, therefore adding new early transient reports to TNS.

We are currently working on improving the training set by adding more examples from confirmed SNe, and manually adding bogus candidates to the training set. Furthermore, we are exploring ideas for model interpretability, adding visualization tools that may help understand why the model predicts a given class for a given event. We are working on using LRP (Bach et al. 2015; Montavon et al. 2019) and occlusion techniques (Zeiler & Fergus 2014) to show what part of the input influences the decision in a specific way, so the expert can use it to choose better candidates.

Regarding the performance of the model, an additional step would be to extend the system to be able to process more than a single alert while keeping the capability of performing well with only one alert. New approaches about how to achieve this have been explored in Carrasco-Davis et al. (2019) and Gmez et al. (2020), feeding a neural network sequentially with the data available so far, and improving the prediction every time a new measurement arrives. In the near future our efforts regarding alert classification, and particularly the SN detection problem, will aim towards the automatization of the entire process of classification of the data stream and reporting objects for follow-up, eliminating or bringing expert assistance to a minimum.

The methodology proposed in this work is suitable to other streams of data based on alerts, such as ATLAS (The Asteroid Terrestrial-impact Last Alert System; Tonry et al. 2018) and the Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019). The latter presents a further challenge

in terms of the amount of data generated, restrictions in processing time due to the data generation rate, the larger number of filters, the lack of comparison catalogs at the survey's depth, the smaller field of view per stamp (currently planned to be only 6"×6") and limited contextual information, and the possibility that either the science or reference image may not be contained in the alerts. We think our work on ZTF data will be a valuable precursor for the next generation of large etendue telescopes.

7. ACKNOWLEDGMENTS

The authors acknowledge support from the Chilean Ministry of Economy, Development, and Tourisms Millennium Science Initiative through grant IC12009, awarded to the Millennium Institute of Astrophysics (RC, ER, CV, FF, PE, GP, FEB, IR, PSS, GC, SE, Ja, EC, DR, DRM, MC) and from the National Agency for Research and Development (ANID) grants: BASAL Center of Mathematical Modelling AFB-170001 (FF) and Centro de Astrofsica y Teconologas Afines AFB-170002 (FEB, PSS, MC); FONDECYT Regular #1171678 (PE), #1200710 (FF), #1190818(FEB), #1200495 (FEB), #1171273 (MC), #1201793(GP); FONDECYT Postdoctorado #3200250 (PSS); Magster Nacional 2019 #22190947 (ER).

Software: Aladin (Bonnarel et al. 2000), Apache ECharts[8], Apache Kafka[9], Apache Spark (**?**), ASTROIDE (BRAHEM et al. 2018), Astropy (Astropy Collaboration et al. 2013), catsHTM (Soumagnac & Ofek 2018), Dask (Rocklin 2015), Jupyter [10], Keras (Chollet & others 2018), Matplotlib (Hunter 2007), NED (Steer et al. 2016), Pandas (McKinney 2010), Prometheus[11], Python[12], scikit-learn (Pedregosa et al. 2011), Simbad-CDS (Wenger et al. 2000), Tensorflow (Martn Abadi et al. 2015), Vue[13], Vuetify[14], PostgreSQL[15].

APPENDIX

A. EXPLORING THE RELATIONSHIP BETWEEN FEATURES AND CLASSES

Table A1. Clipping values for each feature. "max" or "min" in the clipping range for each feature means that the maximum and minimum value is preserved for that feature respectively.

Feature	[min value, max value]
`sgscore1`	[-1, max]
`distpsnr1`	[-1, max]
`sgscore2`	[-1, max]
`distpsnr2`	[-1, max]
`sgscore3`	[-1, max]
`distpsnr3`	[-1, max]
`ifwhm`	[min, 10]
`ndethist`	[min, 20]
`ncovhist`	[min, 3000]
`chinr`	[-1, 15]
`sharpnr`	[-1, 1.5]
non detections	[min, 2000]

[8] https://echarts.apache.org
[9] https://kafka.apache.org/
[10] https://jupyter.org/
[11] https://prometheus.io/
[12] https://www.python.org/
[13] https://vuejs.org/
[14] https://vuetifyjs.com/
[15] https://www.postgresql.org/

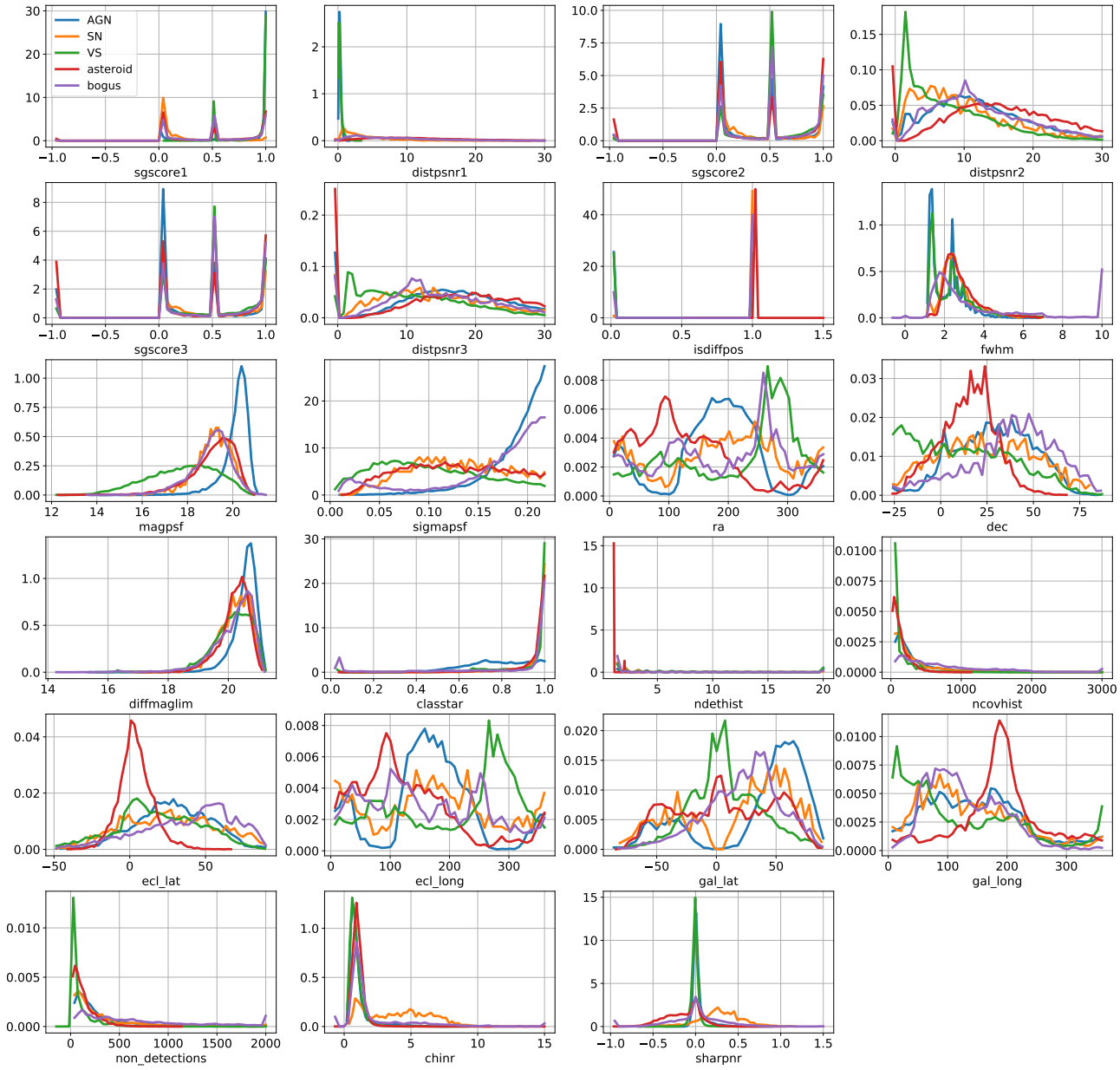


Figure A1. Feature distribution per class of the labeled dataset. Each feature was clipped to the values given in Table A1.

B. CNN GLOSSARY AND TRAINING

B.1. *CNN architecture*

- **Fully connected layer**: Artificial neural networks (ANNs) are mathematical models that are mostly used for classification or regression. ANN make use of basic processing units called *neurons*, which receive vectors $\mathbf{x}$ of data as input, then apply a linear function to them, followed by a non-linear activation function. These neurons are grouped in *layers*, which are called *fully connected layers*. The output produced by a set of neurons of a specific fully connected layer is calculated as:

$$\mathbf{y} = \phi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{B1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input of the layer, $\mathbf{y} \in \mathbb{R}^m$ is the output of the layer, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is a matrix of parameters called *weights*, $\mathbf{b} \in \mathbb{R}^m$ is a vector which contains the so-called *biases* of the layer, and $\phi(\cdot)$ is a non-linear activation

function that follows the linear transformation of x. There are all sort of flavors of non-linear activation functions, the most commonly used are:

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}, \tag{B2}$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}, \tag{B3}$$

$$\text{ReLU}(x) = \max\{0, x\}. \tag{B4}$$

To be precise, $\mathbf{W}$ and $\mathbf{b}$ are referred to as the parameters of a fully connected layer, and they are modified during training to be optimized for the task at hand. Fully connected layers can be sequentially stacked one after the other to integrate an ANN model. For instance, an ANN of two layers, is defined as:

$$\mathbf{z} = \phi(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \tag{B5}$$

$$\mathbf{y} = \phi(\mathbf{W}^{(2)}\mathbf{z} + \mathbf{b}^{(2)}). \tag{B6}$$

The parameters of the ANN are $\theta = (\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)})$. The way of grouping neurons and layers in an ANN is called the *architecture* of an ANN.

- **Softmax output layer:** A commonly used activation function at the output of ANN models is the sigmoid(x), whose output is bounded by $(0, 1)$, and can be interpreted as the probability of activation of a neuron, a property useful for binary classification. A generalization of the aforementioned function, useful for multi-class classification models, is the *softmax* activation function, usually referred to as *softmax output layer*, where there are K neurons x_i, $i \in \{1, \ldots, K\}$, and it is desired to assign a probability to each one, hence, requiring that they add up to one. This is done by the softmax activation function, defined as:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, \quad i \in \{1, \ldots, K\}. \tag{B7}$$

- **Convolutional layer:** ANN with fully connected layers are limited to vector-like inputs, and they do not take into account the presence of correlation between adjacent features. To overcome this limitation, and preserve a degree of spatial or temporal correlation in the input of models, CNNs were proposed. The main component of CNNs are *convolution layers*, which apply a filter or kernel to the input of the layer by a convolution operation. Similar to a fully connected layer, convolutional layer outputs are calculated as follow:

$$\mathbf{y} = \phi(\mathbf{W} * \mathbf{x} + \mathbf{b}), \tag{B8}$$

where $\mathbf{x}$ stands for the input of the layer, $\mathbf{y}$ is the output of the layer, $\mathbf{W}$ is a set of filters to apply by convolution to the input, $\mathbf{b}$ is the vector of biases for each filter, and $\phi(\cdot)$ is the activation function. In this case the $*$ operation between $\mathbf{x}$ and $\mathbf{W}$ is a convolution. The model used in this work applies convolutions to images, so $\mathbf{x} \in \mathbb{R}^{e\times f\times g}$ and $\mathbf{y} \in \mathbb{R}^{u\times v\times l}$ are 3d tensors, while $\mathbf{W} \in \mathbb{R}^{d\times d\times t\times l}$ and $\mathbf{b} \in \mathbb{R}^{l}$. The calculation of every element $y_{i,j,k}$ of $\mathbf{y}$ is derived from the operation of the convolutional layer as follows:

$$y_{i,j,k} = \sum_{m,n,p} x_{i-m,j-n,p} W_{m,n,p,k} + b_k, \tag{B9}$$

where every element i, j, k of the tensor $\mathbf{y}$ is calculated by moving the filters of $\mathbf{W}$ over the tensor $\mathbf{x}$ and applying eq. B9. Each time $\mathbf{W}$ moves over the first two dimensions of $\mathbf{x}$, it skips S pixels, where S is called stride. After applying the convolutional layer, the first two dimensions of $\mathbf{y}$ are smaller in size than the ones of $\mathbf{x}$. The spatial dimensions (first and second dimension) of the tensors $\mathbf{x}$ and $\mathbf{y}$ relate to each other as follows:

$$U = \frac{E-D}{S} + 1, \tag{B10}$$

where U is the size of any of the spatial dimensions of $\mathbf{y}$, E is the size of the respective spatial dimension of $\mathbf{x}$, D is the respective spatial dimension of $\mathbf{W}$ and S is the stride used in the convolution operation.

- **Zero-padding:** It is a commonly used technique to preserve the spatial dimensions of the input $\mathbf{x} \in \mathbb{R}^{e\times f\times g}$ at the output $\mathbf{y} \in \mathbb{R}^{u\times v\times l}$ of a convolutional layer. Zero-padding consists on adding 0's to the edges of the spatial dimensions of $\mathbf{x}$. For a convolutional layer of stride $S = 1$, and kernel size D, the zero-padded input to the convolutional layer must have dimensions such as $\mathbf{x} \in \mathbb{R}^{(e+\lfloor D/2\rfloor)\times(f+\lfloor D/2\rfloor)\times g}$, where $D/2$ is the amount of zero-padding included to achieve same spatial dimensions of $e == u \wedge f == v$, between the layer's input $\mathbf{x}$ and output $\mathbf{y}$.

- **Max pooling:** Pooling layers are used in CNNs to reduce the spatial dimensionality of their inputs. The max pooling used in the model of Fig. 3 returns the maximum value within a window of its input $\mathbf{x}$, in the same way as a convolutional filter, this maximum value extraction window is rolled across the spatial dimensions of the input. In the case of the architecture of Fig. 3, the pooling window is of dimension 2×2 with a stride of 2, i.e., without overlapping of the window, yielding a spatial dimensionality reduction by half each time max pooling it is applied.

- **Batch normalization layer:** It works as a trainable normalization layer that has different behavior during training and evaluation of the model. During training, batch normalization layer calculates the mean and variance of each feature, to normalize them and compute an exponential moving average of mean and variance of the training set. After training of the model, at its evaluation, the whole population statistics adjusted during training are used to normalize evaluation inputs. Batch normalization not only normalizes input values to have a mean value near 0 or a variance value near 1, it also contains a linear ponderation of these inputs, that allows their scaling an shifting. This layer allows the model to emphasize or ignore specific inputs, acting as a regularizer and speeding up training.

- **Dropout:** It is an operation that is usually applied at the output of fully connected layers, and it is used as a regularizer of the model to avoid overfitting of layers with large number of neurons. Similar to a batch normalization layer, dropout performs different operations during training and evaluation. The dropout operation is defined by the **dropout rate** $DR \in [0, 1]$, which is a parameter that, at the training phase of the model, defines the probability of setting each of its inputs to 0, and multiplying the values not set to 0 by $1/(1 - DR)$, such that the sum over all the input values remains the same. At each training step a percentage DR of the outputs of a fully connected layer won't be used, reducing the effective size of that layer. On the other hand, when using the model after training, dropout is deactivated. The desired effect of dropout is to enforce the model to not depend on specific units of every layer.

The model described in Fig. 3, is based on Enhanced Deep-HiTS Reyes et al. (2018), a state of the art classifier for binary classification of real astronomical object and artifact samples. The architecture of this model introduced total rotational invariance, which empirically proved to enhance performance on the classification of astronomical images.

B.2. *Neural network training*

- **Procedure to train a neural network:** The objective of using a neural network f_θ of parameters $\theta \in \Theta$, is to approximate a function $y = f(x)$, with $x \in \mathcal{X}$. In practice there is no access to the whole data distribution $\mathcal{X}$, but to a subset of N data samples of the function to approximate $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, called *training set*. Finding the best parameters θ^* for the neural network $f_\theta(x)$ requires solving the optimization problem:

$$\theta^* = \arg\min_{\theta\in\Theta} \mathcal{C}(\theta) = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(y^{(i)}, f_\theta(x^{(i)})), \tag{B11}$$

where $\mathcal{C}$ is an error functional defined by the function $\mathcal{L}$ that is called **loss function**. The optimization depicted in eq. B11, is achieved by the training of the model through optimization techniques based on **gradient descent**, when $\mathcal{L}$ is chosen as a differentiable function (e.g. cross-entropy). The parameters θ are iteratively adjusted by the following rule, until convergence:

$$\theta_k = \theta_{k-1} - \mu\nabla_\theta \mathcal{C}(\theta). \tag{B12}$$

Because neural networks are composed of many consecutive layers, the direct calculation of $\nabla_\theta \mathcal{C}(\theta)$ is computationally expensive. However, they can be calculated efficiently by **back-propagation**, which is an algorithm that *propagates*

the error from the output of the model until it reaches the first layer of the neural network, back-propagation is based on the differentiation chain rule.

Even when using back-propagation, for neural networks trained on large amounts of data, the calculation of the exact gradient $\nabla_\theta \mathcal{C}(\theta)$ becomes computationally expensive. As a solution to this problem, a non biased estimation of the gradient $\nabla_\theta \tilde{\mathcal{C}}(\theta)$ to apply is used, where the same gradient is calculated over a small random fraction of data, this is called a **batch** and the amount of data samples in the batch is the **batch size**. Therefore, the optimization rule for a batch $\mathcal{B} \subset \mathcal{X}$ is:

$$\theta_k = \theta_{k-1} - \mu \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_\theta \mathcal{L}(y^{(i)}, f_\theta(x^{(i)})), \tag{B13}$$

where μ is a constant called **learning rate**, and establishes how large is the performed training step. This technique of training by batches, is a form of *stochastic gradient descent*, and it guarantees convergence when μ is a well defined sequence in k that satisfies $\sum_k \mu_k = \infty$ and $\sum_k \mu_k^2 < \infty$.

- **Adam optimizer:** An alternative to the optimization rule of eq. B13, is Adam Kingma & Ba (2017), which is an adaptive learning rate optimization algorithm that automatically adjust μ_k. It uses the squared gradients to scale the learning rate and it includes the moving average of the gradients in its formulation, strategy that is known as **momentum**, and its used to avoid converge to a local minima in the optimization. The main hyperparameters of Adam are β_1 and β_2, which relate to the moving averages of the gradients and the squared gradients, respectively, and they regulate the rate at which the learning rate μ_k is adjusted.

C. HYPERPARAMETER RANDOM SEARCH RESULTS

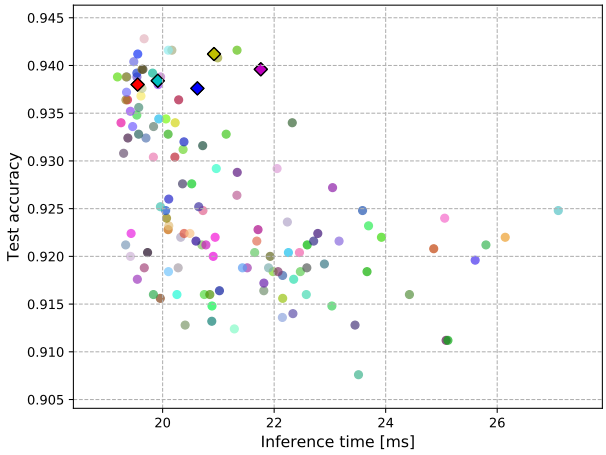
For the hyperparameter random search, 133 different combinations of hyperparameters values were sampled from Table 3, we trained 5 models for each hyperparameter set and then evaluated the test and validation set with every model. In addition, for each model the inference time over a single sample was measured. The training procedure took ≈3 days of continuous training on 5 NVIDIA GTX 1080Ti GPUs. From now on, every time we refer to accuracy or inference time of a model, they are the average measurement of 5 models trained with the same hyperparameters.

The selection of the best model from the hyperparameter random search was done by looking at the performance in the validation set. We took the 5 models with highest validation accuracy and performed a Welchs hypothesis test between the model with highest and lowest validation accuracy among the top-5 models, obtaining a p-value of 0.594, which means that the accuracy differences among the models are not statistically significant. Another important factor when processing the volume of data encountered in astronomy, is the inference time of the used models, we measure inference time for the top-5 models and a Welchs hypothesis test between the fastest and the slowest model got a p-value of 0.330, i.e. no statistically meaningful difference. Finally, the best model among the top-5 is chosen as the one with $\beta = 0.5$, because it shows the most interpretable range of prediction probabilities, according to astronomers. The model chosen as the best has a validation accuracy of 0.950±0.003, test accuracy of 0.941±0.004 and inference time of 20.5±2.6 [ms]. The performance of the top-5 models over the test set is shown both in Figure A2 and Table A2. Every time we refer to the top-5 models we mean the aforementioned models.

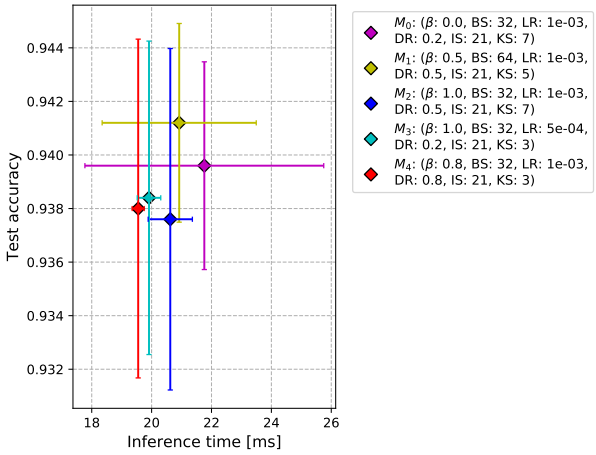
Figure A2 shows plots of test accuracy versus the inference time, Figure A2a shows the results of all the 133 models from the random hyperparameter search: In Figure A2a, the nearest to the top left corner, the better the model, the diamond shapes in this figure correspond to the models with top-5 validation accuracy. Figure A2b shows in detail the performance of these top-5, where each model has its one standard deviation error bars. These models are named M_i, where i corresponds to the position of its validation accuracy w.r.t. to all the 133 models, the lower its validation accuracy, the higher is i. Table A2 shows validation, test accuracy and inference time for the models with top-5 validation accuracy of Figure A2b, where M_1 is selected as the best model, which is used for experiments shown in previous sections. In Table A2, metrics of the best model M_1 are underlined, whereas bold metrics correspond to the highest of their respective column. Coincidentally, model M_1 chosen as the best, has the higher test accuracy among the top-5 models, and when compared to the model M_2 with worst test accuracy, the Welchs hypothesis test p-value is 0.364, meaning that the difference is not statically meaningful.

Figure A2b shows that test accuracy and inference time error bars of each of the top-5 models contain each other.

D. RESULTS OF CLASSIFIER USING IMAGES ONLY



(a) Test accuracy v/s inference time of 133 models from hyperparameter random search.



(b) Mean test accuracy v/s inference time of the 5 models with best validation accuracy from hyperparameter random search.

Figure A2. Accuracy of 133 models from hyperparameter random search. For each model, results consider 5 trainings and respective evaluations on the test set. (a) Test accuracy versus inference time, where each dot is a model with different hyperparameters, the closer to the left top corner is a model, the better its performance. Model represented with diamonds correspond to the 5 models with best validation accuracy. (b) Test accuracy versus inference time for the models represented as diamond in (a), each model has its error bars corresponding to one standard deviation and every model is denoted as M_i, where i corresponds to the ranking of its validation accuracy performance among all the 133 models of (a).

Table A2. Top-5 models with highest validation accuracy from the hyperparameter random search, ranked from M_0 to M_4. There are no statistical difference between the accuracy and inference time of the displayed models. M_1 is chosen as the best model because it has $\beta = 0.5$, which shows the most interpretable range of prediction probabilities, according to astronomers. Metrics of the best model M_1 are underlined, whereas bold metrics are the best of their respective column.

Model Name	Model's Hyperparameters	Validation Accuracy	Test Accuracy	Inference Time [ms]
M_0	β: 0, BS: 32, LR: 1e-03, DR: 0.2, IS: 21, KS: 7	**0.950±0.003**	0.940±0.004	21.8±4.0
M_1	β: 0.5, BS: 64, LR: 1e-03, DR: 0.5, IS: 21, KS: 5	0.950±0.005	**0.941±0.004**	20.9±2.6
M_2	β: 1.0, BS: 32, LR: 1e-03, DR: 0.5, IS: 21, KS: 7	0.949±0.002	0.938±0.006	20.6±0.7
M_3	β: 1.0, BS: 32, LR: 5e-04, DR: 0.2, IS: 21, KS: 3	0.948±0.003	0.938±0.006	19.9±0.4
M_4	β: 0.8, BS: 32, LR: 1e-03, DR: 0.8, IS: 21, KS: 3	0.949±0.003	0.938±0.006	**19.6±0.2**
Welch's t-test p-value M_0 v/s M_4 — M_1 v/s M_4 — M_0 v/s M_4		0.594	0.364	0.330

REFERENCES

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, \textbackslashaap, 558, A33, doi: 10.1051/0004-6361/201322068

Bach, S., Binder, A., Montavon, G., et al. 2015, PLoS ONE, 10, doi: 10.1371/journal.pone.0130140

Barchi, P. H., de Carvalho, R. R., Rosa, R. R., et al. 2020, Astronomy and Computing, 30, 100334, doi: 10.1016/j.ascom.2019.100334

Becker, I., Pichara, K., Catelan, M., et al. 2020, \mnras, 493, 2981, doi: 10.1093/mnras/staa350

Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2018, Publications of the Astronomical Society of the Pacific, 131, 018002, doi: 10.1088/1538-3873/aaecbe

Bertin, E., & Arnouts, S. 1996, Astronomy and Astrophysics Supplement Series, 117, 393, doi: 10.1051/aas:1996164

Boch, T., & Fernique, P. 2014, 485, 277. http://adsabs.harvard.edu/abs/2014ASPC..485..277B

Bonnarel, F., Fernique, P., Bienaym, O., et al. 2000, Astronomy and Astrophysics Supplement Series, 143, 33, doi: 10.1051/aas:2000331

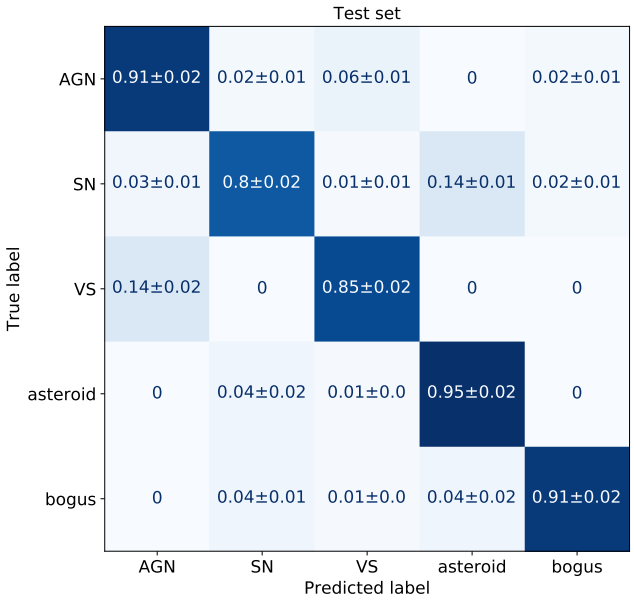


Figure A3. Confusion matrix for the test set when using the stamp classifier without alert metadata features.

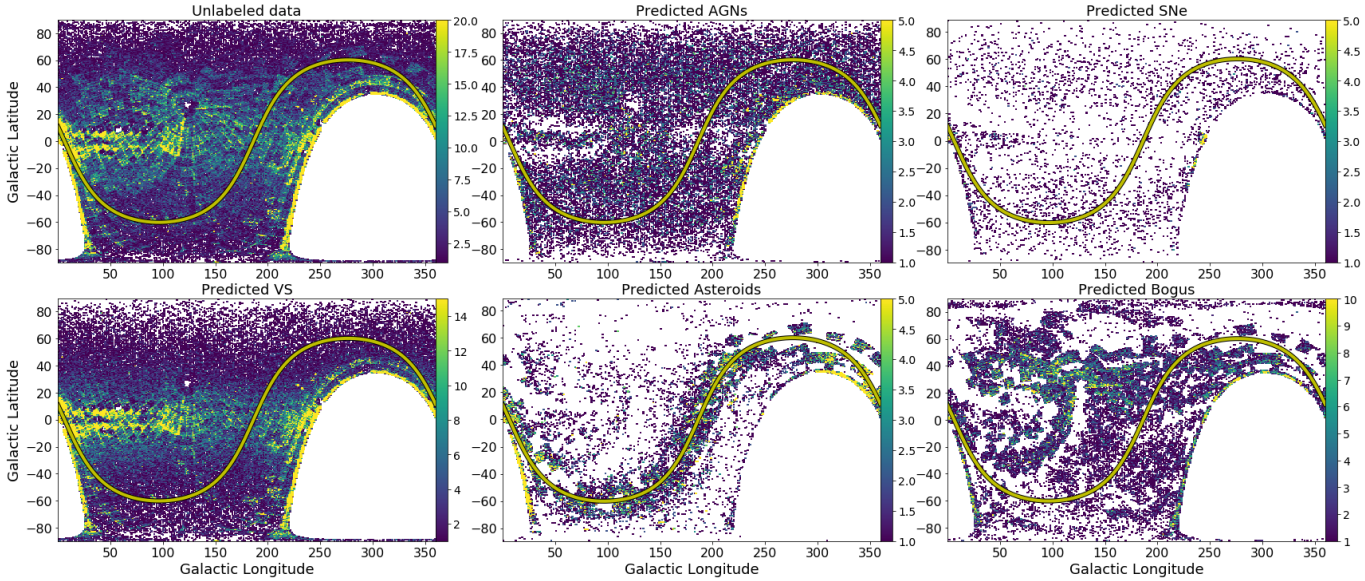


Figure A4. Space distribution for the unlabeled data, and distribution of predictions per class using the stamp classifier without features. The ecliptic is shown with a yellow line with black edges. Extragalactic classes (SNe and AGNs) are found with higher density the Galactic plane, compared to the predictions of the stamp classifier with features, shown in Figure 8. Also, the predicted asteroids by the classifier without using alert metadata features have higher density far from the ecliptic compared to the classifier that uses features.

Boone, K. 2019, The Astronomical Journal, 158, 257, doi: 10.3847/1538-3881/ab5182

BRAHEM, M., Zeitouni, K., & Yeh, L. 2018, IEEE Transactions on Big Data, 1, doi: 10.1109/TBDATA.2018.2873749

Cabrera-Vives, G., Reyes, I., Frster, F., Estvez, P. A., & Maureira, J.-C. 2017, The Astrophysical Journal, 836, 97, doi: 10.3847/1538-4357/836/1/97

Carrasco-Davis, R., Cabrera-Vives, G., Frster, F., et al. 2019, Publications of the Astronomical Society of the Pacific, 131, 108006, doi: 10.1088/1538-3873/aaef12

Chambers, K. C., Magnier, E. A., Metcalfe, N., et al. 2019, arXiv:1612.05560 [astro-ph]. http://arxiv.org/abs/1612.05560

Chollet, F., & others. 2018, Astrophysics Source Code Library, ascl:1806.022. http://adsabs.harvard.edu/abs/2018ascl.soft06022C

Dieleman, S., De Fauw, J., & Kavukcuoglu, K. 2016, arXiv:1602.02660 [cs]. http://arxiv.org/abs/1602.02660

Dieleman, S., Willett, K. W., & Dambre, J. 2015, Monthly Notices of the Royal Astronomical Society, 450, 1441, doi: 10.1093/mnras/stv632

Drake, A. J., Graham, M. J., Djorgovski, S. G., et al. 2014, The Astrophysical Journal Supplement Series, 213, 9, doi: 10.1088/0067-0049/213/1/9

Drake, A. J., Djorgovski, S. G., Catelan, M., et al. 2017, Monthly Notices of the Royal Astronomical Society, 469, 3688, doi: 10.1093/mnras/stx1085

Duev, D. A., Mahabal, A., Masci, F. J., et al. 2019, Monthly Notices of the Royal Astronomical Society, 489, 3582, doi: 10.1093/mnras/stz2357

Flesch, E. W. 2015, Publications of the Astronomical Society of Australia, 32, doi: 10.1017/pasa.2015.10

—. 2019, arXiv:1912.05614 [astro-ph]. http://arxiv.org/abs/1912.05614

Frster, F., Moriya, T. J., Maureira, J. C., et al. 2018, Nature Astronomy, 2, 808, doi: 10.1038/s41550-018-0563-4

Frster et al., F. 2020, \apj

Gal-Yam, A., Arcavi, I., Ofek, E. O., et al. 2014, \textbackslashnat, 509, 471, doi: 10.1038/nature13304

Goldstein, D. A., D'Andrea, C. B., Fischer, J. A., et al. 2015, The Astronomical Journal, 150, 82, doi: 10.1088/0004-6256/150/3/82

Groh, J. H. 2014, \textbackslashaap, 572, L11, doi: 10.1051/0004-6361/201424852

Gmez, C., Neira, M., Hoyos, M. H., Arbelez, P., & Forero-Romero, J. E. 2020, arXiv:2004.13877 [astro-ph]. http://arxiv.org/abs/2004.13877

Hunter, J. D. 2007, Computing in Science Engineering, 9, 90, doi: 10.1109/MCSE.2007.55

Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, The Astrophysical Journal, 873, 111, doi: 10.3847/1538-4357/ab042c

Jayasinghe, T., Kochanek, C. S., Stanek, K. Z., et al. 2018, Monthly Notices of the Royal Astronomical Society, 477, 3145, doi: 10.1093/mnras/sty838

Jayasinghe, T., Stanek, K. Z., Kochanek, C. S., et al. 2019a, Monthly Notices of the Royal Astronomical Society, 486, 1907, doi: 10.1093/mnras/stz844

—. 2019b, Monthly Notices of the Royal Astronomical Society, 485, 961, doi: 10.1093/mnras/stz444

—. 2020, Monthly Notices of the Royal Astronomical Society, 491, 13, doi: 10.1093/mnras/stz2711

Jiang, J.-A., Doi, M., Maeda, K., et al. 2017, \textbackslashnat, 550, 80, doi: 10.1038/nature23908

Kasen, D. 2010, \textbackslashapj, 708, 1025, doi: 10.1088/0004-637X/708/2/1025

Khazov, D., Yaron, O., Gal-Yam, A., et al. 2016, \textbackslashapj, 818, 3, doi: 10.3847/0004-637X/818/1/3

Kingma, D. P., & Ba, J. 2017, arXiv:1412.6980 [cs]. http://arxiv.org/abs/1412.6980

Mahabal, A., Sheth, K., Gieseke, F., et al. 2017, in 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 1–8, doi: 10.1109/SSCI.2017.8280984

Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, Publications of the Astronomical Society of the Pacific, 131, 038002, doi: 10.1088/1538-3873/aaf3fa

Martn Abadi, Ashish Agarwal, Paul Barham, et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/

Martnez-Palomera, J., Frster, F., Protopapas, P., et al. 2018, The Astronomical Journal, 156, 186, doi: 10.3847/1538-3881/aadfd8

Masci, F. J., Laher, R. R., Rusholme, B., et al. 2018, Publications of the Astronomical Society of the Pacific, 131, 018003, doi: 10.1088/1538-3873/aae8ac

Massaro, E., Maselli, A., Leto, C., et al. 2015, Astrophysics and Space Science, 357, 75, doi: 10.1007/s10509-015-2254-2

McKinney, W. 2010, Proceedings of the 9th Python in Science Conference, 56, doi: 10.25080/Majora-92bf1922-00a

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Mller, K.-R. 2019, in Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, ed. W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, & K.-R. Mller, Lecture Notes in Computer Science (Cham: Springer International Publishing), 193–209, doi: 10.1007/978-3-030-28954-6_10

Moriya, T., Tominaga, N., Blinnikov, S. I., Baklanov, P. V., & Sorokina, E. I. 2011, \textbackslashmnras, 415, 199, doi: 10.1111/j.1365-2966.2011.18689.x

Morozova, V., Piro, A. L., & Valenti, S. 2017, \textbackslashapj, 838, 28, doi: 10.3847/1538-4357/aa6251

Mowlavi, N., Lecoeur-Tabi, I., Lebzelter, T., et al. 2018, Astronomy & Astrophysics, 618, A58, doi: 10.1051/0004-6361/201833366

Muthukrishna, D., Parkinson, D., & Tucker, B. E. 2019, The Astrophysical Journal, 885, 85, doi: 10.3847/1538-4357/ab48f4

Nair, V., & Hinton, G. E. 2010, in Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10 (Haifa, Israel: Omnipress), 807–814

Narayan, G., Zaidi, T., Soraisam, M. D., et al. 2018, The Astrophysical Journal Supplement Series, 236, 9, doi: 10.3847/1538-4365/aab781

Naul, B., Bloom, J. S., Prez, F., & van der Walt, S. 2018, Nature Astronomy, 2, 151, doi: 10.1038/s41550-017-0321-z

Noebauer, U. M., Kromer, M., Taubenberger, S., et al. 2017, \textbackslashmnras, 472, 2787, doi: 10.1093/mnras/stx2093

Nordin, J., Brinnel, V., Santen, J. v., et al. 2019, Astronomy & Astrophysics, 631, A147, doi: 10.1051/0004-6361/201935634

Nugent, P. E., Sullivan, M., Cenko, S. B., et al. 2011, \textbackslashnat, 480, 344, doi: 10.1038/nature10644

Oh, K., Yi, S. K., Schawinski, K., et al. 2015, The Astrophysical Journal Supplement Series, 219, 1, doi: 10.1088/0067-0049/219/1/1

Palaversa, L., Ivezić, Ž., Eyer, L., et al. 2013, The Astronomical Journal, 146, 101, doi: 10.1088/0004-6256/146/4/101

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825. http://jmlr.org/papers/v12/pedregosa11a.html

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., & Hinton, G. 2017, arXiv:1701.06548 [cs]. http://arxiv.org/abs/1701.06548

Pichara, K., Protopapas, P., & Len, D. 2016, The Astrophysical Journal, 819, 18, doi: 10.3847/0004-637X/819/1/18

Piro, A. L., & Morozova, V. S. 2016, \textbackslashapj, 826, 96, doi: 10.3847/0004-637X/826/1/96

Piro, A. L., & Nakar, E. 2013, \textbackslashapj, 769, 67, doi: 10.1088/0004-637X/769/1/67

Prez-Carrasco, M., Cabrera-Vives, G., Martinez-Marin, M., et al. 2019, Publications of the Astronomical Society of the Pacific, 131, 108002, doi: 10.1088/1538-3873/aaeeb4

Reyes, E., Estvez, P. A., Reyes, I., et al. 2018, in 2018 International Joint Conference on Neural Networks (IJCNN), 1–8, doi: 10.1109/IJCNN.2018.8489627

Richards, J. W., Starr, D. L., Butler, N. R., et al. 2011, The Astrophysical Journal, 733, 10, doi: 10.1088/0004-637X/733/1/10

Rimoldini, L., Holl, B., Audard, M., et al. 2019, Astronomy & Astrophysics, 625, A97, doi: 10.1051/0004-6361/201834616

Rocklin, M. 2015, Proceedings of the 14th Python in Science Conference, 126, doi: 10.25080/Majora-7b98e3ed-013

Smith, K. W., Williams, R. D., Young, D. R., et al. 2019, Research Notes of the AAS, 3, 26, doi: 10.3847/2515-5172/ab020f

Soumagnac, M. T., & Ofek, E. O. 2018, Publications of the Astronomical Society of the Pacific, 130, 075002, doi: 10.1088/1538-3873/aac410

Steer, I., Madore, B. F., Mazzarella, J. M., et al. 2016, The Astronomical Journal, 153, 37, doi: 10.3847/1538-3881/153/1/37

Stetson, P. B. 1987, Publications of the Astronomical Society of the Pacific, 99, 191, doi: 10.1086/131977

SnchezSez et al., P. 2020, \apj

Tanaka, M., Tominaga, N., Morokuma, T., et al. 2016, The Astrophysical Journal, 819, 5, doi: 10.3847/0004-637X/819/1/5

Tominaga, N., Morokuma, T., Blinnikov, S. I., et al. 2011, \textbackslashapjs, 193, 20, doi: 10.1088/0067-0049/193/1/20

Tonry, J. L., Denneau, L., Heinze, A. N., et al. 2018, Publications of the Astronomical Society of the Pacific, 130, 064505, doi: 10.1088/1538-3873/aabadf

Turpin, D., Ganet, M., Antier, S., et al. 2020, arXiv:2001.03424 [astro-ph]. http://arxiv.org/abs/2001.03424

van Velzen, S., Gezari, S., Hammerstein, E., et al. 2020, arXiv:2001.01409 [astro-ph]. http://arxiv.org/abs/2001.01409

Wenger, M., Ochsenbein, F., Egret, D., et al. 2000, Astronomy and Astrophysics Supplement Series, 143, 9, doi: 10.1051/aas:2000332

Yaron, O., Perley, D. A., Gal-Yam, A., et al. 2017, Nature Physics, 13, 510, doi: 10.1038/nphys4025

Zeiler, M. D., & Fergus, R. 2014, in Computer Vision ECCV 2014, ed. D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars, Lecture Notes in Computer Science (Cham: Springer International Publishing), 818–833, doi: 10.1007/978-3-319-10590-1_53