Detecting optical transients using artificial neural networks and reference images from different surveys

Katarzyna Wardęga  
*The University of Texas Rio Grande Valley*

Adam Zadrozny  
*The University of Texas Rio Grande Valley*

Martin Beroiz  
*The University of Texas Rio Grande Valley*

Richard S. Camuccio  
*The University of Texas Rio Grande Valley*

Mario C. Díaz  
*The University of Texas Rio Grande Valley*

Follow this and additional works at: [https://scholarworks.utrgv.edu/pa_fac](https://scholarworks.utrgv.edu/pa_fac)

Part of the Astrophysics and Astronomy Commons, and the Physics Commons

Recommended Citation
Katarzyna Wardęga, Adam Zadrozny, Martin Beroiz, Richard Camuccio, Mario C Díaz, Detecting optical transients using artificial neural networks and reference images from different surveys, Monthly Notices of the Royal Astronomical Society, Volume 507, Issue 2, October 2021, Pages 1836–1846, [https://doi.org/10.1093/mnras/stab2163](https://doi.org/10.1093/mnras/stab2163)

This Article is brought to you for free and open access by the College of Sciences at ScholarWorks @ UTRGV. It has been accepted for inclusion in Physics and Astronomy Faculty Publications and Presentations by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact justin.white@utrgv.edu, william.flores01@utrgv.edu.
Detecting optical transients using artificial neural networks and reference images from different surveys

Katarzyna Wardęga, Adam Zadrożny, Martin Beroiz, Richard Camuccio and Mario C. Díaz

1 Center for Gravitational Wave Astronomy, The University of Texas Rio Grande Valley, Brownsville, TX 78520, USA
2 Faculty of Physics, University of Warsaw, PL-02-093 Warszawa, Poland
3 National Centre for Nuclear Research, Astrophysics Division, PL-02-093 Warszawa, Poland
4 Department of Physics and Astronomy, Texas Tech University, Lubbock, TX 79409, USA

ABSTRACT

We present a technique to detect optical transients based on an artificial neural networks method. We describe the architecture of two networks capable of comparing images of the same part of the sky taken by different telescopes. One image corresponds to the epoch in which a potential transient could exist; the other is a reference image of an earlier epoch. We use data obtained by the Dr. Cristina V. Torres Memorial Astronomical Observatory and archival reference images from the Sloan Digital Sky Survey. We trained a convolutional neural network and a dense layer network on simulated source samples and then tested the trained networks on samples created from real image data. Autonomous detection methods replace the standard process of detecting transients, which is normally achieved by source extraction of a difference image followed by human inspection of the detected candidates. Replacing the human inspection component with an entirely autonomous method would allow for a rapid and automatic follow-up of interesting targets of opportunity. The toy-model pipeline that we present here is not yet able to replace human inspection, but it might provide useful hints to identify potential candidates. The method will be further expanded and tested on telescopes participating in the Transient Optical Robotic Observatory of the South Collaboration.

Key words: gravitational waves – methods: data analysis – techniques: image processing – telescopes.

1 INTRODUCTION

The primary goal of this paper is to describe a method for detecting transients by comparing two images of the same region of the sky taken at different times and by different telescopes. The method is based purely on machine learning (ML) algorithms, specifically artificial neural networks (ANNs). The ML approach to transient detection is efficient as it can search through a large data set in a short amount of time. Hence, the ML methods are a valuable approach to solving the problem of detecting OTs in the time domain.

Difference image analysis (DIA) is the standard method used to search for OTs. DIA methods are based on subtracting a reference image from a target image. The method attempts to compensate for the difference in point spread functions (PSFs) of each image. Compensating for differences in PSF allows one to subtract images taken by different telescopes or under varying atmospheric conditions. However, even with PSF compensation, the resulting image difference might include some left behind residual flux that can be confused by detection algorithms as false OTs. Many DIA algorithms have been proposed since the original Phillips & Davis (1995) paper, notably those by Alard & Lupton (1998), Bramich (2008), and Zackay, Ofek & Gal-Yam (2016). Modern methods like Zackay et al. (2016) are hypothesis testing that a transient exists compared with the null hypothesis that a transient does not exist, results in a subtraction-like operation.

Regardless of the DIA method used, it is customary to train ML agents (e.g. random forest algorithms or neural networks) to sift through all the OT candidates, remove the spurious subtraction artifacts (‘bogus sources’), and retain the likeliest true OT candidates (Díaz et al. 2016; Klencki & Wyrzykowski 2016; Masci et al. 2016; Duev et al. 2019; Artola et al. 2020). A real/bogus classifier can be avoided if there is a manual operator, but it becomes cumbersome for large surveys where bogus sources can outnumber potential real ones by 100 to 1. For these reasons, it is possible that most systematic searches of the sky, like those for GW optical counterparts, will require a real/bogus classifier at the end of the analysis pipeline.

ML methods seem like a convenient element in the search for OTs. There could be many ways to use them. One of it presented by Sedaghat & Mahabal (2018) was to use ML for doing image subtraction, using images from the same survey, in search for OTs. In their method, they trained the network to produce an image with expected transient as a results. We present a different approach. We use images from different surveys but also, like in Sedaghat & Mahabal (2018), we train ML algorithms on the images directly, but our ML method produces classification as a result. In our method, an ML classifier takes two small image insets which are cropped around a detected source on the target image. One inset contains the detected source and the other is cropped around (rectangle 21 × 21
pixels) the same location on the reference image. When the source appears on the target inset, but is missing on the reference inset, the classifier calls the case an OT. When there is a source present on both insets, the classifier calls the case a non-OT. Providing the classifier with a sufficient number of example OT and non-OT cases will train it to be robust at detecting all true OTs on subsequent imaging runs. In our approach, we used ANN to compensate on registration and scintillation noise. For the network to fully compensate scintillation and registration noise, we need to create a test set of images taken at different conditions and times. In the case of the test set on Section 2, the images were taken on two different nights.

Bypassing DIA has several advantages. The neural network method we propose is robust against PSF variations across different surveys and filters. Observatories lacking an extensive reference archive could benefit from this method regardless of the references used, as long as the sky region is covered by some comparable photometric survey. Since it is typical for DIA methods to be computationally expensive, avoiding them leads to a drastic reduction in the processing speed for pipelines and an overall simplification in their design.

To test the feasibility of our proposed method, we built and trained two ANN models—one is a convolutional neural network (CNN) and the other is a dense layer network (DLN). The models accept target-reference inset pair samples as input and return the likelihood of the other being an OT as output. We trained the models on simulated data and calculated which prediction the simulations gave on test reference inset pair samples as input and return the likelihood of the other being an OT as output. For the network to fully compensate scintillation and registration noise, we need to create a test set of images taken at different conditions and times. In the case of the test set on Section 2, the images were taken on two different nights.

### 2 Method

To test our proposed method, we ran an experiment to prove the validity of our assumptions. The experiment consists on testing two different approaches to ML architectures based on ANNs. Then after training and validating them with simulated data, test them on pairs of real images. One member of the pair is from the CTMO and the reference member of the pair comes from the SDSS survey.

This section is organized as follows. In Section 2.1, we describe what kind of images from CTMO were used and how were downloaded and aligned equivalent images from SDSS. In Section 2.2, we present how we created the testing and training data set. In Section 2.3, we describe the architecture of the networks. Finally in Section 4, we present the results of the final metric values for our experiment.

In Section 3, we make another similar experiment with a set of images that has been analysed before in search for optical transients using a DIA method (Artola et al. 2020) with a Random Forest real/bogus classifier and also with a CNN-based real/bogus classifier. This second experiment allows for a more direct comparison of the method proposed here and the more conventional one based on DIA followed before.

#### 2.1 Image pre-processing

We targeted five galaxies (Table 1) covered by SDSS using the instrumentation of CTMO. Four of the five targets were taken on 2020 February 8 with the current optical configuration of CTMO, which consists of a PlaneWave Corrected Dall-Kirkham 17 arcsec astrograph with a ProLine 16803 CCD camera. Each image is unfiltered, has 60-s exposure time, taken at 2 × 2 binning, and has an FOV of 80 × 80 arcmin. We observed the fifth target, IC 4559, at an earlier date 2019 July 7 when CTMO had a different optical setup: The instrument used for these data was an Apogee F16M CCD camera. This image is unfiltered, taken at 2 × 2 binning with 300-s exposure time, and has a FOV of 50 × 50 arcmin.

We used the CTMO Analysis Library (CAL) to bias- and dark-subtract, as well as flatfield-correct, each image (Camuccio 2020). We used two-dimensional spatially varying mesh to subtract the median background of each image. Since each target consisted of a series of exposures, we plate-solved each image and aligned them per series using their world coordinate system (WCS) header metadata. We created a median-combined stack of the aligned images per series (Note: approximate limiting magnitude SDSS is 13). We used the SKYVIEW function from the ASTROQUERY (Ginsburg et al. 2019) package to download reference images from SDSS. Knowing the centre coordinates and FOV of each CTMO image, we requested the SDSS reference in the g filter with a size of 2000 × 2000 pixels. All SDSS images are taken from Data Release 9 (DR9) and have an exposure time of 54 s. We expect each image in a given pair to have different orientations. For an effective alignment solution, each pixel per picture should represent the same astronomical coordinates. To achieve image alignment, CAL employs the REPROJECT package from ASTROPY. The REPROJECT package aligns the SDSS image with the CTMO image and crops it to have the same FOV.

#### 2.2 Creating data sets

We anticipate transient events to look like new stellar sources in the sky. We wanted to construct ML methods so that they would recognize new sources in both follow-up observations and previously observed fields. Using the entire image as input to the neural network proved burdensome. Therefore, we created a data set with smaller images—the data set is composed of cropped images for each source detected on the images. In this paper, we focused only on simple cases when transients occur as a single source in the sky as we wanted to test toy-model first. The next step is creating simulations on top of galaxies, which we would like to cover in the next paper.

We created a data set of 3370 samples from five CTMO-SDSS image pairs (hereafter the ‘test data set’). Half of the samples were transients and the other half were non-transients. To train any ML model, one requires many samples (>10 000). For this reason, we simulated a data set for the training component (methodology of creating a simulation data sets is described in 2.2.2).

##### 2.2.1 Test data set

We postulate that source extraction programmes could find transient events based on the assumption that they would look like stellar sources. We built transient and non-transient samples from CTMO and SDSS source sub-images. Non-transient samples are a pair of sub-images with the same detected source—one from CTMO and the other is a dense layer network (DLN). The models accept target-reference inset pair samples as input and return the likelihood of the other being an OT as output.
other from SDSS. Transient samples are a pair of sub-images, one from CTMO containing a source, the other SDSS images containing no source – only background.

First, we detected sources on the CTMO image. We used the Source Extraction and Photometry (SEP) library in PYTHON (Bertin & Arnouts 1996; Barbary 2016). The programme detects objects from each image (in this study at 3σ confidence) and provides each of their coordinates as provided by the WCS header solution.

After source extraction, we normalized both images to a common signal level. Each image pair was taken with different instruments, so the first step was to quantify the difference in signal. CTMO images exhibit a much higher resolution than the SDSS ones. The increased depth CTMO images, is possibly due to the their unfiltered nature, whereas the SDSS images were obtained through a g band filter.

We made sub-images containing a single object from the list of detected sources on each CTMO image. An entire CTMO image is 2048 × 2048 pixels and each sub-image was 21 × 21 pixels centered on the coordinates of the detected source. Similarly, we made cuts of the aligned SDSS image at each detected source position, giving a pair of cropped images (one from CTMO and one from SDSS showing the same part of the sky). A few examples non-transient samples are shown in Fig. 1.

We did not observe any transients on these images, so we created artificial transient samples. We produced a sub-image containing a single object from the CTMO image (in the way described in the previous paragraph) and chose a spot on the SDSS image where there was only background. A few examples of these transient samples are shown in Fig. 2.

2.2.2 Simulated data set

For the training set, we simulated point sources superimposed on a mean background with noise. We fit the parameters of the programme to obtain samples that are similar to the samples in the test data set. We set the sample size as an image of 21 × 21 pixels. The background variance is generated from a normal distribution with a fixed mean level of zero analog-to-digital units (ADU) and a standard deviation of 0.5 ADU. The profile for each point source is a two-dimensional Gaussian distribution, with different sigma (σ) on the two main axes, and an arbitrary rotation with respect to the (x, y) pixel axes of the image. σ for the major and minor axes are chosen randomly from a uniform distribution. The orientation of the Gaussian profile with respect to the image axis is also selected uniformly over the unit circle. Simulated data were created using algorithms, human inspection was only involved on testing, if algorithms were working properly. However, even this was leaving some sort of bias depending on human preparing data set, since people might be sensitive to different S/N levels, which might influenced preparation of the data set.

For the source simulation, it is important to decide which image corresponds to the CTMO and SDSS images. The CTMO sources are brighter and larger in size. We set the amplitude of the brighter source to (35 ± 10) ADU and σ to (5 ± 1.5) pixels. For the dimmer source, we set the amplitude to (5 ± 15) ADU and σ to (0.5 ± 1.5) pixels.

A sample consists of a pair of small images and labels indicating whether the pair is a transient (label is "1") or not (label is "0"). For non-transient samples, both images contain a simulated object. One image simulates a source from CTMO image, while the other simulates a source from SDSS image. For transient samples, only one image contains a simulated object, whereas the other contains only simulated background. During the simulation, we chose the likelihood of generating a point source on the background to be 0.5, meaning that 50 per cent of the samples are transient samples while the rest are non-transient samples. Examples of simulated transient and non-transient samples are shown in Figs 3 and 4.

2.3 Building the neural network models

We built two different models to identify the existence of transients in the analysed images. One model uses convolutional layers, which are particularly useful for image analysis (Cun et al. 1990; LeCun et al. 1998; the CNN model). The other model uses dense layers,
Using ANNs to detect OTs comparing surveys

Figure 3. Simulated transient samples.

Figure 4. Simulated non-transient samples.

which are the basic structure of ANNs (McCulloch & Pitts 1943; the DLN model). The training process in ML requires fitting a large quantity of free parameters to the model and therefore a large amount of training sample data. Since data containing real transients are scarce, we used simulated samples in the training phase and data collected from real images in a final testing phase. The performance measures that we report are from the testing phase. We tested how both models predicted the existence of transients using test image data from CTMO and reference images from SDSS. To download and analyse SDSS images, we used the ASTROPY package (Robitaille et al. 2013; Price-Whelan et al. 2018). We explain how we generated the training samples and the testing samples in Chapter 2.2.

We created two types of networks with different topologies – one a CNN and the other a DLN. We trained both networks on the simulated data set and tested them on the test data set. We used the Keras library (Chollet 2015) with TensorFlow backend (Abadi et al. 2015) to construct the models and scikit-learn libraries (Pedregosa et al. 2011) to evaluate prediction of the models.

2.3.1 Convolutional model with single multilayer input

We built and tested the first model using convolutional layers, hence it is considered a convolutional model. For this task, we built the network using the sequential model in Keras. As input the model takes one image with two channels – one channel accepts the CTMO image and the other accepts the SDSS image. The model is a binary classifier – as an output it returns either “1” (a transient sample) or “0” (a non-transient sample). The network structure is shown in Fig. 5. The number of parameters in each layer and additional properties like the activation function are shown in Table 2. The total number of parameters of the CNN is 1475.

2.3.2 Dense model with double input

In the second model, we use primarily dense layers. As input, the model takes two images separately and then combines them. We built network using functional model in Keras. The structure of the network is shown in Fig. 6. The number of parameters in each layer and some additional properties are shown in Table 3. The total number of parameters of this model is 37 594, considerably more than the previous model.

2.4 Validation and test metrics

We trained both networks using 10 000 samples of simulated data. We split the samples into two subset: 8000 samples to train the network and 2000 samples to validate the results. We trained the CNN

Figure 5. Schema of the CNN model.

Figure 6. Schema of the DLN model.
Table 2. A summary of the CNN model parameters.

| Layer                        | Number of parameters | Properties          |
|------------------------------|----------------------|---------------------|
| Convolutional2D              | 190                  | AF = RELU           |
| Convolutional2D              | 455                  | AF = RELU           |
| MaxPooling2D                 | 0                    | pool size = (3, 3)  |
| Dropout                      | 0                    | 0.25                |
| Convolutional2D              | 138                  | AF = RELU           |
| MaxPooling2D                 | 0                    | pool size = (2, 2)  |
| Flatten                      | 0                    |                     |
| Dense                        | 40                   | AF = RELU           |
| Dropout                      | 0                    | 0.5                 |
| Dense                        | 550                  | AF = RELU           |
| Dropout                      | 0                    | 0.3                 |
| Dense                        | 102                  | AF = SOFTMAX        |

AF stands for ‘activation function’. The RELU function applies a rectified linear unit activation function. The SOFTMAX function converts a real vector to a vector of categorical probabilities.

Figure 6. Schema of the DLN model.

Table 3. A summary of the DLN model parameters.

| Layer                        | Number of parameters | Properties          |
|------------------------------|----------------------|---------------------|
| Input layer 1                | 0                    |                     |
| Input layer 2                | 0                    |                     |
| Dense input1                 | 1408                 | 64, AF = RELU       |
| Dense input2                 | 1408                 | 64, AF = RELU       |
| Dense input1                 | 2080                 | 32, AF = RELU       |
| Dense input2                 | 2080                 | 32, AF = RELU       |
| Dense input1                 | 264                  | 8, AF = RELU        |
| Dense input2                 | 264                  | 8, AF = RELU        |
| Dense input1                 | 36                   | 4, AF = RELU        |
| Dense input2                 | 36                   | 4, AF = RELU        |
| Concatenate                  | 0                    |                     |
| Flatten                      | 0                    |                     |
| Dense                        | 21632                | 128, AF = RELU      |
| Dense                        | 8256                 | 64, AF = RELU       |
| Dense                        | 130                  | 2, AF = SOFTMAX     |

Table 4. Metrics of the CNN and DLN models.

| Metric          | CNN model score | DLN model score |
|-----------------|-----------------|-----------------|
| Accuracy        | 0.989           | 0.969           |
| Precision       | 0.981           | 0.949           |
| Recall          | 0.996           | 0.99            |
| F1 score        | 0.989           | 0.97            |

Table 5. Confusion matrices of the CNN and DLN models.

|                  | Real / Classified | Non-transient | Transient |
|------------------|-------------------|---------------|-----------|
| 1-model          | Non-transient     | 1653          | 32        |
| (CNN)            | Transient         | 6             | 1679      |
| 2-model          | Non-transient     | 1595          | 90        |
| (DLN)            | Transient         | 13            | 1672      |

The test data consists of 1685 samples of transients and the same amount of non-transients. The CNN model mistakenly classified 32 non-transients as transients and only six transients as non-transients. The dense model made additional errors in non-transient classification. The errors might be caused by the sources having lower statistical significance in the SDSS images in comparison to the CTMO images, so there might be samples in which the SDSS source is of the same order of intensity as the background. The network cannot tell the difference between the dim source and the background, and thus misidentifies these samples as transients.

It is possible to avoid the mistake of false recognition by adding more lower signal-to-noise reference samples into the training data set. Another step could be changing the training data set altogether. If more CTMO data were available, it would be possible to create a training data set from real images in the same way like that for the test data set. Consequently, there would be no need to use simulation data.

Regardless, considering the two types of errors, it is preferable to have a non-transient event classified as a transient, not the opposite, because in this case one does not miss any potential transient event. Having a higher miss rate for transients would only cause additional checks for some non-transient cases. Classification error examples...
are shown in Figs 7 and 8. The most common error is produced when the SDSS source is weak. Another type of error is when the CTMO source is bright and large, when it nearly covers the entire sub-image. In one particular case, the network made an error when attempting to identify two sources in one sub-image.

Both models exhibit high accuracy. The accuracy is not 100 per cent in either case, which means that the networks are not overfitted. The CNN model demonstrated slightly better results than the DLN model, probably caused by the dense layers having many more parameters to train. The performance of the convolutional layers demonstrates that they are generally much better for image analysis. The next step of this project could be to build a model with a double input, such as the DLN model, but using convolutional layers rather than the dense layering.

3 COMPARISON OF DIA APPROACH AND ANN APPROACH ON DATA CONNECTED TO GW170104

In this section, we would like to present results of comparing DIA approach and ANN approach\(^1\) on search for of optical counterparts connected with GW170104. The initial search for astronomical transients was addressed by Artola et al. (2020). The authors analysed images taken by the TOROS Collaboration during the LIGO Scientific Collaboration’s second observation run (November 2016–August 2017) - O2. TOROS followed up three GW alerts of which two were truly astrophysical: GW170104 and GW170817. In this paper, we only analyse the GW170104 follow-up data. The data for GW170104 were taken by the Estacion Astronomica Bosque Alegre (EABA) in Cordoba, Argentina. TOROS observed the most massive galaxies within the high-probability region of localization for the GW events in 2017 January, and produced a reference set of the images of the same objects, retrieved later in 2017 November. The example of an image set looks like that shown in Figs 9 and 10.

The transient detection method used by Artola et al. (2020) involved DIA. The transform involves using a convolutional kernel to reduce the differences in PSFs on both images. The method used by the authors to find and apply the kernel was introduced by Bramich (2008). Following image transformation, the image is subtracted from the reference to reveal new sources. The DIA method generates a large number of spurious source artifacts (i.e. ‘bogus sources’). An ML algorithm is then used to distinguish between real and bogus sources.

The authors of Artola et al. (2020) generated synthetic ‘real’ sources to create a training set for teaching an ML algorithm to distinguish between real and bogus transients. The method involved repeatedly injecting the profile of a star into an image. Then, they subtracted the images and extracted sources to detect objects on the difference image. Some detected sources were injected objects (i.e. 

\(^1\)In this, a reference image to compare was taken by the same telescope, not an image taken by SDSS.
‘real’ transients) and the rest were subtraction artifacts (i.e. ‘bogus’ transients) left from subtraction process implemented according to Bramich (2008). Having samples of real and bogus transients, the authors built and trained a random forest, decision trees, and a support-vector machine – the best results were obtained by random forest.

Although the problem addressed by Artola et al. (2020) is similar to the one addressed in this paper, the methods are quite different in nature. Models based on DIA distinguish between real and bogus sources collected from a single, difference image. Our method bypasses the subtraction step and, instead, works directly on the target-reference pair of images by focusing on one source at a time and identifying it as a transient or non-transient. Additionally, DIA methods require examples of real and bogus transients to train ML algorithms, while our method requires examples of transients (equivalent to reals) and non-transients. Nevertheless, to compare both methods, we applied the algorithm to the same data used by Artola et al. (2020).

We created the training and testing data as follows. We extracted all samples for the test data sets from the original 13 images taken during the GW170104 follow-up event as described in Artola et al. (2020). The transient samples have a source, with PSF profile of stars, visible on one image and the background on the other image – they are equivalent to the set of ‘real’ transients in the DIA method. The non-transient samples are a pair of thumbnails of the same objects detected by SExtractor with $3\sigma$ threshold in target and reference images. The comparison data set has a total of 3557 samples with labels. An example of transient and non-transient samples is shown in Fig. 11.

We retrained the models with different input sizes matching the conditions of Artola et al. (2020). We simulated a new training data set and we adjusted the background noise level and standard deviation of the simulated training samples to 0 and 2.5, respectively, to match those of the test set. Furthermore, we set the amplitude of the simulated sources to an average of 3 ADU and a standard deviation of 10 ADU. The sources are shaped like Gaussian profiles with a $\sigma$ value of $(30 \pm 10.5)$ pixels. In this case, we simulated both sources with the same parameters, creating a total of 10 000 samples. Examples of simulated transient and non-transient samples are shown in Fig. 12.

The number of parameters to train is different than analysis presented in Chapter 2, because the size of the sub-image in one sample is bigger ($43 \times 43$ pixels). The total number of parameters of the CNN model is 2195 and for the DLN model is 62938 – a significant difference. The number of parameters in each model and some additional properties are shown in Tables 6 and 7.

The confusion matrix for these models is shown in Table 9. The main error is, again, in the non-transient classification. In Table 8, we compare the metrics for three models: the two networks and the random forest (RF) algorithm tested in Artola et al. (2020). We cannot treat this comparison as entirely accurate, because the
classification problems are inherently different. Regardless, the DLN model obtained the overall best results.

The main advantage of our method is that it combines subtraction step with real/bogus classification. Because of this, our method can compare images which are significantly different (e.g. taken by different instruments), meaning optical transient could be detected without taking a reference image hours or days later. Hence, our method allows us to detect optical transients with very low latency. Additionally, our method does not require additional classification between bogus and real transients.

### 4 CONCLUSION

We have shown that it is possible to detect OTs by comparing images from two different telescopes. Our paper presents a toy-model which combines DIA with post DIA ML-classification. We think this feature is especially useful in the fast detection of kilonovae during EM follow-up observations of GW events, and readily adoptable for small observatories to participate in these targets of opportunity.

We tested two neural network models – one based on CNNs and other based on dense layers. Our models achieved high accuracy (values are around 0.98 for both models for different S/N range, except for 5–10 S/N where accuracy is around 0.8). The main error in both networks was misidentifying non-transient samples as transients. A reason for false positive detection could be that both images are of different intensity scales (i.e. a given source might have different pixel intensities between target and reference image subsets). There are sample cases for which the object is much weaker in the SDSS image and, therefore, the network sees it as part of background.

We tested both models on data taken by the TOROS Collaboration in follow-up to the GW170104 event. Initially, in order to detect transients in these data, DIA was the primary method, followed by an ML inspection of source-extracted objects on the difference images to distinguish between transients and artifacts. With our method, the models classified whether or not the sample images contained a transient, and they achieved a high accuracy score: 0.91 for the CNN model 0.918 for the DLN model (RF score was 0.89). In this comparative study, the DLN model obtained the best results.

In order to expand this project, it would be useful to build other models with better efficiency. Models with convolutional layers contain less parameters and, hence, are much easier and quicker to train which will be useful in the analysis of larger images or data sets.

A next step could be to combine two different models, like a model with double input but using convolutional layers. Each of the model made different type of errors, hence merging features of both of them might eliminate most incorrect classifications (see: 4). Another idea for lowering number of error is to add more challenging samples into the training data set e.g. lower signal-to-noise reference samples. Final step could be changing the training data set altogether. If more CTMO data were available, it would be possible to create a training data set from real images in the same way like that for the test data set. Consequently, there would be no need to use simulation data.

The goal of this project is to apply these algorithms to TOROS data and incorporate them into the standard analysis pipeline. The first step is to test the method on TOROS data. We could test if the models detect the real kilonova observed by the TOROS Collaboration in follow-up to GW170817.

This work presents a little bit different approach to the problem of transient detection by combining ‘DIA’ and ML ‘real’ / bogus classification in a single step. What is more, it compares images from two different surveys. This allows to use an images from big surveys like SDSS as reference image in search for potential OTs.

---

**Table 7.** A summary of the DLN model for O2 data.

| Layer                 | Number of parameters | Properties |
|-----------------------|----------------------|------------|
| Input layer 1         | 0                    |            |
| Input layer 2         | 0                    |            |
| Dense input1          | 2816                 | 64, AF = RELU |
| Dense input2          | 2816                 | 64, AF = RELU |
| Dense input1          | 2080                 | 32, AF = RELU |
| Dense input2          | 2080                 | 32, AF = RELU |
| Dense input1          | 264                  | 8, AF = RELU  |
| Dense input2          | 264                  | 8, AF = RELU  |
| Dense input1          | 36                   | 4, AF = RELU  |
| Dense input2          | 36                   | 4, AF = RELU  |
| Concatenate           | 0                    |            |
| Flatten               | 0                    |            |
| Dense                 | 44 160               | 128, AF = RELU |
| Dense                 | 8256                 | 64, AF = RELU |
| Dense                 | 130                  | 2, AF = SOFTMAX |

**Table 8.** Metrics of the CNN model, DLN model, and RF algorithm for O2 data.

| Metric     | CNN model score | DLN model score | RF score |
|------------|-----------------|-----------------|----------|
| Accuracy   | 0.91            | 0.918           | 0.89     |
| Precision  | 0.856           | 0.866           | 0.92     |
| Recall     | 0.993           | 0.997           | 0.86     |
| F1 score   | 0.919           | 0.927           | 0.89     |

**Table 9.** Confusion matrices of the CNN model, DLN model, and RF algorithm for O2 data.

|               | Real/Classified | Non-transient | Transient |
|---------------|-----------------|---------------|-----------|
| Model 1       | Non-transient   | 1379          | 322       |
| (CNN)         | Transient       | 4             | 1842      |
| Model 2       | Non-transient   | 1403          | 308       |
| (Dense)       | Transient       | 12            | 1834      |

---

**ACKNOWLEDGEMENTS**

The TOROS collaboration acknowledges support from Ministerio de Ciencia, Tecnología e Innovación Productiva (MINCyT) and Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET) from Argentina, grants from the National Science Foundation of the United States of America, NSF PHYS 1156600 and NSF HRD 1242090, and the government of Salta province in Argentina.

AZ and National Centre for Nuclear Research are grateful for financial support from MNiSW grant DIR/WK/2018/12 and NCN grant UMO-2017/26/M/ST9/00978.

KW is grateful for scholarship from The University of Texas Rio Grande Valley for the academic year 2019–2020, during which this project was carried out as part of student exchange between University of Warsaw, Faculty of Physics and The University of Texas Rio Grande Valley, Faculty of Physics.
The authors thank Lucas Macri for his helpful comments to the manuscript.

Funding for SDSS-III has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, and the U.S. Department of Energy Office of Science. The SDSS-III web site is http://www.sdss3.org/.

SDSS-III is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS-III Collaboration including the University of Arizona, the Brazilian Participation Group, Brookhaven National Laboratory, Carnegie Mellon University, University of Florida, the French Participation Group, the German Participation Group, Harvard University, the Instituto de Astrofisica de Canarias, the Michigan State/Notre Dame/JINA Participation Group, Johns Hopkins University, Lawrence Berkeley National Laboratory, Max Planck Institute for Astrophysics, Max Planck Institute for Extraterrestrial Physics, New Mexico State University, New York University, Ohio State University, Pennsylvania State University, University of Portsmouth, Princeton University, the Spanish Participation Group, University of Tokyo, University of Utah, Vanderbilt University, University of Virginia, University of Washington, and Yale University.

DATA AVAILABILITY

The data underlying this article, collected by TOROS collaboration, will be shared on reasonable request to the corresponding author. SDSS data used in this paper is available from the public data archives of the SDSS.

REFERENCES

Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems https://www.tensorflow.org/

Alard C., Lupton R. H., 1998, ApJ, 503, 325

Artola R. et al., 2020, MNRAS, 493, 2207

Barbary K., 2016, J. Open Sour. Softw., 1, 58

Bertin E., Arnouts S., 1996, A&A, 117, 393

Bramich D. M., 2008, MNRAS, 386, L77

Camuccio R., 2020, Master’s thesis, Univ. Texas Rio Grande Valley

Chollet F., 2015, Keras.https://keras.io

Cun Y., LeCun Y., 1998, Adv. Neural Inf. Process. Syst., 2, 396

Díaz M. C. et al., 2016, ApJ, 828, 6

Duev D. A. et al., 2019, MNRAS, 489, 3582

Ginsburg A. et al., 2019, AJ, 157, 98

Gunn J. E. et al., 1998, AJ, 116, 3040

Gunn J. E. et al., 2006, AJ, 131, 2332

Klencki J., Wyżykowski L., 2016, in Różańska A., Bejger M., eds, Proc. PAS Vol. 3, Real-Time Detection of Transients in OGLE-IV with Application of Machine Learning. Adam Mickiewicz Univ., Poznań, p. 56

LeCun Y., Bottou L., Bengio Y., Haffner P., 1998, Proc. IEEE, 86, 2278

Masci F. J. et al., 2016, PASP, 129, 014002

Mcculloch W., Pitts W., 1943, Bull. Math. Biophysics., 5, 115

Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825

Phillips A. C., Davis L. E., 1995, Astron. Data Analysis Softw. Syst. IV, 77, 4

Price-Whealan A. M. et al., 2018, Apj, 156, 123

Robitaille T. P. et al., 2013, A&A, 558, A33

Sedaghat N., Mahabal A., 2018, MNRAS, 476, 5365

Zackay B., Ofek E. O., Gal-Yam A., 2016, Apj, 830, 27

APPENDIX A: VALIDATION AND TEST METRICS

In this appendix, we present a more detailed version on how simulations and S/N was calculated. We also present accuracy of our solution versus S/N.

We trained both networks using 10,000 samples of simulated data. We split the samples into two subset: 8000 samples to train the network and 2000 samples to validate the results. We trained the CNN and DLN models in 30 epochs using the Adam optimizer, and we evaluated the performance of the training with the accuracy metric.

Table A1. Metrics of the CNN and DLN models for S/N in range 0–5 calculated for 40 samples.

| Metric         | CNN model score | DLN model score |
|----------------|-----------------|-----------------|
| Accuracy       | 0.985 ± 0.0247  | 0.85 ± 0.0565   |
| Precision      | 1               | 0.85            |
| Recall         | 0.95            | 0.85            |
| F1 score       | 0.97            | 0.85            |

Table A2. Metrics of the CNN and DLN models for S/N in range 5–10 calculated for 34 samples.

| Metric         | CNN model score | DLN model score |
|----------------|-----------------|-----------------|
| Accuracy       | 0.8235 ± 0.0654 | 0.7941 ± 0.0693 |
| Precision      | 0.76            | 0.86            |
| Recall         | 0.94            | 0.71            |
| F1 score       | 0.84            | 0.77            |

Table A3. Metrics of the CNN and DLN models for S/N in range 10–15 calculated for 80 samples.

| Metric         | CNN model score | DLN model score |
|----------------|-----------------|-----------------|
| Accuracy       | 0.985 ± 0.0175  | 0.975 ± 0.0175  |
| Precision      | 0.95            | 0.95            |
| Recall         | 1               | 1               |
| F1 score       | 0.98            | 0.98            |

Table A4. Metrics of the CNN and DLN models for S/N in range 15–20 calculated for 398 samples.

| Metric         | CNN model score | DLN model score |
|----------------|-----------------|-----------------|
| Accuracy       | 0.9925 ± 0.0043 | 0.9573 ± 0.0101 |
| Precision      | 0.99            | 0.92            |
| Recall         | 1               | 1               |
| F1 score       | 0.99            | 0.96            |

Table A5. Metrics of the CNN and DLN models for S/N in range 20–30 for 1234 samples.

| Metric         | CNN model score | DLN model score |
|----------------|-----------------|-----------------|
| Accuracy       | 0.9918 ± 0.0026 | 0.9684 ± 0.0005 |
| Precision      | 0.99            | 0.94            |
| Recall         | 0.99            | 0.99            |
| F1 score       | 0.99            | 0.97            |
Using ANNs to detect OTs comparing surveys

Table A6. Metrics of the CNN and DLN models for S/N in range 30 and bigger for 1584 samples.

| Metric     | CNN model score    | DLN model score    |
|------------|--------------------|--------------------|
| Accuracy   | 0.9899 ± 0.0025    | 0.9798 ± 0.0035    |
| Precision  | 0.98               | 0.96               |
| Recall     | 0.99               | 0.99               |
| F1 score   | 0.99               | 0.98               |

Table A7. Confusion matrices of the CNN and DLN models for S/N in range 0–5 calculated for 40 samples.

| Real / Classified | Non-transient | Transient |
|-------------------|---------------|-----------|
| 1-model Non-transient | 20            | 0         |
| (CNN) Transient    | 1             | 19        |
| 2-model Non-transient | 17            | 3         |
| (DLN) Transient    | 5             | 12        |

Table A8. Confusion matrices of the CNN and DLN models for S/N in range 5–10 calculated for 34 samples.

| Real / Classified | Non-transient | Transient |
|-------------------|---------------|-----------|
| 1-model Non-transient | 12            | 5         |
| (CNN) Transient    | 1             | 16        |
| 2-model Non-transient | 15            | 2         |
| (DLN) Transient    | 5             | 12        |

Table A9. Confusion matrices of the CNN and DLN models for S/N in range 10–15 calculated for 80 samples.

| Real / Classified | Non-transient | Transient |
|-------------------|---------------|-----------|
| 1-model Non-transient | 38            | 2         |
| (CNN) Transient    | 0             | 40        |
| 2-model Non-transient | 38            | 2         |
| (DLN) Transient    | 0             | 40        |

Table A10. Confusion matrices of the CNN and DLN models for S/N in range 15–20 calculated for 398 samples.

| Real / Classified | Non-transient | Transient |
|-------------------|---------------|-----------|
| 1-model Non-transient | 196           | 3         |
| (CNN) Transient    | 0             | 199       |
| 2-model Non-transient | 182           | 17        |
| (DLN) Transient    | 0             | 199       |

Table A11. Confusion matrices of the CNN and DLN models for S/N in range 20–30 calculated for 1234 samples.

| Real / Classified | Non-transient | Transient |
|-------------------|---------------|-----------|
| 1-model Non-transient | 609           | 8         |
| (CNN) Transient    | 2             | 790       |
| 2-model Non-transient | 581           | 36        |
| (DLN) Transient    | 3             | 614       |

Table A12. Confusion matrices of the CNN and DLN models for S/N in range 30 and bigger calculated for 1584 samples.

| Real / Classified | Non-transient | Transient |
|-------------------|---------------|-----------|
| 1-model Non-transient | 778           | 14        |
| (CNN) Transient    | 2             | 790       |
| 2-model Non-transient | 762           | 30        |
| (DLN) Transient    | 2             | 790       |

Figure A1. Samples with moved SDSS source in which models made classification error.

The resulting accuracy reflects a compromise between achieving the best results and avoiding an overfitting of the network.

After training and validation, we calculated the prediction of each model for test data samples. The prediction output is the likelihood of the sample being a transient. A value of one means absolute confidence that the source is a transient, and a value of zero indicates a non-transient source.

To present the results in clearer way, we calculated S/N for all test samples using PHOTUTILS aperture package. We divided test data sets into subset based on different S/N ranges (calculated for CTMO images): from 0 to 5, 5 to 10, 10 to 15, 15 to 20, 20 to 30, and 30 and bigger. For each range, we calculated four metrics (accuracy, precision, recall, F1 score) and confusion matrix. The accuracy metric is given with bigger precision and with en error. The results are shown in Tables A1, A2, A3, A4, A5, and A6.

The confusion matrices are shown in Tables A7, A8, A9, A10, A11, and A12. The confusion matrix shows how many times the network makes an error and the type of error. The diagonal of the matrix contains the number of correctly classified samples per class, and the off-diagonal elements are the miss-classification for each class. For a two-class system, the off-diagonal elements are the errors of classifying a transient as a non-transient and vice versa.

The both models performed high accuracy. For most of samples metrics are around 0.98. The only range where performance is a little bit worse (around 0.8 for accuracy metric) is from 5 to 10 S/N – in this range, we had also the fewest samples. Confusion matrices show that both models made significantly more errors in non-transient classification. In addressed problem that is better results because network is less likely to miss any transient. The test data consists of 1685 samples of transients and the same amount of non-transients. The CNN model mistakenly classified 32 non-transients as transients and only 6 transients as non-transients. The dense model made 90 errors in non-transient classification and 13 in transient.
Analysis of S/N shows that for SDSS images, there are 609 samples where S/N is negative (value around −1). It is possible to obtain such results where there is no source in the image – only background. We noticed that there is also few samples where SDSS source is not in the middle of an image. This also causes negative S/N and leads to making errors by both models. The example of such error is showed in Fig. A1. CNN model classifies wrongly 9 such samples and DLN model 27.

Another most common error was in non-transient classification in cases when SDSS source is much weaker, only few ADU above background. The network cannot tell the difference between the dim source and the background, and thus misidentifies these samples as transients. This type of error was done mostly by CNN model. Fig. A2 shows the samples with weaker SDSS source.

In Fig. A3 are presented samples for which NN model made error in transient classification. In first on CTMO image, there are two sources and on second CTMO source is really bright. CNN model made errors in 6 samples in transient class, but all of them are results of wrongly labelled test data. Samples which are labelled as transient actually looked like non-transient. This means that CNN model classified them correctly and it does not make any error in transient class.

The interesting observation is the fact that both models made different kind of errors. There are only 14 samples which both networks classified wrongly (e.g. Fig. A4).

Regardless, considering the two types of errors, it is preferable to have a non-transient event classified as a transient, not the opposite, because in this case one does not miss any potential transient event. Having a higher miss rate for transients would only cause additional checks for some non-transient cases. Classification error examples are shown in Figs A1, A2, A3, and A4.

Both models exhibit high accuracy. The accuracy is not 100 per cent in either case, which means that the networks are not overfitted. The CNN model demonstrated slightly better results than the DLN model, probably caused by the dense layers having many more parameters to train. The performance of the convolutional layers demonstrates that they are generally much better for image analysis.

This paper has been typeset from a TeX/LaTeX file prepared by the author.