The Astronomical Journal, 166:115 (29pp), 2023 September https://doi.org/10.3847/1538-3881/ace9d8

What's the Difference? The Potential for Convolutional Neural Networks for Transient Detection without Template Subtraction

Tatiana Acero-Cuellar[1,2], Federica Bianco[1,3,4,5], Gregory Dobler[1,3,4], Masao Sako[6], and Helen Qu[6]
The LSST Dark Energy Science Collaboration

[1] University of Delaware, Department of Physics and Astronomy, 217 Sharp Lab, Newark, DE 19716, USA
[2] Universidad Nacional de Colombia, Observatorio Astronómio Nacional, Bogotá, Colombia
[3] University of Delaware, Joseph R. Biden, Jr. School of Public Policy and Administration, 184 Academy Street, Newark, DE 19716, USA
[4] University of Delaware, Data Science Institute, Newark, DE 19716, USA
[5] Vera C. Rubin Observatory, Tucson, AZ 85719, USA
[6] Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA

Received 2022 March 22; revised 2023 June 28; accepted 2023 July 19; published 2023 August 18

Abstract

We present a study of the potential for convolutional neural networks (CNNs) to enable separation of astrophysical transients from image artifacts, a task known as "real–bogus" classification, without requiring a template-subtracted (or difference) image, which requires a computationally expensive process to generate, involving image matching on small spatial scales in large volumes of data. Using data from the Dark Energy Survey, we explore the use of CNNs to (1) automate the real–bogus classification and (2) reduce the computational costs of transient discovery. We compare the efficiency of two CNNs with similar architectures, one that uses "image triplets" (templates, search, and difference image) and one that takes as input the template and search only. We measure the decrease in efficiency associated with the loss of information in input, finding that the testing accuracy is reduced from ∼96% to ∼91.1%. We further investigate how the latter model learns the required information from the template and search by exploring the saliency maps. Our work (1) confirms that CNNs are excellent models for real–bogus classification that rely exclusively on the imaging data and require no feature engineering task and (2) demonstrates that high-accuracy (>90%) models can be built without the need to construct difference images, but some accuracy is lost. Because, once trained, neural networks can generate predictions at minimal computational costs, we argue that future implementations of this methodology could dramatically reduce the computational costs in the detection of transients in synoptic surveys like Rubin Observatory's Legacy Survey of Space and Time by bypassing the difference image analysis entirely.

Unified Astronomy Thesaurus concepts: Astronomical methods (1043); Convolutional neural networks (1938); Transient detection (1957)

1. Introduction

Modern observational astronomy has shown us that the Universe is not static and immutable; on the contrary, it is a lively and dynamic system. We now know and understand a variety of different phenomena that can give rise to variations in the brightness and color of astrophysical objects, including explosive stellar death (supernovae, kilonovae, gamma-ray bursts, etc.), less dramatic and powerful stellar variability (flares, pulsations), and variability arising from geometric effects (such as planetary transients and microlensing). The timescales for these phenomena range from seconds to years. When the variations are terminal like supernovae (SNe) or stochastic like stellar flares, they are often referred to as "transients." Detecting optical astrophysical transients characteristically requires sequences of images across a significant temporal baseline. Surveys designed to study the ever-changing skies, like the Dark Energy Survey (DES; The Dark Energy Survey Collaboration 2005), the future Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019), and many others, search for spatially localized changes in brightness in previously observed patches of sky.

Due to the rarity of astrophysical transients, a long baseline in conjunction with the observation of a large area of the sky is typically required to detect a statistically significant sample of transients like SNe and individual examples of rare events, such as kilonovae. The process itself is arduous and requires considerable human intervention at multiple stages. The first step is typically the creation of high-quality templates of each region of the sky that is to be searched; the templates are then subtracted from nightly images, a process known in astrophysics as difference image analysis (DIA) that was initially pioneered by Crotts (1992) and Tomaney & Crotts (1996) and then formalized by Alard & Lupton (1998), and there is a rich history of subsequent improvements in the efficiency and accuracy of DIA models. Templates (tmpl) are typically constructed as stacks of high-quality (favorable observing conditions) sky images. These high-quality images must then be aligned with the "today" image, typically called the "search image" (srch), and degraded to match its point-spread function (PSF; which we note may vary across a single large field-of-view image) and scaled to match its brightness. The product generated by the subtraction of the templ from the srch image is the so-called "difference image" (diff; see, for example, a description of the DIA processing pipeline for DES in Kessler et al. 2015). Transients can then be detected as clusters of adjacent pixels deviating significantly from the background. However, even the best existing DIA algorithms produce diff images with large pixel value deviations from the ideal zero average that would be expected if no changes occurred in a patch of sky. Transients, variable stars, and moving objects will result in detections, but a large number of artifacts will also typically be detected by these thresholding schemes.

Machine-learning models offer an excellent opportunity to improve the efficiency of transient detection at this stage, automating the classification between "real" astrophysical transients and "bogus" artifacts; these models are often referred to as "real–bogus" (RB). Generally, the applications of machine learning to this problem are based on the extraction of features from the (diff) images that are then fed to models like random forests (RFs), *k*-nearest neighbors, or support vector machines (Goldstein et al. 2015; Sánchez et al. 2019; Mong et al. 2020). These models have achieved high accuracy and enabled the discovery of transients at scale in larger synoptic surveys, for example, in the Palomar Transient Factory (Bloom et al. 2012) and Zwicky Transient Facility (ZTF; Mahabal et al. 2019).

The process of engineering features for machine-learning models allows the experts to embed domain knowledge in the models. In the case of RB classification, the features engineered to be used by models such as the ones described above usually rely on visual inspection of a subset of the data. However, these features may overlook abstract associations between image properties that can be effective for classification and, in fact, may be biased toward human perception and theoretical expectations. An alternative approach involves the use of models that can learn features directly from the data, such as convolutional neural networks (CNNs; LeCun et al. 1989). Here it is possible to train a model using the images themselves, skipping the step of feature design entirely.

In the literature, RB CNN-based models appear as early as 2016 (Cabrera-Vives et al. 2016), and they are typically based on the analysis of the diff images arising from the DIA. While neural networks may be computationally demanding in the training phase, the "feed-forward" classification that arises from a pretrained model is typically rapid and computationally light,[7] leaving DIA as the computational bottleneck in the process of astrophysical transients' detection. Yet, in principle, the entirety of the information content embedded in the diff–tmpl–srch image triplet is also contained in the tmpl–srch image pair. In this paper, we explore the use of CNNs as RB models, concentrating on the potential for building high-accuracy models that do not require the construction of diff images.

This paper represents a first, critical step in the process of conceptualizing and realizing a DIA-free model for astrophysical transients. Here we investigate whether neural networks can discriminate between astrophysical transients and artifacts without a diff image, while still relying on DIA for detection. This is the necessary premise for the complete elimination of DIA, and estimating the impact of the information lost by dropping the diff will enable future work toward the development of models that will not rely on DIA for detection and will detect and characterize transients (measure magnitude) from the tmpl–srch image pair only.

This paper is organized as follows. In Section 2, we discuss the DIA history and methodology. In Section 3, we present the DES data that we used to build our RB classification models and the preprocessing steps. In Section 4, we discuss our methodology, illustrating the CNN architectures used and, in Section 4.4, presenting and discussing the use of saliency maps to gain insights into the models. In Section 5, we show the results of building a model that does not use the diff image as input, named *noDIA-based* model, and compare its performance to one that does, named *DIA-based* model; we outline future work and conclude with a discussion of the broader implications for this result in light of upcoming surveys in Section 6.

This study is reproducible, and all code that supports the analysis presented here is available in a dedicated `GitHub` repository.[8]

2. State-of-the-art and Traditional Solutions to Detection of Transients

2.1. Diff Imaging

Diff images are produced by subtracting a templ, an image generated by coadding multiple images (e.g., Kessler et al. 2015), from a sky image, and they are currently the basis for most astrophysical transient srch algorithms. The diff image allows brightness changes to be detected even if embedded in Galaxy light, for example, in the case of extragalactic explosive transients. Great efforts have been made to improve the quality and effectiveness of the diff images. Although the name may suggest that the process simply entails subtracting images from each other, the procedure is in fact riddled with complications for the following reasons. First, the images used to build the templ and srch images are principally taken within different atmospheric conditions (Zackay & Ofek 2017) generating variations in the quality of the images. The construction of a proper templ is also a delicate task; typically, templ are built by stacking tens of images taken under favorable sky conditions at different times. This improves the image quality but also mitigates issues related to variability in the astrophysical objects captured in the image (Hambleton et al. 2020); one wants to capture each variable source at its representative brightness. Typically, then, the templ image is of higher quality than the srch image, and it is degraded to match the srch image PSF and scaled to match its brightness. Yet the scaling and PSF may vary locally in the image plane, especially for images from large field-of-view synoptic surveys such as the 2.2 deg^2 DES or ∼10 deg^2 Rubin images. Finally, it is important that the images are perfectly aligned both in the creation of the templ and in the subtraction process to create the diff image. This implies accounting for rotation as well as potentially different warping effects on the images and templ. Once the PSF match and alignment are done, it is possible to subtract the degraded templ from the srch images to obtain the diff image. To degrade the image quality of the templ to match the srch image, a convolution kernel that must be applied to the templ needs to be determined (Alard & Lupton 1998),

$$\mathrm{tmpl}(x, y) \otimes \mathrm{Kernel}(u, v) = \mathrm{srch}(x, y), \tag{1}$$

where tmpl is the high-quality image, the templ image; srch is the one night image, or srch image; and $\otimes$ is the convolutional operation. The arguments x and y represent the coordinates of the pixel matrix that compose the images; u and v are the coordinates of the kernel matrix.

[7] Although the model may require a large amount of memory to cache.

[8] https://github.com/taceroc/DIA_noDIA

Table 1
Comparison of the Accuracy and Computational Cost for Training, Testing, and Validation Data for the *DIA-based* (Figure 4, left) and *noDIA* (Figure 4, right) Models

Model	Accuracy			Training[8] CPU (hr)	Prediction[8] Clock Time (ms)
	Train	Test	Val.		
DIA-based	0.965	0.961 ± 0.004	0.960	∼35	1.00 ± 0.03
noDIA	0.920	0.911 ± 0.005	0.914	∼56	0.30 ± 0.01

To solve the computationally expensive problem of matching PSFs, the kernel can be decomposed in terms of simple functions, for instance, Gaussian functions, and the method of least-squares can be used to determine the best values for the kernel. The fitted solution of one srch image can be determined in a short computational time. However, the computational cost scales with the image size and resolution. Surveys and telescopes constructed with the goal of discovering new transients are generally designed to collect tremendous amounts of data to maximize the event rate (detection of astrophysical transients). For the DES, the computational cost per 2.2 deg^2 image is ∼15.5 CPU hr (with roughly two-thirds of that time spent on PSF matching).[9] The upcoming Rubin LSST will collect more than 500 images every night, each with 3.2 gigapixels. This process is thus bound to turn out to be very expensive. In this paper, we train our models on postage stamps where transients were detected or simulated (see Section 3); thus, a direct comparison of the computational cost is not trivial. For comparison, a detailed discussion of the computational cost of our models is included in Section 5.2, and the CPU node hours required to train and generate predictions from our models are reported in Table 1. We note here that with a deep neural network (DNN) approach to this problem, the computational cost is high in training, but the predictions require minimal computational time.

A bad subtraction can occur because of poor PSF matching, alignment, or correction of image warping. In all of these cases, the subtraction would lead to artifacts or "bogus" alerts like the one shown in the diff image in Figures 1(a) and (b). In particular, in Figure 1(a), the diff image shows a so-called "dipole," where one side of a suspected transient is dark and the other side bright; this typically arises in the case of misalignments, but it might also be caused by moving objects in the field or differential chromatic refraction (Carrasco-Davis et al. 2021). Conversely, Figure 1(b) shows a bogus alert caused by an image artifact: a column of bad pixels in the srch image. At this location, there is no astrophysical object in the image thumbnail, no host galaxy or star that could give rise to variations.

Figures 1(c) and (d) show genuine transients in our DES training data; in both of these examples, there is a clear transient in the diff images (high pixel values at the center of the image).

2.2. Autoscan and Other Feature-based RB Models

We developed our models on data collected in the first year of DES. Thus, a direct precursor of our work is Goldstein et al. (2015), who created an automated RB based on an RF supervised-learning model (Ho 1995) to detect transients, particularly SNe, in the DES data, hereafter referred to as `autoscan`. For these kinds of models, the process of selecting and engineering features is pivotal. `Autoscan` is based on 38 features derived from the diff, srch, and tmpl images. The selection and computation of these features was done by attempting to quantitatively represent what humans would leverage in visual inspections. For instance, `r_aper_psf` distinguishes a bad subtraction of srch and tmpl that would lead to a diff qualitatively similar to Figure 1(a); the feature `diffsum` measures the significance of the detection by summing the pixel values in the center of the diff image; and the feature `colmeds`, indicating the CCD used for the detection, is designed to identify artifacts specific to a CCD, like bad rows/columns of pixels.

In other RB models, like in Sánchez et al. (2019), the feature selection is performed purely statistically; features are initially selected based on variance thresholds. In the same work, different techniques are explored to reduce the number of features and thus the complexity of the classification problem. For example, an RF model was trained using all features. Then a feature importance analysis enabled a reduction of the dimensionality of the problem by removing possible redundant or irrelevant features. Examples of models based on features closer to the data include Mong et al. (2020), where the features are simply the flux values of the pixels around the center of the image.

2.3. DNN Approaches

The CNNs have demonstrated enormous potential in image analysis, including object detection, recognition, and classification across domains (Deng et al. 2009). Examples of astrophysics applications of CNNs include Dieleman et al. (2015) for galaxy morphology prediction; Kim & Brunner (2016) for star–galaxy classification; Gabbard et al. (2018) for signal/background separation for gravitational-wave (GW) searches, where the GW time series are purposefully encoded as images to be analyzed by a CNN; and many more.

The CNNs are particularly well suited to learning discriminating features from image input data. They can work on high-dimensional spaces (here the dimensionality of the input is as large as the number of pixels in the image) due to the generalizability of the convolution operation to n dimensions while preserving relative position information. Vectors of raw pixel values can theoretically be used to train traditionally feature-based models, such as RFs, but pixel-to-pixel position data in higher dimensions is unequivocally lost. Previous studies already compared feature-based supervised models, like RF, and supervised CNNs for RB, demonstrating that CNNs generally lead to increased accuracy. In Gieseke et al. (2017), an accuracy of ∼0.984 is achieved with an RF model in the RB task, and it increases to ∼0.990 when applying CNNs to the same data. In Cabrera-Vives et al. (2016), an accuracy of ∼0.9889 is achieved with an RF and is increased to ∼0.9932 with a CNN. In Cabrera-Vives et al. (2017), an RF gives 0.9896 and a CNN 0.9945. In Liu et al. (2019), the accuracy improves from ∼0.9623 with an RF to ∼0.9948 with a CNN.

In Duev et al. (2019), a CNN RB classifier called `braai`, developed for the ZTF (Bellm et al. 2018), achieves an ∼98% accuracy for the training and validation data set (the area under the curve, AUC = 0.99949). The `braai` implements a custom VGG16 (Simonyan & Zisserman 2014) architecture.

[9] DES team (2023, private communication).

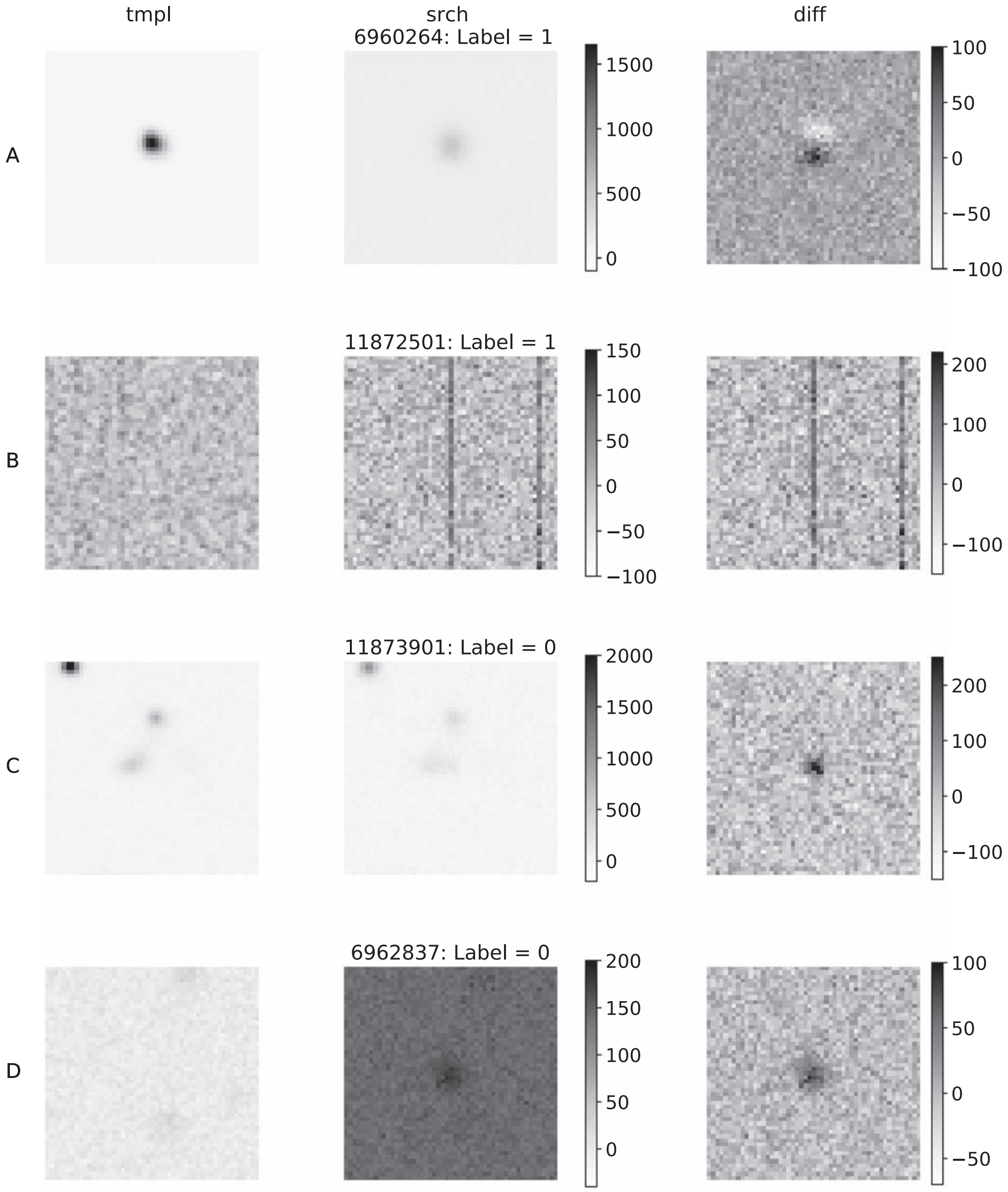

Figure 1. Example of four DIA sets from the first season of the DES. From left to right, the images correspond to the tmpl, srch, and diff image; the diff is generated as the subtraction of tmpl and srch. Each pair of tmpl and srch images is mapped to the same color range. We refer to these three-image sets as "image triplets" or DIA sets. Panels (a) and (b) show artifacts human-labeled as bogus (`label=1`). Panels (c) and (d) show transients labeled as real (`label=0`). Above each triplet is the unique ID of the transient (see Goldstein et al. 2015).

Going beyond RB, a model for image-based transient classification through CNNs has been prototyped by the Automatic Learning for the Rapid Classification of Events team (Carrasco-Davis et al. 2021). The model classifies active galactic nuclei, SNe, variable stars, asteroids, and artifacts in ZTF (Bellm et al. 2018) survey data with a reported accuracy exceeding 95% for all types except SNe (87%). This CNN model was trained using a combination of srch, tmpl, and diff.

Here we explore the potential and intricacies of leveraging AI to bypass the DIA step. The CNN RB models mentioned above differ not only in their architecture (for example, a single or multiple sequences of convolutional, pooling, dropout, and dense layers) but also in the choice of input. For instance, Gieseke et al. (2017) used the tmpl, srch, and diff images in combination for training their CNN; Cabrera-Vives et al. (2016) augmented this image set with an image generated as the diff divided by an estimate of the local noise; and Cabrera-Vives et al. (2017) trained an ensemble of CNNs on different rotations of this fourfold image set. Conversely, Liu et al. (2019) used the diff as the sole input. Although all of these attempts have shown good results with accuracy higher than 90%, these models all rely on DIA to construct the diff. Taking into account that diff is built from tmpl and srch and, in principle, should carry no additional information content than the srch–templ pair alone, a logical step to follow would be to only consider the two latter images.

A first attempt in this direction is presented in Sedaghat & Mahabal (2018), where the authors developed a convolutional autoencoder (encoder–decoder) named `TransiNet`. The model is developed and tested on both real and synthetic data. Synthetic data were created using background images from the Galaxy Zoo data set in Kaggle (Harvey et al. 2013), and then simulated transients were implanted in the srch images. Templ and srch images from the Supernova Hunt project of the

Catalina Real-time Transient Survey (Drake et al. 2009) were also used. Data were fed to the autoencoder to generate a diff image that contains only the transient (the CNNs do not generate background noise). The CNN model was trained and tested only on synthetic data, separately trained on a combination of synthetic and real data, and tested on the real data. The former model achieves scores (precision and recall) of 100%; the latter model has a precision of 93.4% and recall of 75.5% and establishes a precedent for the possibility of avoiding the construction of the DIA diff to reliably detect optical transients.

Another notable work where diff images are not used is Carrasco-Davis et al. (2019). The authors implemented recurrent CNN to train a sequence of images (instead of the classical templ and srch images) to classify seven types of variable objects. The model was trained using synthetic data and tested using data from the High cadence Transient Survey (Förster et al. 2016). The average performance recall of the model is 94%.

Wardega et al. (2021) trained a model that could distinguish between optical transients and artifacts using a srch image from the Dr. Cristina V. Torres Memorial Astronomical Observatory (CTMO; a facility of the University of Texas Rio Grande Valley[10]) and a templ image from the Sloan Digital Sky Survey (SDSS; Gunn et al. 2006). They trained two artificial neural network models, a CNN and a dense layer network, on simulated data and tested the models using data from CTMO and SDSS. The data used for training and testing had specific characteristics; transients were a combination of a source in the CTMO images (or srch image) and background in the SDSS image (or tmpl image), and artifacts were a combination of a source in both the CTMO and SDSS images. Within this data set, both models yield high accuracy (>95%). However, studies based on more diverse and realistic data, e.g., sources near galaxies or embedded in clusters, are needed to demonstrate the feasibility of this approach. The data used in this paper fulfill this condition and are described in Section 3.

3. Data

This study is designed as a detailed comparison RB of CNN-based models with and without diff in input. Our starting point is the well-known `autoscan` RF-based RB (Goldstein et al. 2015; see Section 2.2), which has supported the DES's thousands of discoveries since its first season; we train our model on the data that `autoscan` was trained on and benchmark our results to the performance of `autoscan`. The choice of `autoscan` as our point of reference and benchmark is motivated by its application to the discovery of transients in a state-of-the-art facility, the DES (The Dark Energy Survey Collaboration 2005), which can be considered a precursor of upcoming surveys like the LSST. The latter, expected to start in 2025, will deliver ∼20 Tb of high-resolution sky image data each night, covering a footprint of ∼20,000 deg^2 every ∼3 nights, with an expected millions of transients per night, demanding rapid methodological and technical advances in the accuracy and efficacy of transient detection models (Ivezić et al. 2019). In particular, the properties of the DES images are expected to be similar to those of the Rubin LSST given the similar image resolution ($0\overset{''}{.}26$ and $0\overset{''}{.}2$ pixel^{-1} for DES and LSST, respectively, which results in seeing-limited images taken from nearby sites in Chile with similar sky properties) and imaging technologies (both cameras employ similar chips, wave-front sensing, and adaptive optic systems; Xin et al. 2016), although the field of view of the Rubin LSST is much larger, and the overall image quality is expected to be superior to precursor surveys.

The data used in this work consist of postage stamps of images collected by the DES during its first observational season (Y1), 2013 August through 2014 February[11] (Abbott et al. 2018). The data correspond to 898,963 DIA sets, a tmpl image, a srch image, and their diff. The construction of the templ images for the DES Y1 leveraged the data collected in season two (Y2), as well as the science verification images (observations collected prior to the survey start in order to evaluate the performance of the instrument). More information on the DES DIA pipeline can be found in Kessler et al. (2015). Of these DIA sets, 454,092 contain simulated SNe Ia, which constitute the real astrophysical transient set (`label = 0`), and 444,871 are human-labeled images from DES, i.e., the bogus set (`label = 1`).[12] Each image is 51 × 51 pixels, corresponding to approx 180 arcsec^2 of sky.

Some examples of the data are shown in Figure 1. Each transient is identified by a unique ID. The metadata include the labels associated with each image, as well as the 38 features used for classification in Goldstein et al. (2015). Because we only analyze postage stamps with detections, we still implicitly rely on the DIA to enable the detection step at this stage of our work. The tmpl images in our postage stamps, however, are not PSF-matched.

3.1. Scaling and Normalization

A word about data preparation and normalization is in order, as astrophysical images are inherently very different from the images upon which CNNs have been built. When training CNN models for image analysis, each image is typically simply scaled to a common range (0–1). However, the dynamic range of an astrophysical image is typically large, and the distribution of pixel values is generally very different from Gaussian, with the majority of pixels sitting at low values (the sky) and a few pixels at or near saturation (which in some cases may carry the majority of the information content). Furthermore, in the DIA set case, the pixel value distribution of the diff differs qualitatively from the tmpl and srch ones. While tmpl and srch are typically naturally positive valued with a long tail at the bright end (right-skewed because of the presence of bright astrophysical sources such as galaxies that host transients or stars that vary), the diff image is, in the absence of variable or transient sources, symmetric around zero (see Figure 2).

The diff images were standardized to have a mean $\mu = 0$ and standard deviation $\sigma = 1$. The srch and tmpl images were instead scaled to map the $\mu \pm 3\sigma$ interval of the original image to 0–1. This scheme allows us to retain resolution in the shape of the core of the distribution while also retaining extreme pixel values. Figure 2 shows the distribution of pixel values for four DIA sets for the same data as in Figure 1, which include two

[10] https://www.utrgv.edu/physics/outreach/observatory/index.htm

[11] The data can be found at https://portal.nersc.gov/project/dessn/autoscan/#.

[12] Each image is offered in two formats: ".gif" and ".fits." The former is an 8-bit compressed format (convenient for visual inspection, as it can be opened with commonly available software); the latter, the "flexible image transport system," is a common data format for astronomical data sets that enables high precision with a large dynamic rage. The data in .fits format were used in this work. More information related to how to manipulate this astronomical data format is available at https://docs.astropy.org/en/stable/io/fits/.

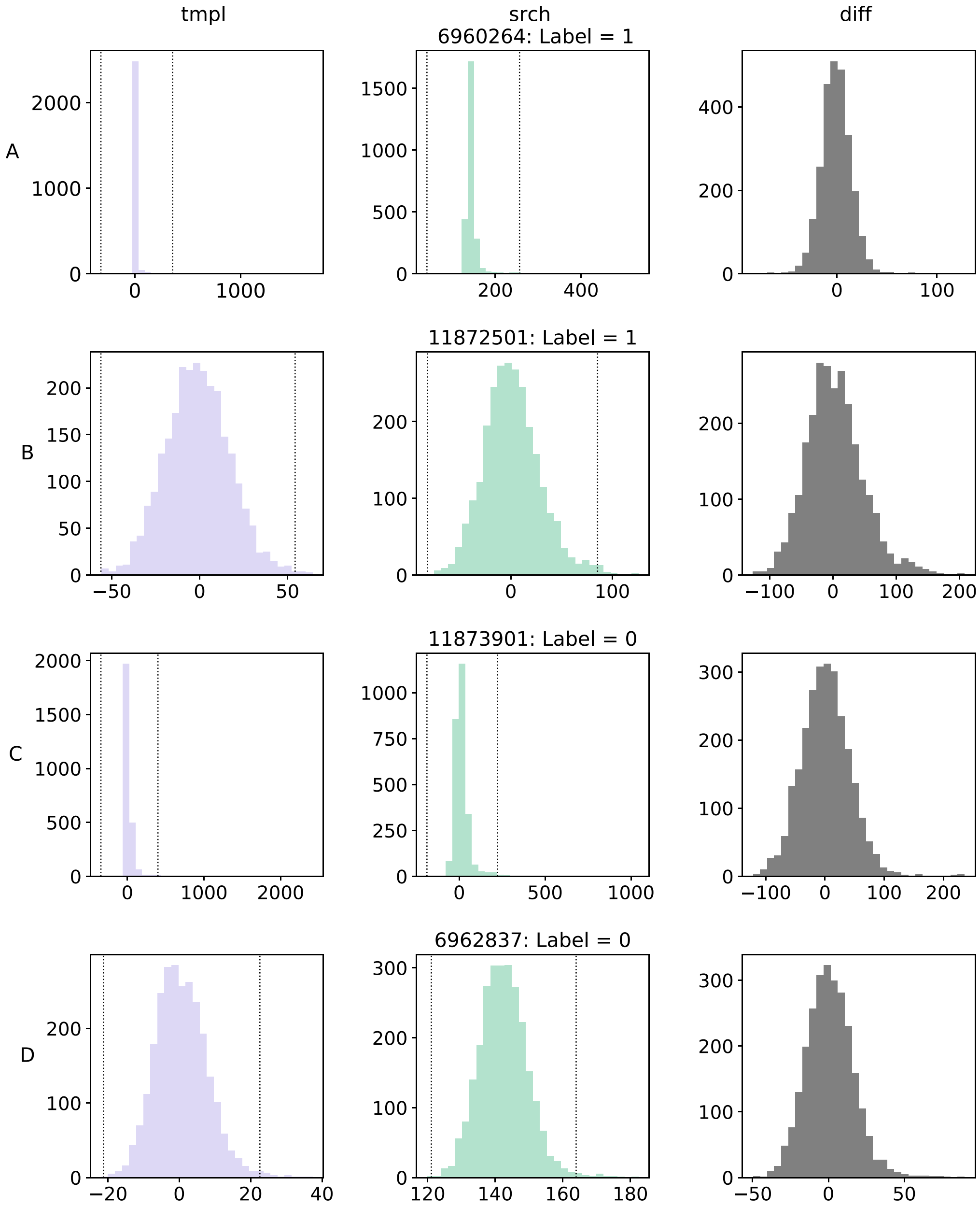

Figure 2. Histogram of pixel values for the data shown in Figure 1 (before scaling and normalization). While all diff images (right) show a bell-shaped distribution, the templ and search images show different behaviors regardless of the real or bogus label. In this image, the x range for each subplot extends to cover the full range of values in the distribution. While this display choice decreases one's ability to discern details in the core of the distribution, it highlights the information on skewness and asymmetry. For example, panels (b) and (d) have similar pixel value distributions; however, panel (b) is bogus, and panel (d) is real. Similarly, panels (a) and (c), a real and a bogus transient, respectively, both show right-skewed distributions for tmpl and srch. The vertical lines show the $\mu \pm 3\sigma$ interval for the srch and tmpl images. The diff images (right column) are standardized individually to a mean of zero and a standard deviation of 1. The srch and tmpl are instead scaled, setting the pixel contained inside the 3σ interval (vertical lines on the histograms) to the range 0–1. This allows one to retain negative values or also values above 1 while keeping the core of the distributions within a homogeneous range. In Appendix A.1, we provide more details about the data preprocessing and include additional plots showing the distribution of data before and after the preprocessing tasks.

real and two bogus labels. The distribution moments used for standardization are shown. Appendix A shows the pixel distributions before and after scaling for the same data in more detail.

One further decision has to be made in combining the three images in the DIA set to feed them to the CNN. While the images would commonly be stacked depthwise, we stacked the scaled diff, srch, and tmpl horizontally. Thus, the size of the data input to our CNN is (N_{tr} × 51 × 153), where N_{tr} is the number of transients to be considered. Four examples of the data "triplets" in input to our *DIA-based* CNN are in the left panels of Figure 3. Following this horizontal structure, we closely mimic the way that humans scanned this type of data for classification, and we will take advantage of this scheme when examining the models' decisions in Section 4.4. We inspected the impact of this choice by comparing the accuracy of a model that was given the images stacked in depth (51 × 51 × 3) with the model with 51 × 153 images in input, with otherwise identical architecture after the first hidden layer, and found that our choice does not affect the overall performance (both models achieved 96% accuracy, as will be discussed in Section 5; see also Appendix A, Figure 14).

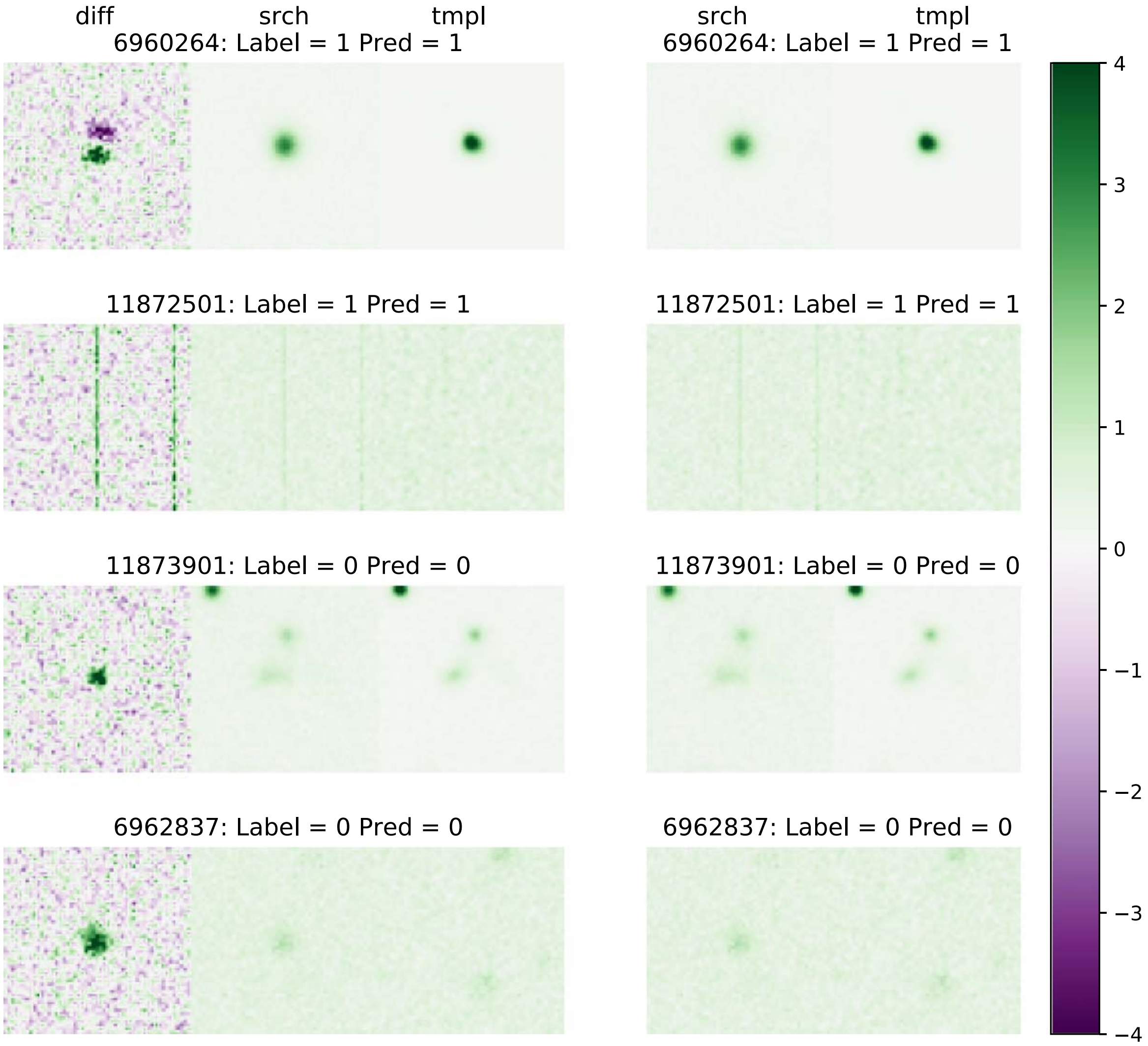
diff
srch
tmpl
6960264: Label = 1 Pred = 1
srch
tmpl
6960264: Label = 1 Pred = 1
11872501: Label = 1 Pred = 1
11872501: Label = 1 Pred = 1
11873901: Label = 0 Pred = 0
11873901: Label = 0 Pred = 0
6962837: Label = 0 Pred = 0
6962837: Label = 0 Pred = 0
4
3
2
1
0
−1
−2
−3
−4

Figure 3. Horizontally stacked data for the examples shown in Figure 1. On the left are the composite images used as input to the *DIA-based* CNN model: the composite follows the order, from left to right, diff, srch, tmpl. On the right are the composite images used for the *noDIA* model, composed of srch and tmpl. Each image element was scaled or normalized following the description given in Section 3 and Figure 2 before combining them into a single image. Above each composite are the unique transient ID, original label, and prediction made by our model. The four transients were classified correctly by both *DIA-based* and *noDIA* models. Purple indicates negative and green indicates positive pixel values.

Since our goal is to measure the impact of reducing the information passed to the model in the input (not using the DIA), for our *noDIA* models, the training and testing data sets were constructed in the same way as the previous triplets, with the tmpl and srch side by side but without the diff image. Some examples are in the right panel of Figure 3.

4. Methodology

4.1. Basics of Neural Networks

Neural networks are models that learn important features and feature associations directly from data. They can be used as supervised-learning models for classification or regression. They consist of a series of layers of linear combinations of the input data, each with real value parameters known as weights and biases, combined with activation functions that enable learning nonlinear, potentially very complex, relationships in the data. The weights tell us the relevance of the input feature with respect to the output, and the biases are the offset values that determine the output. Fitting these quantities to the data minimizes the loss, meaning the prediction is as close as possible to the original target/label (Nielsen 2015). For DNNs, “deep” refers to the fact that there are multiple hidden layers between the input and output layer. With this architecture, the features learned by each layer do not follow a human selection; rather, the features arise in the analysis of the data (LeCun et al. 2015). Among DNNs, CNN models have layers that represent convolutional filters. More information on DNNs and CNNs can be found in Krizhevsky et al. (2012), Dieleman et al. (2015), and Agarap (2018), as well as Gieseke et al. (2017) and many others.

We used the `Keras`[13] implementation of the CNN (Chollet et al. 2015).

4.2. DIA-based *and* noDIA *Model Architecture*

The input of our neural networks is the horizontally stacked images of size 51 × 153 for the *DIA-based* model (diff, srch, tmpl) and 51 × 102 for the *noDIA* model (srch, tmpl). For both the *DIA-based* and the *noDIA*, 100,000 images are used to build the model: 80,000 images for training and 20,000 for

[13] https://keras.io/api/layers/convolution_layers/convolution2d/

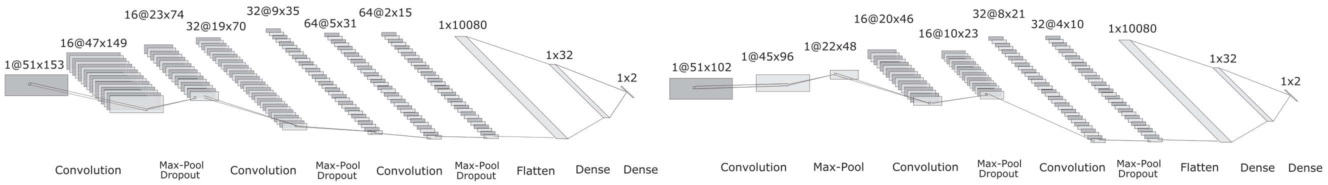

Figure 4. Architecture of the neural networks used in this project to classify real and bogus transients. Left: *DIA-based* model that uses image triplets as input (diff, tmpl, srch). The input layer is 51 × 153 (see left panels of Figure 3); a convolution layer (5 × 5) learns 16 filters; max pooling (2 × 2) and dropout; convolution (5 × 5) learns 32 filters; maximum pooling (2 × 2) and dropout; convolution (5 × 5) learns 64 filters; maximum pooling (2 × 2) and dropout; flatten layer, Dense (32) and the output is a Dense (2)-class layer. Right: *noDIA* model that uses the tmpl and srch images only. The input layer is 51 × 102 (see right panels of Figure 3); a convolution layer (7 × 7) learns one filter; maximum pooling (2 × 2); convolution (3 × 3) learns 16 filters; maximum pooling (2 × 2) and dropout; convolution (3 × 3) learns 32 filters; maximum pooling (2 × 2) and dropout; flatten layer, Dense (32) and the output was a Dense (2)-class layer. The illustrations were made using NN-SVG tool by LeNail (2019).

validation. An additional set of 20,000 images is used for testing; i.e., the predictions on this test are only done after the model hyperparameters are set, and the results reported throughout are based on this set. The images were selected randomly from the 898,963 *DIA* sets, and while training with a larger set can certainly lead to higher accuracy, the number of images was sufficient for this comparison of *DIA-based* and *noDIA* models while being conservative with limited computational resources. The data are composed of 50,183 images labeled as bogus and 49,817 labeled as real.

The network architectures used for this work are shown in the left panel of Figure 4 for the *DIA-based* model and the right panel for the *noDIA* model. More details about the architecture can be found in Appendix B. Both architectures follow a similar structure.

In designing the neural networks, we started with the *DIA-based* model and developed an architecture that would match the performance of Goldstein et al. (2015). While examples exist in the literature of RB models with higher measured performance (see Section 2 and Cabrera-Vives et al. 2016, 2017; Gieseke et al. 2017; Duev et al. 2019; Liu et al. 2019, etc.), we emphasize that each of these models is applied to a different data set, such that their performance cannot be treated as a benchmark. Furthermore, our goal here is not to replace the DES RB model with a higher-performing one but to measure the impact of the loss of information content in the input caused by removing the diff. Matching the performance of the accepted model for RB separation within the DES is a sufficient result for our purposes. In fact, more complex architectures have been implemented on these data without a significant performance improvement (see Appendix B.2). This leads us to believe that at least a fraction of the 3% incorrect predictions are associated with noisy and incorrect labels (see Section 4.3).

With this model in hand, we created a *noDIA* with a similar structure in order to enable direct comparison and measure the effect of the change in the input data.

In the development of our models, we followed two general guidelines.

1. When designing the models, our goal was to push the accuracy of the *DIA-based* CNN model to match the accuracy of `autoscan`. While a more exhaustive architectural exploration or a hyperparameter grid search may well lead to increased efficacy, matching the accuracy of `autoscan` (at ~97%) is sufficient for our demonstration. The final architecture used is one that reached the same accuracy as Goldstein et al. (2015) and the false-positive (FP) and false-negative (FN) rates most similar to the ones obtained in Goldstein et al. (2015; see Figure 5 and Section 5). Once we matched the `autoscan` performance, we focused on the potential for removing the diff image from the input.
2. While the architecture of the *DIA-based* CNN was designed in an attempt to achieve a specific target performance, the architecture of *noDIA* is deliberately kept as close as possible to that of the *DIA-based* model. This enables a direct comparison of the effects of the removal of the diff image. The architecture of *noDIA*, shown on the right in Figure 4, was not explicitly optimized for RB classification; rather, it was inherited from the *DIA-based* model, only modifying the original design to adapt to the different dimensionality of the input data. One further deliberate modification is implemented in the choice of a single-filter first layer for *noDIA*. The diff image is produced by matching the PSF of the science image in the templ (and scaling the brightness with a trivial scaling factor). Everything else needed for the RB classification is contained in the tmpl–srch pair, as it is in the tmpl–srch–diff triplet. Thus, the CNN that is not offered the diff image needs to learn the image PSF, which is constant across the postage stamp–sized image, and it should be possible to model it with a single filter.

4.3. Performance Assessment

Although our goal was to measure the performance impact of the loss of input, as part of our performance assessment tasks, we conducted an extensive hyperparameter search on the existing *noDIA* architecture and tested alternative prepackaged architectures for the *DIA-based* known to perform well on image classification.

We performed grid searches varying the kernel size of the convolutional layers and the batch size (see Appendix B.2, Table 10). The optimal parameters are reported for each model in Appendix B. We retrained two premade deep-learning models, VGG16 (Simonyan & Zisserman 2014) and ResNet 50-V2 (He et al. 2016; see Appendix B.1, Table 5), with the same data used to train the *DIA-based* model presented here. Neither outperformed our *DIA-based* model, while both presented issues with overfitting. Premade architectures force constraints on the size of the images used to train the model; thus, these models were trained with input images stacked in depth (51 × 51 × 3; see Section 3.1). The small size of our postage stamps also limits the available architectures to models with relatively few layers (due to the repeated application of pooling layers). Model VGG16 achieves a performance similar

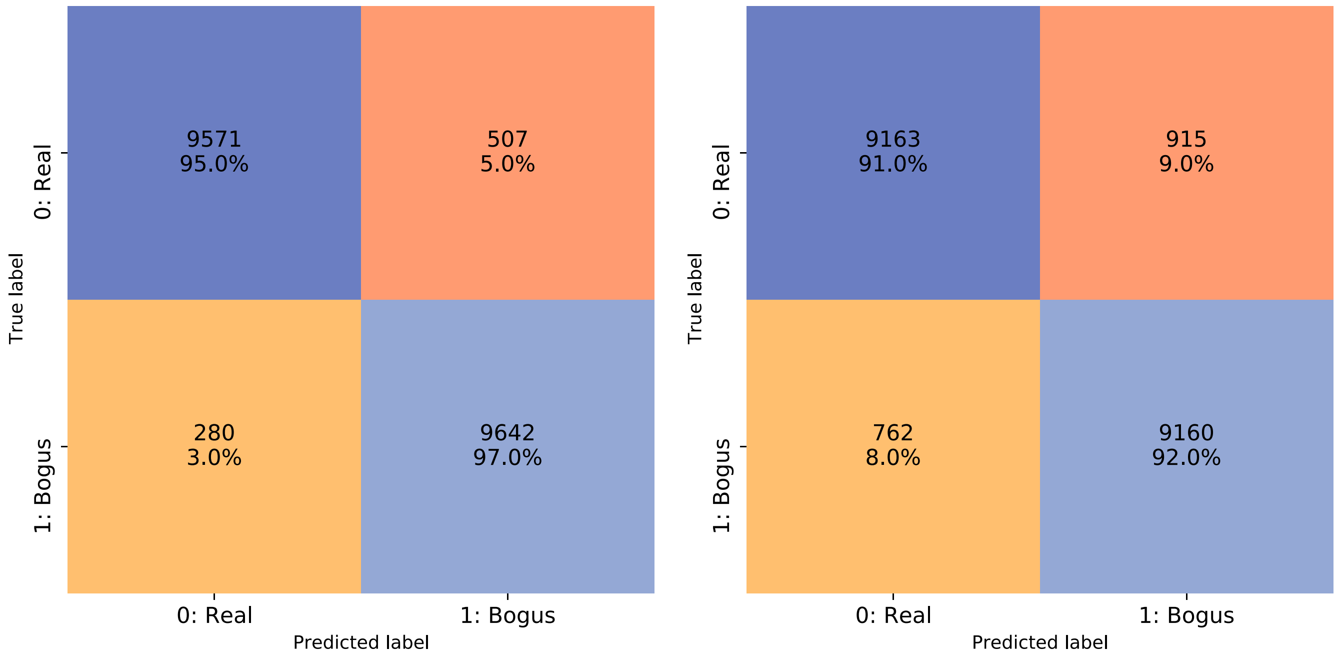

Figure 5. Confusion matrix for our testing data, a set composed of 10,078 objects labeled as real and 9922 as bogus. The squares of the matrix, from the top left in the clockwise direction, indicate TP (`label=0, prediction=0`), FN (`label=0, prediction=1`), TN (`label=1, prediction=1`), and FP (`label=1, prediction=0`). We note that here, somewhat unusually, zero corresponds to real and positive and 1 to bogus and negative, as we chose to remain consistent with the original labeling of the data presented in Goldstein et al. (2015). On the left, the confusion matrix of our *DIA-based* model shows that of the 10,078 transients labeled as real, 9571 (i.e., 95%) were correctly classified, and the rate for the TN objects is even higher at 97%. On the right, the confusion matrix for the *noDIA* model shows that of the 10,078 transients labeled as real, 9163 (i.e., 91%) were predicted correctly, and the TN rate is 92%.

to our model's in terms of testing accuracy with 0.9603, but it trains faster and begins overfitting after ∼20 epochs (where overfitting is visually diagnosed from the loss curves identifying the epoch at which the validation loss starts increasing in spite of continued improvements in the training loss). ResNet 50-V2 shows signs of overfitting throughout the entire training process and only achieves a test accuracy of 0.9416. We conclude that this historical data set contains some level of label inaccuracy such that surpassing the performance of `autoscan` may be effectively impossible.

We remind the reader that in this data set, simulated SNe are by default labeled as real. However, we found that many images labeled as bogus, upon visual inspection, could be reclassified as transients (we estimate between 3% and 10% of the bogus labels could be reclassified; see Appendix D). This data set has now been archived and cannot be further validated by, for example, assessing the recurrence of transients at a sky position to validate the bogus nature of a nonsimulated detection. Thus, while CNN models do exist in the literature with higher accuracy, our model's performance is considered optimal at 0.961. We performed a k-fold cross-validation with $k = 6$ for the *noDIA* model. The average accuracy for the test data set is 0.911 ± 0.005.

4.4. Saliency Maps

Saliency maps quantify the importance of each pixel of an image in input to a CNN in the training process. They provide some level of interpretability through a process akin to feature importance analysis by enabling an assessment of which pixels the model relies on the most for the final classification. If the task were, for example, to identify cats and dogs in images, the expectation will be to find that the most important pixels are located within the dog or cat bodies and not in the surroundings, while if the task were to identify activities performed by cats and dogs, one may find the important pixels both within the dogs and in the surroundings, particularly in objects associated with the performed tasks. Furthermore, some portions of the subject's body may be more distinctive (ears, nose), and we would expect more importance to be given to those pixels. The "importance" of a pixel would be simply measured by the weight the trained neural network assigns to that pixel in the case of a single-layer perceptron, but in the case of DNNs, a highly nonlinear operation is performed on the input data, and the importance, or saliency, of an input feature, or pixel, is harder to assess. Saliency maps have been extracted from CNNs and studied in earlier works, chiefly in Lee et al. (2016). Subsequently, saliency maps have been used as a tool for improving the efficiency of the performance of the CNNs (see, for example, Lee et al. 2021). Within the field of transient detection, Reyes et al. (2018) leveraged saliency maps to identify the most relevant pixels and improve performance on transient classification. We go one step further and use the saliency maps to investigate how the model leverages the DIA image and what information the model uses in its absence.

We follow Simonyan et al.'s (2014) definition of saliency. Denoting the class score with S_c, such that the neural network output on image $\mathcal{I}$ is represented by $S_c(\mathcal{I})$, the saliency map is given by

$$S_c(\mathcal{I}) \approx w^T \mathcal{I} + b, \tag{2}$$

$$w = \left.\frac{\partial S_c}{\partial \mathcal{I}}\right|_{\mathcal{I}_0}. \tag{3}$$

That is, w^T is the weight, b is the bias of the model, w is the derivative of S_c with respect to $\mathcal{I}$ calculated specifically in the local neighborhood of pixel $\mathcal{I}_0$, and the approximation sign indicates that a first-order Taylor expansion has been used to approximate the solution. Each pixel in an image in the training set is associated with the corresponding pixel in the saliency map, and the saliency score (the importance) of that pixel measures the change in model output as a function of changes in the value of that input pixel by back-propagation. The higher the value of a pixel in a saliency map, the more influence that pixel has in the final classification. In this work, we also refer to these maps as maps of pixel importance.

In our case, given the side-by-side organization of the three elements of the input image set, the saliency maps can help us assess how much the *DIA-based* model relied on the diff to enable correct classification and thus provide some intuition about the difficulty of the challenge offered to the *noDIA* model.

For the *DIA-based* model, we have an expectation guided by intuition that a greater concentration of important pixels should be found in the diff image. In Section 5, we will consider the veracity of this hypothesis both qualitatively, by visually inspecting the saliency maps, and by designing a saliency-based metric that enables a quantitative approach. We calculated the normalized sum of the saliency pixel values for each third of the image triplet corresponding to diff, srch, and tmpl. Indicating with I the importance of the segment of an image, ($I_{\rm diff}$, $I_{\rm srch}$, $I_{\rm tmpl}$), where p is a pixel and s_p is its corresponding saliency value, and utilizing the subscripts d, s, and t to refer to pixels in the diff, srch, and tmpl, respectively, we have

$$\begin{aligned} I_{\rm diff} &= \frac{\sum_{p_d} s_{p_d}}{\sum_p s_p}, \\ I_{\rm srch} &= \frac{\sum_{p_s} s_{p_s}}{\sum_p s_p}, \\ I_{\rm tmpl} &= \frac{\sum_{p_t} s_{p_t}}{\sum_p s_p}. \end{aligned} \tag{4}$$

The numerators capture the importance of each third of an image, while the denominator normalizes each metric by the total sum of the saliency pixel values, so that $I_{\rm diff} + I_{\rm srch} + I_{\rm tmpl} = 1$.

This metric allows us to assess the relative importance of the diff (srch or tmpl) component of the image in performing RB classification. Results from these metrics are discussed in detail in Section 5.1.

5. Results

The accuracies of our models and their respective errors, calculated as the standard deviation, are presented in Table 1. The *DIA-based* model, by design, reached the accuracy of our benchmark model `autoscan`: 97% on true-negative (TN) and 95% on true-positive (TP) rates (see Section 4). We remind the reader that we use the `autoscan` convention for the definition of TN and TP; positive is a real transient (label = 0), and negative is bogus (label = 1). There is a drop of ∼4% between the accuracy of the *DIA-based* and *noDIA* models. Along with the accuracy, in Table 1, we present the computational costs of training on 20,000 images measured in CPU node hours[14] and the clock time for the prediction for a single image.

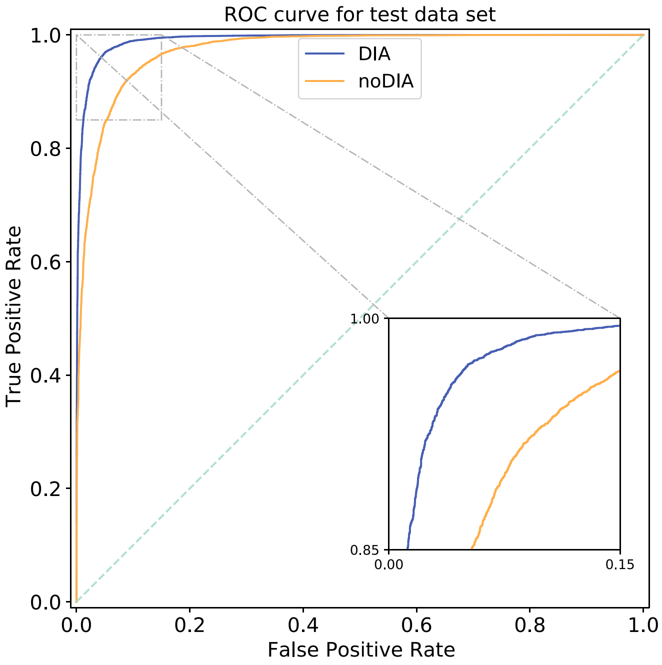

Figure 6. The ROC curve for the 20,000 images used for testing. The blue line is for the *DIA-based* model, and the AUC is 0.992. The orange line is for the *noDIA* model, and the AUC is 0.973. This figure is discussed in Section 5.

Confusion matrices for the testing set are shown for the *DIA-based* model in the left panel of Figure 5 and the *noDIA* model in the right panel. In this figure, as in the confusion matrices and histograms that we will present in Section 5.1, correct predictions are indicated in shades of blue and incorrect in shades of orange, where darker shades are associated with the true labels. The percent accuracy for each class, TPs, TNs, FPs, and FNs, and the number of images in each class are reported within the figure.

The receiver operating characteristic (ROC) curve shows the relation between the TP rate (TP/(TP + FN)), also known as recall, and the FP rate (FP/(FP + TN)) when changing the threshold value (e.g., a threshold of 0.5 will indicate that values greater than 0.5 would be classified as bogus). The ROC for the testing data for the *DIA-based* and *noDIA* models are presented in Figure 6. The AUC, which can be used as a comprehensive metric of the aggregated classification performance of a model (Hanley & McNeil 1983; Hernández-Orallo et al. 2012), is 0.992 and 0.973 for the *DIA-based* and *noDIA* models, respectively. The loss and accuracy curves in the left panel of Figure 7 for the *DIA-based* model show some evidence of overfitting (the validation curve flattens compared to the training curve; starting in the epoch 350) came to an end; meanwhile, for the *noDIA* model in the right panel of Figure 7, after 650 epochs, there was no visual evidence of overfitting, indicating that the model is still learning generalized information from the data, yet the accuracy improvements from epoch 350 to 650 were small.

The nature of the *noDIA* model leads to one hypothesize that because the input data contain less information, this model takes longer to learn features from the data to be able to classify them. The *noDIA* model, in fact, took longer (more epochs) to come to stable accuracy and loss values. The loss and accuracy curves are also noisier for the validation of the *noDIA* model in the right panels of Figure 7 compared to the *DIA-based* model. This also can be explained with the same argument; the *noDIA*

[14] The computational cost was calculated by using the formula given in https://docs.nersc.gov/jobs/policy/ and the allocation information given by the system Iris https://iris.nersc.gov. Training time is reported in CPU hours. Prediction time is reported per "postage stamp" (1 ms), and the classification operation is trivially parallelizable (can be run independently on each postage stamp).

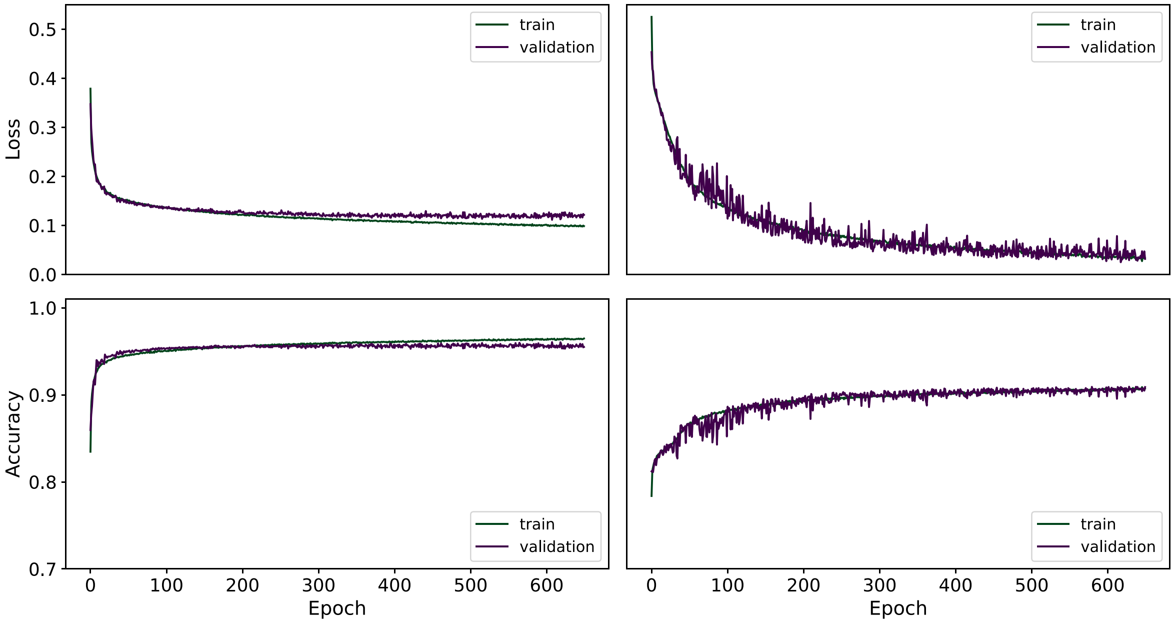

Figure 7. Loss (top) and accuracy (bottom) for both models presented in this work. Details of the loss function, early stop and saving strategies, and optimization method are available in Appendix B. Purple lines correspond to the validation data (20% or 20,000 images) and green to the training data (80% or 80,000 images). Left: results for our *DIA-based* model (Figure 4). Right: results for the *noDIA* model (Figure 4). Both models show convergence at a >90% accuracy with no major overfitting. This figure is discussed in detail in Section 5.

model had a harder problem to solve, and this is reflected in a noisier path to minimization. We conclude that this ~4% loss in accuracy is directly related to the loss of information in the input caused by dropping the diff in input. In addition, we tested if longer training or slightly richer architectures could make up for the loss of diff and found that neither extending the training beyond 650 epochs nor adding convolutional layers improved performance (see Appendix B.1, Tables 8 and 9).

5.1. A Peek into the Model Decisions through Saliency Maps

In Section 3.1, we described how the importance of individual image pixels in the RB prediction performed by our models can be measured and the design of a saliency-based metric to assess which component of the image is most important to perform the RB classification. Here we inspect the saliency maps, both visually and quantitatively, through the measured values of I_{diff}, I_{srch}, and I_{tmpl}.

In Figures 8 and 9, we show the four transients we considered as examples throughout this work, the same images used in Figures 1 and 3, and the corresponding saliency maps for the *DIA-based* and *noDIA* model, respectively. Figures 10 and 11 report the results of Equation (4) for the objects in our training set.

Let us start with some considerations about the saliency maps for the four image examples for the *DIA-based* model (Figure 8), specifically from panels (c) and (d), where the transients were correctly predicted as real. We observe that the greatest concentration of important pixels for both of these images is found in the leftmost third of the image: $I_{\mathrm{diff}} \sim 0.5$ for both. From our experience in labeling real/bogus by visual inspection and consulting with some of the human scanners that labeled the original `autoscan` images, we speculate that this behavior is similar to what a human scanner would do; if there is clearly a real transient in the diff image, the scanner would not need to study the srch and tmpl images in detail.

Figures 8(a) and (b) show correctly classified bogus transients. In Figure 8(a), a bogus likely produced by a moving object displaying a classical dipole, the majority of the important pixels are located in the tmpl ($I_{\mathrm{tmpl}} \sim 0.54$), concentrated around the location of the central source and the location where its "ghost" image is (the coordinates corresponding to the same location of the brightest pixels in the diff but in the tmpl portion of the composite image). In Figure 8(b), there is no central source, and the detection is triggered by an image artifact. The important pixels are found in all three image segments and spread around a large area of each image; the model has inspected the image in its entirety to decide the classification. Following our considerations about the similarity between the CNN and human decision process, having discussed the typical visual inspection process with members of the DES team that human-labeled this data set, we find that here too, the CNN closely mimics what a human scanner would do; because there is no clear central source (a real object) in the diff, the scanner would not simply draw a conclusion based on the diff but instead would analyze the srch and tmpl images to extract more information from the context and enable a robust classification. However, it should be noted that no quantitative studies of the features the human scanners use to classify transients has been done; thus, this remains simply an intriguing suggestion.

For the case of the *noDIA* model, the expectation was less clear; both the srch and the tmpl images are necessary to "reconstruct" the information contained in the diff, and while the pixels overlapping with the central transient are obviously expected to be important, the pixels that surround it are necessary to essentially reproduce the scaling and PSF-matching operations between tmpl and srch that the DIA performs. Accordingly, the saliency maps presented in Figure 9 are more difficult to interpret; in all four cases, important pixels are found all over the composite images.

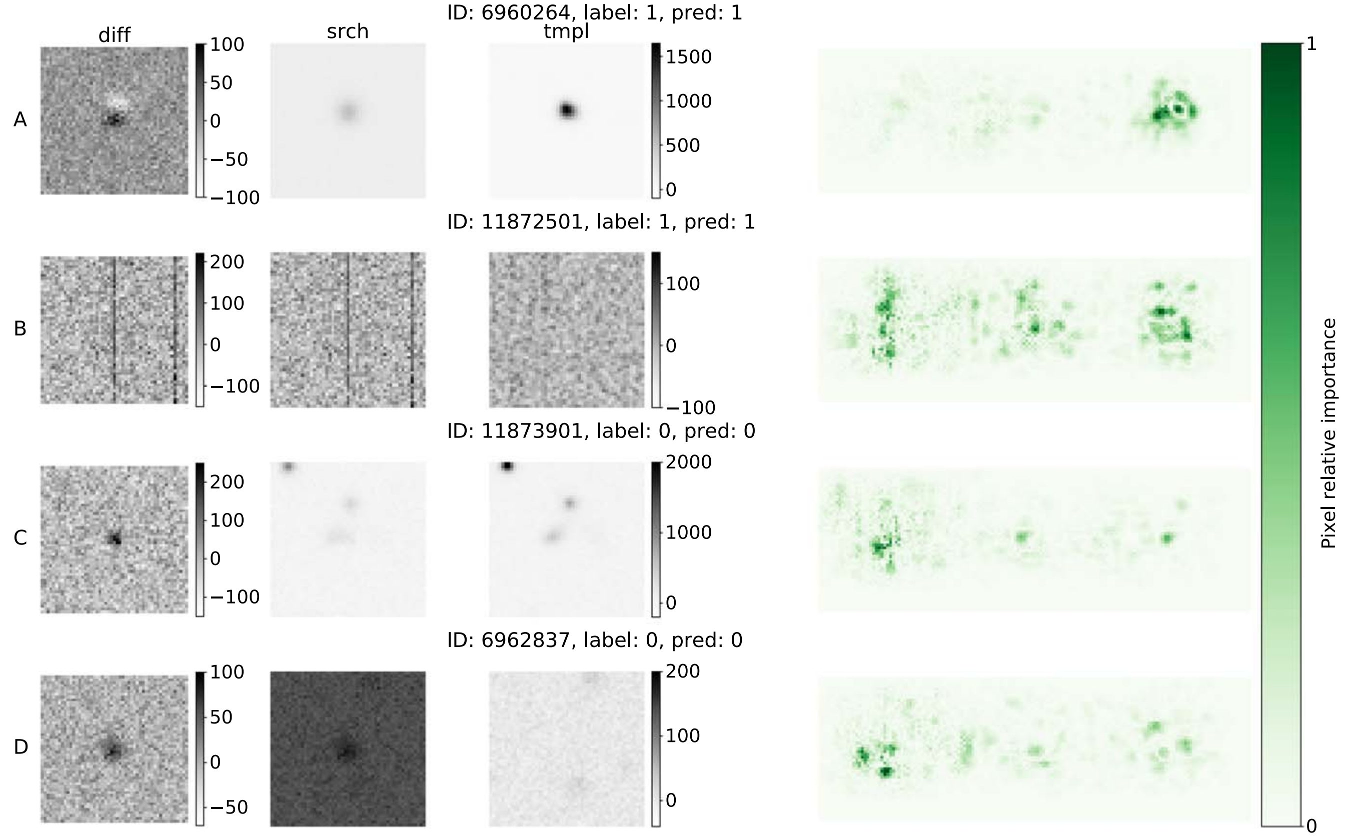
ID: 6960264, label: 1, pred: 1
diff
srch
tmpl
A
100
50
0
−50
−100
1500
1000
500
0
ID: 11872501, label: 1, pred: 1
B
200
100
0
−100
100
0
−100
ID: 11873901, label: 0, pred: 0
C
200
100
0
−100
2000
1000
0
ID: 6962837, label: 0, pred: 0
D
100
50
0
−50
200
100
0
Pixel relative importance
1
0

Figure 8. Saliency map for the four transients in Figure 1 for the *DIA-based* model. On the left, in gray, the original diff, srch, and tmpl images are plotted in their natural flux scale (before normalization). On the right is the saliency map for the combined image. The intensity of a pixel's color on the white-to-green scale indicates the pixel's relative importance; the maps are normalized to 1 individually, such that dark green corresponds to a high saliency score, with 1 corresponding to the most important pixel in the image triplet. With the side-by-side organization of the input data, these maps enable a visual understanding of the importance of each element of the combined image in the RB classification. We note how in some cases (panel (a)), the decision is largely based on the tmpl, rather than the diff image, and in some cases, all three image elements contribute similarly to the decision (panel (b)). This figure is discussed in more detail in Section 4.4.

To explore how the choice of important pixels may depend on the image label and the correct classification, we report the fraction of images for which the diff (srch, tmpl) is the dominant source of important pixels within the confusion matrix in Figure 10. To do this, we use a rough but intuitive cutoff; if the normalized sum of the saliency pixels in a third of the image is larger than $\frac{1}{3}$, then we deduce that the model principally used that component for its decision. For example, where $I_{\rm diff} > 0.33$, we conclude that the model principally relied on diff to make the RB classification. With this cutoff, we can assess if there are differences in the model behavior when classifying objects as a function of their labels or classification. In all four cases (all combinations of real and bogus label and prediction), the concentration of important pixels is largest in the portion of the image corresponding to the diff in the *DIA-based* model. It is, however, interesting to note that in order to correctly classify the bogus, the *DIA-based* model uses the templ and srch images more heavily than in all other cases (diff, srch, tmpl, = 66%, 13%, 21% for TN, while $I_{\rm diff} > 80\%$ for TP, FP, and FN).

The cutoff method described above does not allow us to distinguish between cases where multiple sections of the images were used jointly, perhaps with similar importance, from cases where the model truly relied on only one section of the image. For that, we take a closer look at the distribution of saliency values. In Figure 11, we show the distribution of values of the three metrics defined in Equation (4) for each of the four cases: TP and TN in shades of blue and FP and FN in shades of orange, following the color scheme adopted in Figures 5 and 10. For the *DIA-based* model (the first two columns), for the majority of the 20,000 images in the training set, $I_{\rm diff} > 0.33$, but there is a secondary pick in the $I_{\rm diff}$ distribution near $I_{\rm diff} \sim 0.1$ populated entirely by TN cases, complementary to a long right tail in the $I_{\rm tmpl}$ distribution ($I_{\rm tmpl} > 0.4$). This confirms that the correct classification in the presence of real transients relies on diff, but tmpl and srch become important to correctly classify bogus transients, just like we saw in the exemplary cases in Figure 8. We also note that the general shape of each distribution ($I_{\rm diff}$, $I_{\rm srch}$, $I_{\rm tmpl}$) is similar for the TP and TN cases (blue) and for the FP and FN cases (orange).

For the *noDIA* model, the important pixels are concentrated in the tmpl for most images ($I_{\rm tmpl} > 0.5$) for both correct and incorrect classifications. This is somewhat counterintuitive, since the tmpl does not contain the transient itself. However, one may speculate that this is because the tmpl, a higher-quality image, contains more accurate information about the context in which the transient arises, e.g., whether it is located near a galaxy or not. This information is important to the classification. It is also interesting to note that for the transients predicted as real in both TP and FP, the fraction of images that leveraged primarily the tmpl is approximately two-thirds, and for images predicted as bogus in both TN and FN, it is approximately three-fourths.

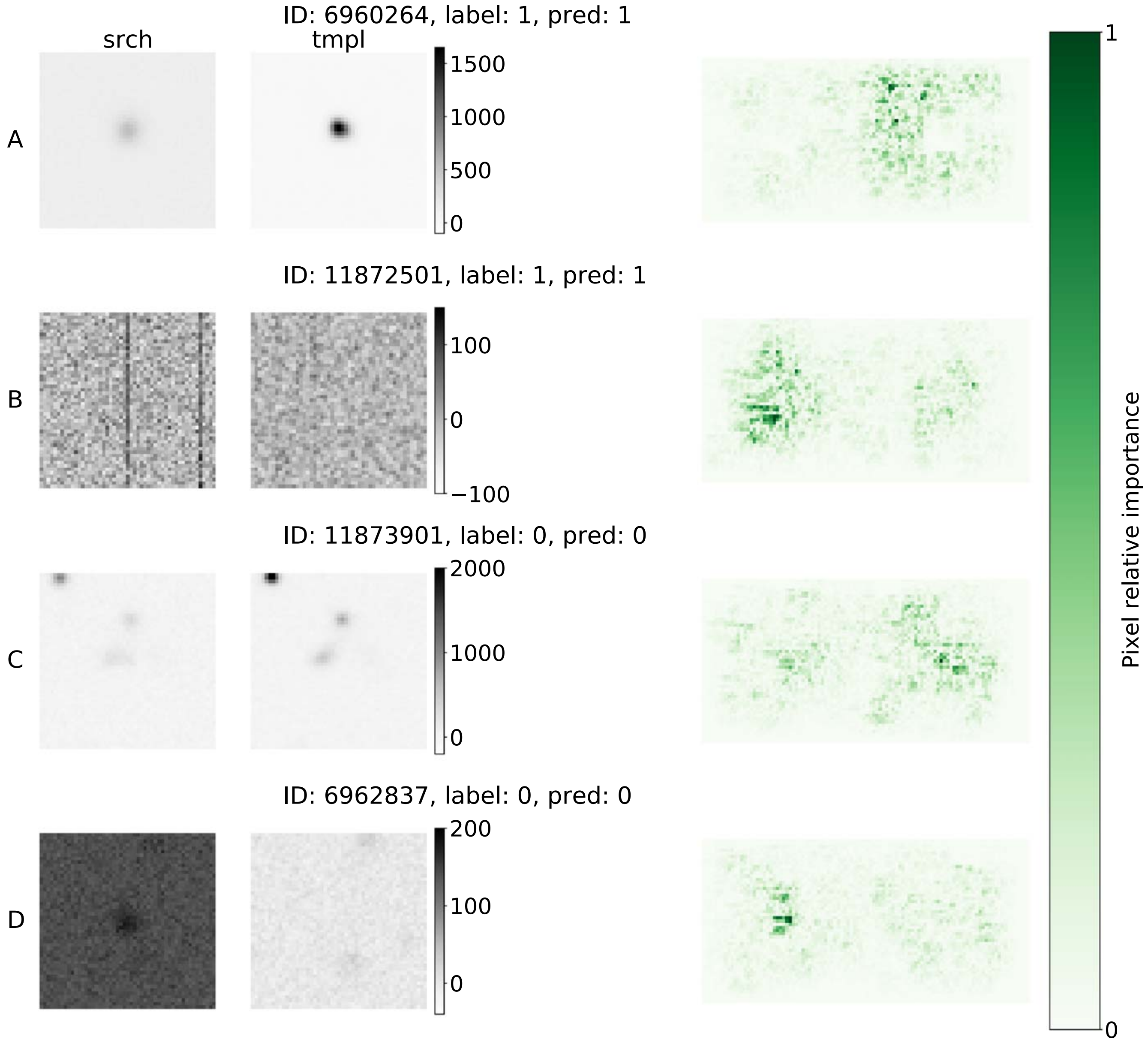

Figure 9. Same as Figure 8 but for the *noDIA* model. This figure is discussed in more detail in Section 4.4.

To help guide the interpretation of the saliency maps, a few more maps are plotted in Appendix D, where we provide six examples per class of the confusion matrix for both the *DIA-based* and *noDIA* models.

5.2. Computational Cost of Our Models

The computational cost of our models, reported in CPU node hours in Table 1, confirms that while training a CNN model for RB can be computationally expensive, and significantly more so if the diff is not used in input (*noDIA*), the model prediction only takes a few seconds, even on large data sets. Using a neural network–based platform, the computational costs are front-loaded. In transient detection, this could mean that the observation-to-transient discovery process can be rapid, while computation time can be spent during off-sky hours (principally to build templ). Furthermore, while the training time is longer for the *noDIA* model than for the model that uses the diff in input (*DIA-based*), the computational cost of the forward pass (prediction) scales superlinearly with the size of the feature set (pixels), so that our *noDIA* model takes less than half the time of the *DIA-based* one to perform the RB task. With a clock time of 0.3 ms per 51 × 51 pixel postage stamp, predicting over the full DES focal plane would take ∼1 minute. However, we note that at this stage of our work, we still rely on the DIA in several ways; while the tmpl and srch in input to *noDIA* are not PSF-matched, this proof of concept is performed on transients that were detected in diff images, and we leverage the alignment of tmpl and srch and centering of the postage stamp that arose from the DIA (see Section 6). Conversely, predictions would only need to be done in correspondence of the sources detected in the tmpl and srch images and not for the entire CCD plane.

6. Future Work and Limitation of This Work

This work is targeted to the investigation of CNN RB model performance with and without diff in input on a single data set,

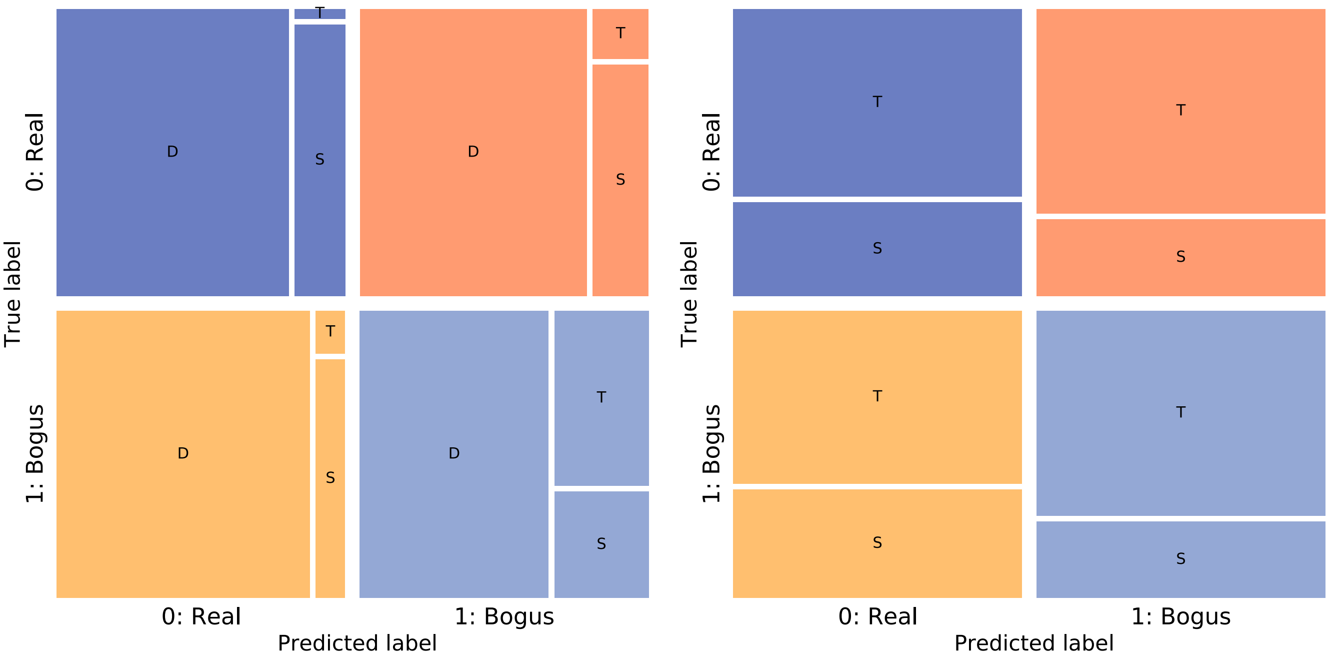

Figure 10. Confusion matrix reporting the proportion of transients for which the highest concentration of important pixels is found in the diff, srch, or tmpl portion of the input image for the *DIA-based* model results (left) and *noDIA* model (right). Left: *DIA-based*. For 80% (7683) of the 9571 transients correctly classified as real, the classification principally relied on the diff image ($I_{\rm diff} > 1/3$); 18% (1758) relied on the srch, and 1% (130) relied on the tmpl. For incorrect real classifications, 88% of the images relied principally on diff, 10% on srch, and 2% on tmpl. For incorrect bogus classifications, 79% of the images relied principally on diff, 17% on srch, and 4% on tmpl. For correct bogus classifications, 66% of the images relied principally on diff, 13% on srch, and 21% on tmpl. Right: *noDIA*. Of the 9163 transients classified correctly as real, for 66% of them, the classification relied principally on the tmpl image. For the incorrect real classifications, 61% of the cases were principally based on tmpl. For the incorrect bogus classifications, 71% of the cases were principally based on tmpl. For correct bogus classifications, 72% of the cases were principally based on tmpl.

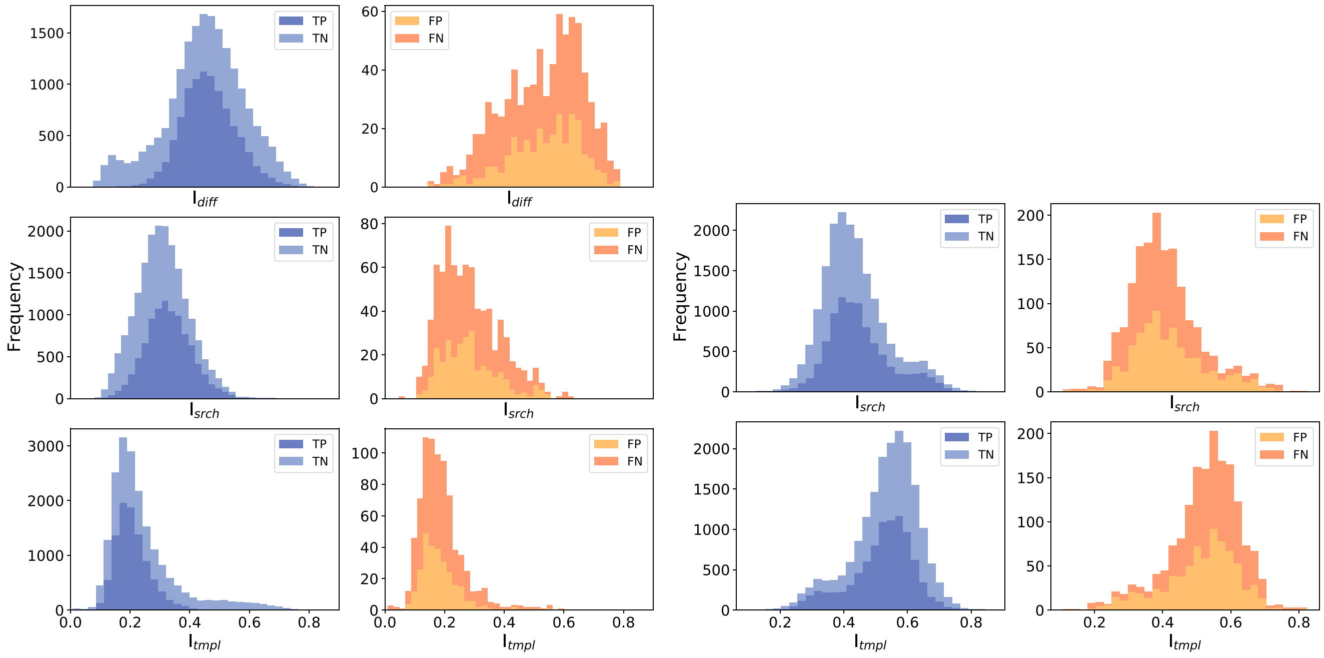

Figure 11. Distribution of $I_{\rm diff}$ (top), $I_{\rm srch}$ (middle), and $I_{\rm tmpl}$ (bottom) value for the 20,000 transients in the test data (see Equation (4)). The colors correspond to the quadrants of the confusion matrix to which the transient belongs according to the model prediction, with *DIA-based* predictions in the first and second columns and *noDIA* in the third and fourth. Blue shades correspond to correct predictions (TP and TN) and orange to FP and FN. Note that the *y*-axis values are different in each plot, and the FP and FN histograms contain far fewer observations. On the $I_{\rm tmpl}$, for the *DIA-based* model (left), the TP classification shows a preference for the diff images, and the distribution peaks at $I_{\rm diff} \sim 0.5$. For TN, however, small $I_{\rm diff}$ values are more common, with an significant fraction of observations in the $0 < I_{\rm diff} < 0.33$ region. This behavior is complemented by a long $I_{\rm tmpl}$ tail in the TN distribution ($0.4 < I_{\rm tmpl} < 0.8$). The FP and FN distributions are qualitatively similar to the TP and TN but noisier, as they contain fewer than 10% of the objects. Right: *noDIA* model. All four classes have qualitatively similar distributions in $I_{\rm tmpl}$ and $I_{\rm srch}$. Classifications rely mostly on the tmpl in all cases ($I_{\rm tmpl} > 0.5$). The mathematical meaning of the plotted quantities is described in detail in Section 4.4. The figure is discussed in Section 5.1.

the same data set upon which the development of the RF-based `autoscan` was based (see Section 3). This approach enables a straightforward comparison, but it comports some limitations.

The labels in our data set come from simulations of SNe (`label=0`) and visual inspection that classifies artifacts and moving objects (`label=1`). We reserve for future work the investigation of the efficacy of the model on transients of different natures, including quasars, strong lensed systems, tidal disruption events, and SNe of different types. These transients may have characteristically different associations with the host galaxy, including preferences for different galaxy types and locations with respect to the galaxy center, compared to the simulated SN events in our training set.

Specifically in terms of Rubin LSST data, an additional source of variability may be introduced by differential chromatic refraction effects (Abbott et al. 2018; Richards et al. 2018) or stars with significant proper motion, which, due to the exquisite image quality of the Rubin images, would be detectable effects.

While we demonstrated the CNN model's potential in the detection of transients without DIA, we did not address the question of completeness as a function of srch or tmpl depth or the potential for performing accurate photometry without DIA.

We trained multiple network architectures to attempt to improve our prediction performance, as discussed in Section 4. None of the models we trained achieved an accuracy that significantly outperformed the RF-based `autoscan`, and none of the *noDIA* versions of our models compensated for the ∼4% loss induced by the removal of the diff in input. Our efforts included training the original *noDIA* for many more epochs, tuning hyperparameters, adding convolutional layers, and adding layers after the last convolutional layer but before the flattened layers to reduce bottlenecks. We intend to continue investigating alternative architectures in future work.

Finally, we note that our models did in fact implicitly leverage some of the information generated by the DIA even when they did not use the diff image itself as input. First, the tmpl and srch images are dewarped. Since the transient alerts are generated from aligned DIA images, the transient source is always located at the image center in our data. We use postage stamps that are, however, not PSF-matched or scaled to match the templ brightness. To move beyond a proof of concept, in future work, we will retrain and apply our models to images whose alignment does not depend on the existence of a transient.

7. Conclusions

In this work, we have measured the accuracy loss associated with the removal of the diff image in input to a CNN trained in classifying true astrophysical transients from artifacts and moving objects (a task generally known as "real–bogus"). We have demonstrated that, while a model with ∼91.1% accuracy can be built without leveraging the results of a diff image analysis (DIA) pipeline that constructs a "template-subtracted" image, there is a loss of performance of a few percentage points.

Starting from the DES data set that supported the creation of the well-known real–bogus `autoscan` model (Goldstein et al. 2015), we first built a CNN-based model, dubbed *DIA-based*, that uses a sky template (tmpl), a nightly image (srch), and a template-subtracted version of the nightly image (diff) that performs real–bogus classification at the level of the `autoscan` model. Our *DIA-based* model reaches 97% accuracy in the bogus classification with an area under the curve of 0.992 and does not require a human decision in the feature engineering or extraction phase.

We then created *noDIA*, a model that uses only the tmpl and srch images and can extract information that enables the identification of bogus transients with 91.1% accuracy. We attribute this performance decrease directly to the loss of information about the PSF of the original image, since, in addition to what is contained in the tmpl and srch pair, the diff contains information about the PSF used to degrade the tmpl to match the quality of the srch image (Section 2). Thus, the convolutional architecture has been unable to recover that information.

We further investigated what information enables the real–bogus classification in both the *DIA-based* and *noDIA* models and demonstrated that a CNN trained with the DIA output primarily uses the information in the diff image to make the final classification and that the model examines a diff–srch–tmpl image set fundamentally differently in the cases where there is a transient than in the cases where there is not one. The *noDIA* model, conversely, takes a more comprehensive look at both tmpl and srch images but relies primarily on tmpl to enable the reconstruction of the information found in the diff.

Implementation of this methodology in future surveys could reduce the time and computational cost required for classifying transients by entirely omitting the construction of the diff images.

Acknowledgments

T.A.C. and F.B.B. acknowledge the support of the LSST Corporation and the Enabling Science Grant program that partially supported T.A.C. through grant No. 2021-040, Universidad Nacional de Colombia and University of Delaware for the Beyond Research Program (2020), and Summer Research Exchange (2021). M.S. and H.Q. were supported by DOE grant DE-FOA-0002424 and NSF grant AST-2108094.

T.A.C. prepared the data, conducted the analysis, created and maintains the machine-learning models, and wrote the manuscript. F.B.B. advised on data selection, curation, and preparation; model selection and design; and revised the manuscript. G.D. advised on data preparation and model selection and design and revised the manuscript. M.S. advised on data selection, shared knowledge about the compilation and curation of the original data set, and revised the manuscript. H. Q. advised on data preparation and model design and revised the manuscript.

This paper has undergone internal review in the LSST Dark Energy Science Collaboration. The internal reviewers were Viviana Acquaviva, Suhail Dhawan, and Michael Wood-Vasey.

The DESC acknowledges ongoing support from the Institut National de Physique Nucléaire et de Physique des Particules in France, the Science & Technology Facilities Council in the United Kingdom, and the Department of Energy, the National Science Foundation, and the LSST Corporation in the United States. The DESC uses resources of the IN2P3 Computing Center (CC-IN2P3–Lyon/Villeurbanne—France) funded by the Centre National de la Recherche Scientifique; the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under contract No. DE-AC02-05CH11231; STFC DiRAC HPC Facilities, funded by UK BIS National E-infrastructure capital grants; and the UK particle physics grid, supported by the GridPP Collaboration. This work was performed in part under DOE contract DE-AC02-76SF00515.

This project used public archival data from the Dark Energy Survey (DES). Funding for the DES Projects has been provided by the U.S. Department of Energy, the U.S. National Science Foundation, the Ministry of Science and Education of Spain, the Science and Technology Facilities Council of the United Kingdom, the Higher Education Funding Council for England, the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, the Kavli Institute of Cosmological Physics at the University of Chicago, the Center for Cosmology and Astro-Particle Physics at the Ohio State University, the Mitchell Institute for Fundamental Physics and Astronomy at Texas A&M University, Financiadora de Estudos e Projetos, Fundação Carlos Chagas Filho de Amparo ã Pesquisa do Estado do Rio de Janeiro, Conselho Nacional de Desenvolvimento Científico e Tecnológico and the Ministério da Ciência, Tecnologia e Inovação, the Deutsche Forschungsgemeinschaft, and the Collaborating Institutions in the Dark Energy Survey. The Collaborating Institutions are Argonne National Laboratory, the University of California at Santa Cruz, the University of Cambridge, Centro de Investigaciones Enérgeticas, Medioambientales y Tecnológicas–Madrid, the University of Chicago, University College London, the DES-Brazil Consortium, the University of Edinburgh, the Eidgenössische Technische Hochschule (ETH) Zürich, Fermi National Accelerator Laboratory, the University of Illinois at Urbana-Champaign, the Institut de Ciéncies de l'Espai (IEEC/CSIC), the Institut de Física d'Altes Energies, Lawrence Berkeley National Laboratory, Ludwig-Maximilians Universität München and the associated Excellence Cluster Universe, the University of Michigan, the National Optical Astronomy Observatory, the University of Nottingham, The Ohio State University, the OzDES Membership Consortium, the University of Pennsylvania, the University of Portsmouth, SLAC National Accelerator Laboratory, Stanford University, the University of Sussex, and Texas A&M University.

Based in part on observations at Cerro Tololo Inter-American Observatory, National Optical Astronomy Observatory, which is operated by the Association of Universities for Research in Astronomy (AURA) under a cooperative agreement with the National Science Foundation.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under contract No. DE-AC02-05CH1123.

This research was supported in part through the use of the DARWIN computing system DARWIN—A Resource for Computational and Data-intensive Research at the University of Delaware and in the Delaware Region, which is supported by the NSF under grant No. 1919839, Rudolf Eigenmann, Benjamin E. Bagozzi, Arthi Jayaraman, William Totten, and Cathy H. Wu, University of Delaware, 2021.

This material is based upon work supported by the University of Delaware Graduate College through the Unidel Distinguished Graduate Scholar Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s).

T.A.C. thanks the LSSTC Data Science Fellowship Program, which is funded by LSSTC, NSF Cybertraining grant No. 1829740, the Brinson Foundation, and the Moore Foundation; her participation in the program has benefited this work.

All of our code is written in `Python` using the following packages:

1. `TensorFlow` (Abadi et al. 2015),
2. `Keras` (Chollet et al. 2015),
3. `numpy` (Harris et al. 2020),
4. `astropy` (Astropy Collaboration et al. 2013, 2018),
5. `pandas` (McKinney 2010; pandas development team 2020),
6. `matplotlib` (Hunter 2007), and
7. `seaborn` (Waskom 2021).

Data Availability

The data underlying this article are available at https://portal.nersc.gov/project/dessn/autoscan/#tdata and explained in more detail in Goldstein et al. (2015).

Appendix A Preprocessing

A.1. Scaling Astrophysical Images for Input to Neural Networks

To visualize and compare the behavior of the flux of the srch and tmpl images, the value distribution for the four transients presented in Figure 1 is shown as a violin plot in Figure 12. The first transient on the left in Figure 12 is labeled as bogus, the srch distribution pixel values are in general greater, positive and nonzero center than the values for tmpl, the same behavior is observed to the last transient on the right, but this one is labeled as real. This same comparison can be applied to the transients plotted in the middle of Figure 12; both show similar distributions, but one is bogus, and the other is real. For the four transients presented, the pixel values have long tails, outside the $\mu \pm 3\sigma$ values. The scaling of the srch and tmpl images for the four transients according to the description given in Section 3.1 is visualized in Figure 13.

A.2. Organization of the Image Components

We have chosen to organize the three elements of the input data, diff, tmpl, and srch, horizontally as a 51×153 input array instead of a more traditional depth stack that makes the data shape $51 \times 51 \times 3$. We made this choice because it allows a more intuitive and clear saliency analysis. We have inspected the impact of this choice and found no performance degradation. Here we present a confusion matrix for a *DIA-based* model with input $51 \times 51 \times 3$ tensors; compared to the performance of the *DIA-based* model presented in Section 5 and Figure 5, there is a small decrease in TN and a corresponding increase in FN rates. Similarly, we trained a *noDIA* version of this model with identical architecture except for the shape of the input layer ($51 \times 51 \times 2$) and found no significant improvement (and slightly more imbalance in the TP and TN classes). The confusion matrix resulting from the $51 \times 51 \times 3$ image arrangement is shown in Figure 14.

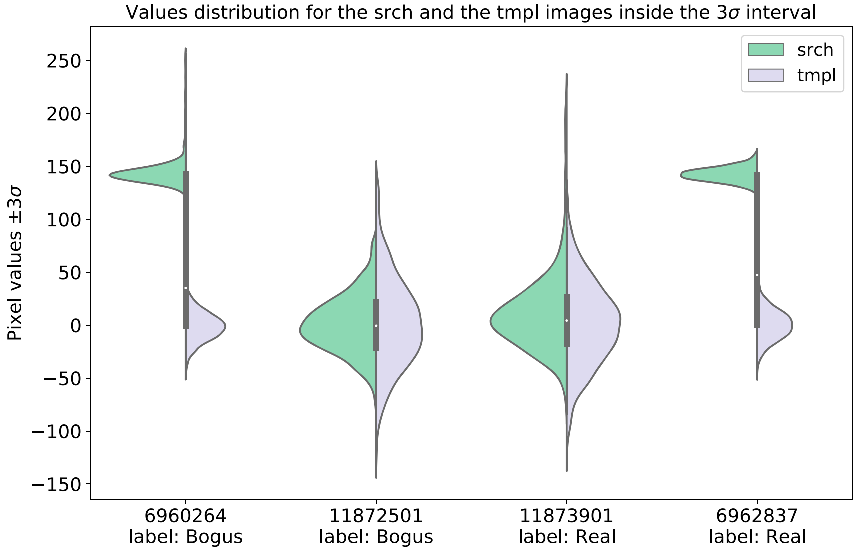

Figure 12. Violin plot of pixel values inside the $\mu \pm 3\sigma$ interval for the srch and temp images of the data shown in Figure 1 (before normalization). These values were scaled between zero and 1. The left and right curves correspond to the srch and temp distribution of values, respectively. The distribution of values corresponds to that shown in Figure 2 for srch and temp, and the green and purple colors correspond to Figure 2.

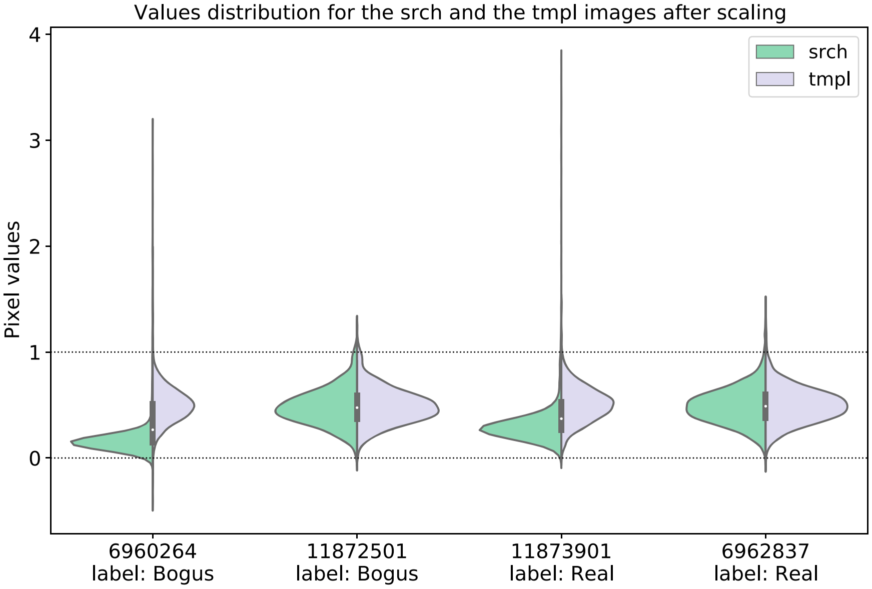

Figure 13. Violin plot of pixel values for the srch and temp images of the data shown in Figure 1 (after normalization). Values between the vertical lines in zero and 1 were mapped to the values in Figure 12. Negative values and above 1 correspond to the scaled values outside the 3σ clip.

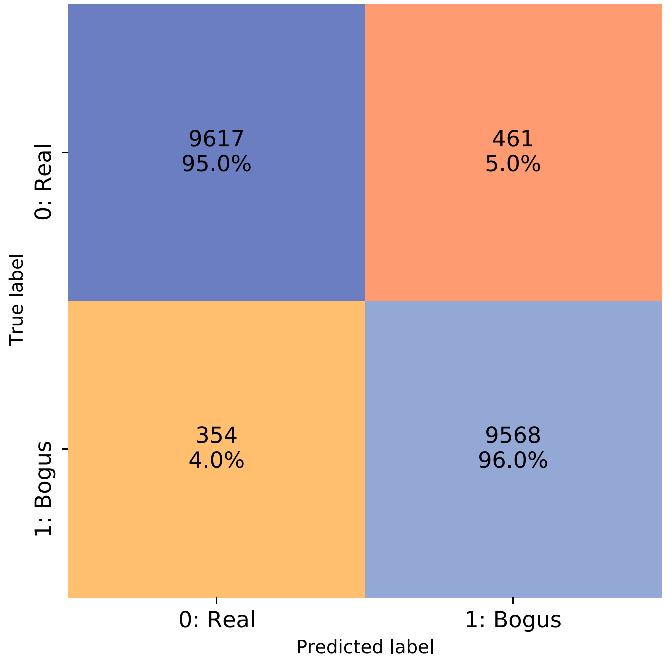

Figure 14. Same as Figure 5 (left) but for a *DIA-based* model that takes input data stacked in a depth axis, i.e., 5 × 151 × 3. Overall, the performance of the model is not significantly affected by the choice of input data, with an ∼1% decrease in the TN rate.

Appendix B
Detailed Architecture of *DIA-based* and *noDIA* Models

We designed a network of 12 layers using `tensorflow.keras` in python for the *DIA-based* case (Table 3) and a network of 11 layers for the *noDIA* case (Table 4). Here we show the details of the architectures of these models. Table 2 shows the compilation hyperparameters, including optimizer, learning rate, loss function, etc., which are shared by all models. In Tables 3 and 4, we show the number of neurons or filters in each layer, the size of the filters and padding choice in convolutional layers, and the activation functions.

B.1. Other Architecture Models

In addition to the CNN *DIA-based* and *noDIA* models presented in this paper, five alternative architectures were tested (see Section 4). The tables in this appendix show the details of the architectures of these models. The structure of the tables is the same as for Tables 3 and 4. Table 5 shows the architecture for ResNet 50-V2 (He et al. 2016) and VGG16 (Simonyan & Zisserman 2014). Only the final nonconvolutional layers are shown, as the convolutional elements are maintained as designed in the respective papers. Tables 6–9 describe the modifications of the final *DIA-based* and *noDIA* architecture described in Tables 3 and 4; they include additional dense layers to assess the possible impact of bottlenecks in information caused by large jumps in the number of neurons between consecutive layers (Tables 6 and 7) and deeper models with additional convolutional and dense layers (Tables 8 and 9), but none of these layers led to improvements in the model performance; thus, the simpler versions were chosen as our final models.

Table 2
Compilation Hyperparameters Shared by All of Our Models

Compilation Configuration	
Optimizer	Stochastic gradient descent
Learning rate	0.01
Batch size	200
Loss function	Sparse categorical cross-entropy
Metric	Accuracy
Save best model	Max. validation accuracy
Stop training	After 100 epochs with no improvements in validation loss

Table 3
DIA-based Model Architecture

Type	Filters	Size	Padding	Activation
Conv2D	16	(5, 5)	Valid	relu
MaxPooling	···	(2, 2)	···	···
Dropout	···	0.4	···	···
Conv2D	32	(5, 5)	Valid	relu
MaxPooling	···	(2, 2)	···	···
Dropout	···	0.4	···	···
Conv2D	64	(5, 5)	Valid	relu
MaxPooling	···	(2, 2)	···	···
Dropout	···	0.4	···	···
Flatten	···	···	···	···
Dense	32	···	···	relu
Dense	2	···	···	softmax

Table 4
noDIA Model Architecture

Type	Filters	Size	Padding	Activation
Conv2D	1	(7, 7)	Valid	relu
MaxPooling	···	(2, 2)	···	···
Conv2D	16	(3, 3)	Valid	relu
MaxPooling	···	(2, 2)	···	···
Dropout	···	0.4	···	···
Conv2D	32	(3, 3)	Valid	relu
MaxPooling	···	(2, 2)	···	···
Dropout	···	0.4	···	···
Flatten	···	···	···	···
Dense	32	···	···	relu
Dense	2	···	···	softmax

Table 5
ResNet 50-V2 (He et al. 2016) and VGG16 (Simonyan & Zisserman 2014) Architectures Tested on the DIA Data

Type	Filters	Size	Padding	Activation
		Specific Convolutional Architecture		
Dropout	···	0.4	···	···
Flatten	···	···	···	···
Dense	32	···	···	relu
Dense	2	···	···	softmax

Note. Only the feed-forward layers are shown. For the specific convolutional architecture details, please refer to the original papers.

B.2. Hyperparameter Grid Search

The combination of hyperparameters tested for the *noDIA* model (see Section 4) and the respective testing accuracy. The

Table 6
noDIA Model with Additional Dense Layers (Size = [24, 16, 8]) after Dense Layer Size 32

Type	Filters	Size	Padding	Activation
Conv2D	1	(7, 7)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Conv2D	16	(3, 3)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Dropout	⋯	0.4	⋯	⋯
Conv2D	32	(3, 3)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Dropout	⋯	0.4	⋯	⋯
Flatten	⋯	⋯	⋯	⋯
Dense	32	⋯	⋯	relu
Dense	24	⋯	⋯	relu
Dense	16	⋯	⋯	relu
Dense	8	⋯	⋯	relu
Dense	2	⋯	⋯	softmax

Table 7
noDIA Model with Additional Dense Layers (Size = [1200, 800, 400, 160, 80]) after Flattening Layer and Additional after Dense Layer size 32

Type	Filters	Size	Padding	Activation
Conv2D	1	(7, 7)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Conv2D	16	(3, 3)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Dropout	⋯	0.4	⋯	⋯
Conv2D	32	(3, 3)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Dropout	⋯	0.4	⋯	⋯
Flatten	⋯	⋯	⋯	⋯
Dense	1200	⋯	⋯	relu
Dense	800	⋯	⋯	relu
Dense	400	⋯	⋯	relu
Dense	160	⋯	⋯	relu
Dense	80	⋯	⋯	relu
Dense	32	⋯	⋯	relu
Dense	24	⋯	⋯	relu
Dense	16	⋯	⋯	relu
Dense	8	⋯	⋯	relu
Dense	2	⋯	⋯	softmax

Table 8
noDIA Model with Additional Convolutional Layers

Type	Filters	Size	Padding	Activation
Conv2D	1	(7, 7)	Valid	relu
Conv2D	8	(5, 5)	Valid	relu
Conv2D	16	(3, 3)	Valid	relu
Conv2D	24	(3, 3)	Valid	relu
Dropout	⋯	0.4	⋯	⋯
Conv2D	32	(3, 3)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Dropout	⋯	0.4	⋯	⋯
Flatten	⋯	⋯	⋯	⋯
Dense	32	⋯	⋯	relu
Dense	2	⋯	⋯	softmax

Table 9
noDIA Model with Additional Convolutional Layers and Dense Layers (Size = [24, 16, 8]) after Dense Layer Size 32

Type	Filters	Size	Padding	Activation
Conv2D	1	(7, 7)	Valid	relu
Conv2D	8	(5, 5)	Valid	relu
Conv2D	16	(3, 3)	Valid	relu
Conv2D	24	(3, 3)	Valid	relu
Dropout	⋯	0.4	⋯	⋯
Conv2D	32	(3, 3)	Valid	relu
MaxPooling	⋯	(2, 2)	⋯	⋯
Dropout	⋯	0.4	⋯	⋯
Flatten	⋯	⋯	⋯	⋯
Dense	32	⋯	⋯	relu
Dense	24	⋯	⋯	relu
Dense	16	⋯	⋯	relu
Dense	8	⋯	⋯	relu
Dense	2	⋯	⋯	softmax

Table 10
Grid Search for *noDIA* Model Varying the Kernel Size of the Convolutional Layers 1, 3, and 5 (see Table 4) and Batch Size

Kernel	Kernel	Kernel		
Layer 1	Layer 3	Layer 6	Batch Size	Test Acc.
5	3	3	100	0.8980
5	3	3	200	0.8892
7	3	3	100	0.8968
7	5	3	100	0.9000
7	5	3	200	0.8958
7	5	5	100	0.9032
7	5	5	200	0.9004
10	3	3	100	0.8960
10	3	3	200	0.8853
10	5	3	100	0.9054
10	5	3	200	0.9008
10	5	5	100	0.9038
10	5	5	200	0.9012
15	3	3	100	0.8953
15	3	3	200	0.8784
15	5	3	100	0.8848
15	5	3	200	0.9023
15	5	5	100	0.8937
15	5	5	200	0.8854

hyperparameter grid search was implemented using `sklearn.model_selection.RandomizedSearchCV`.

Our models are available in a dedicated GitHub repository.[15]

Appendix C
Saliency Maps for Various Transients

We include a series of saliency maps in the following eight figures. Several interesting behavioral patterns can be observed. The figures are organized by model and classification as follows: TN, Figures 15 and 16; TP, Figures 17 and 18; FN, Figures 19 and 20; and FP, Figures 21 and 22.

In Figures 15 and 16, we include the transient's contours overplotted onto the saliency maps for the *DIA-based* and *noDIA* model, respectively, to guide the reader's eye. Notice

[15] https://github.com/taceroc/DIA_noDIA

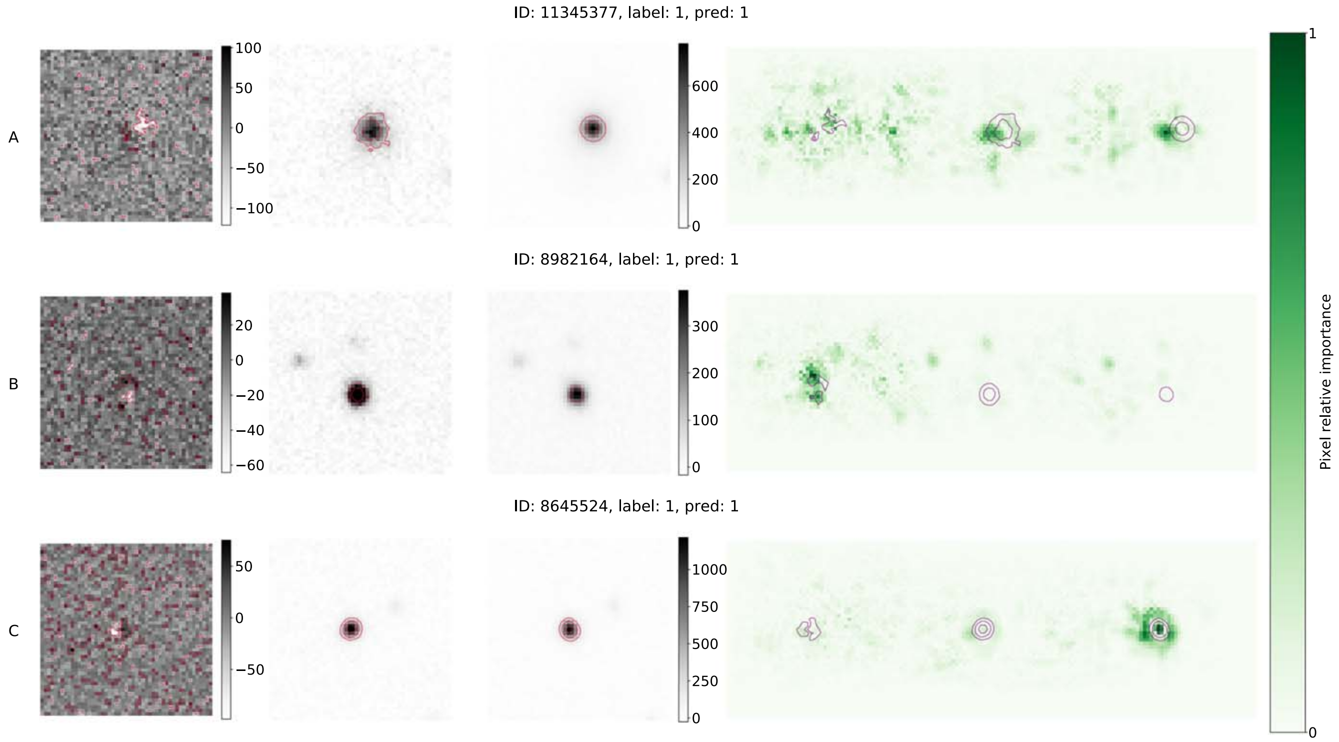

Figure 15. =Transients (diff–srch–tmpl) and their respective saliency maps for *DIA-based* model TN predictions (correctly identified as bogus). A contour plot of light intensity from the original images is overplotted, delineating the bright sources. (a) $I_{\rm diff} \sim 0.4$. (b) $I_{\rm diff} \sim 0.69$. (C) $I_{\rm tmpl} \sim 0.42$. In panel (c), the saliency for the tmpl portion of the image looks strikingly similar to the selection of pixels that is made in aperture photometry, with pixel values considered in the core of the source, ignored in a region immediately around the source, and again considered farther out to calculate the source's background. This figure is further discussed in Appendix C.

the offset of the *DIA-based* model's focus in Figure 15(a) with respect to the transient; the model is principally inspecting the diff and tmpl at the location corresponding to the bright patch in the diff. In Figure 15(c), a correctly predicted bogus, is characterized by dipole probably arising from a poorly centered DIA, the model inspect mainly the tmpl and even seem to reproduce traditional aperture photometry, with pixel values measured at the core of the transient, and in the sky surrounding the transient. The behavior is principally different for the same transients when they are inspected, and also correctly predicted, by the *noDIA* model (Figure 16). The focus of the model in all three cases is away from the transients and shifted to the surroundings; the model is learning the transients' context and extracting information to enable the comparison of tmpl and diff (essentially, to enable the image differencing). This behavior is generally seen throughout all examples in Figures 17–22. In addition, for each figure, we highlight potential reasons for failed predictions and potential inaccuracies in the labeling that may lead to an artificial lowering of our measured accuracy.

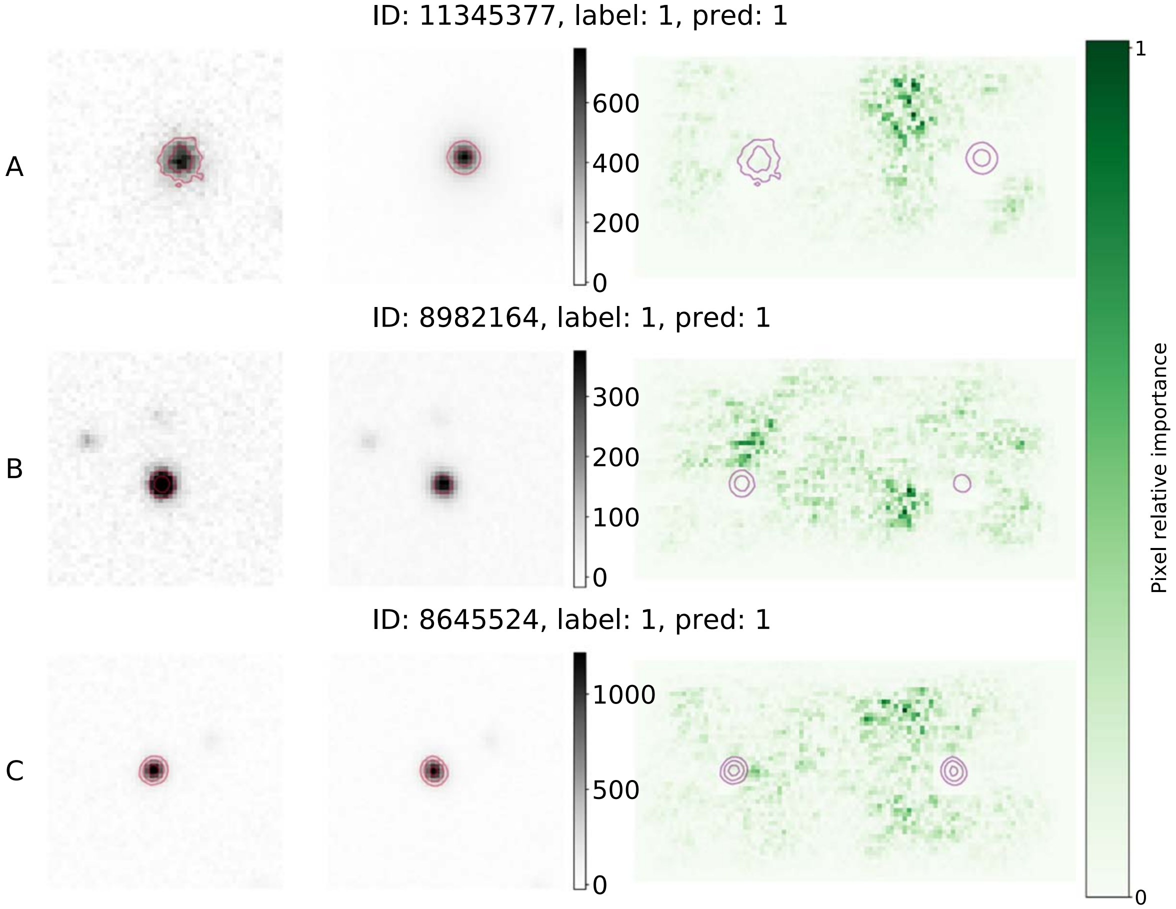

Figure 16. Same transients (srch–tmpl) as in Figure 15 and their respective saliency maps for *noDIA* model TNs (correctly identified as bogus). Important pixels are found at nearly all locations in the image, rather than in a small region around the center. The model needs to learn the properties of the image at large to enable a comparison of the tmpl and diff. This figure is further discussed in Appendix C.

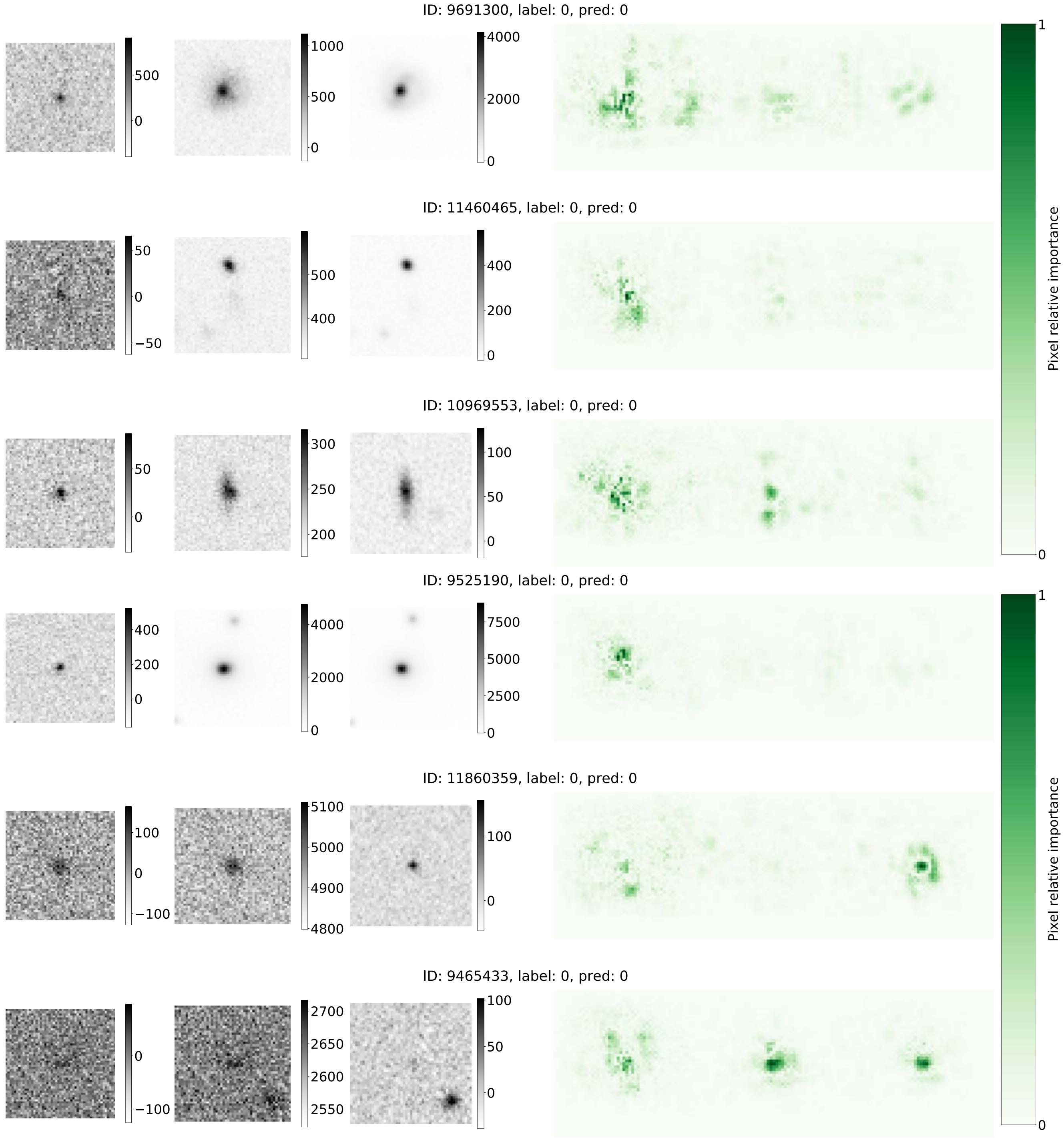

Figure 17. Transients (diff–srch–tmpl) and their respective saliency maps for *DIA-based* model TPs (correctly identified real astrophysical transients). The important pixels are generally found in the diff portion of the image for the *DIA-based* model, as discussed in Section 5, but there are exceptions. Here we show several cases of TP (real transient) classifications where the component of the image that was principally leveraged by the model was the diff (the leftmost third) but two cases where the *noDIA* relied principally on the srch and/or tmpl (bottom two panels).

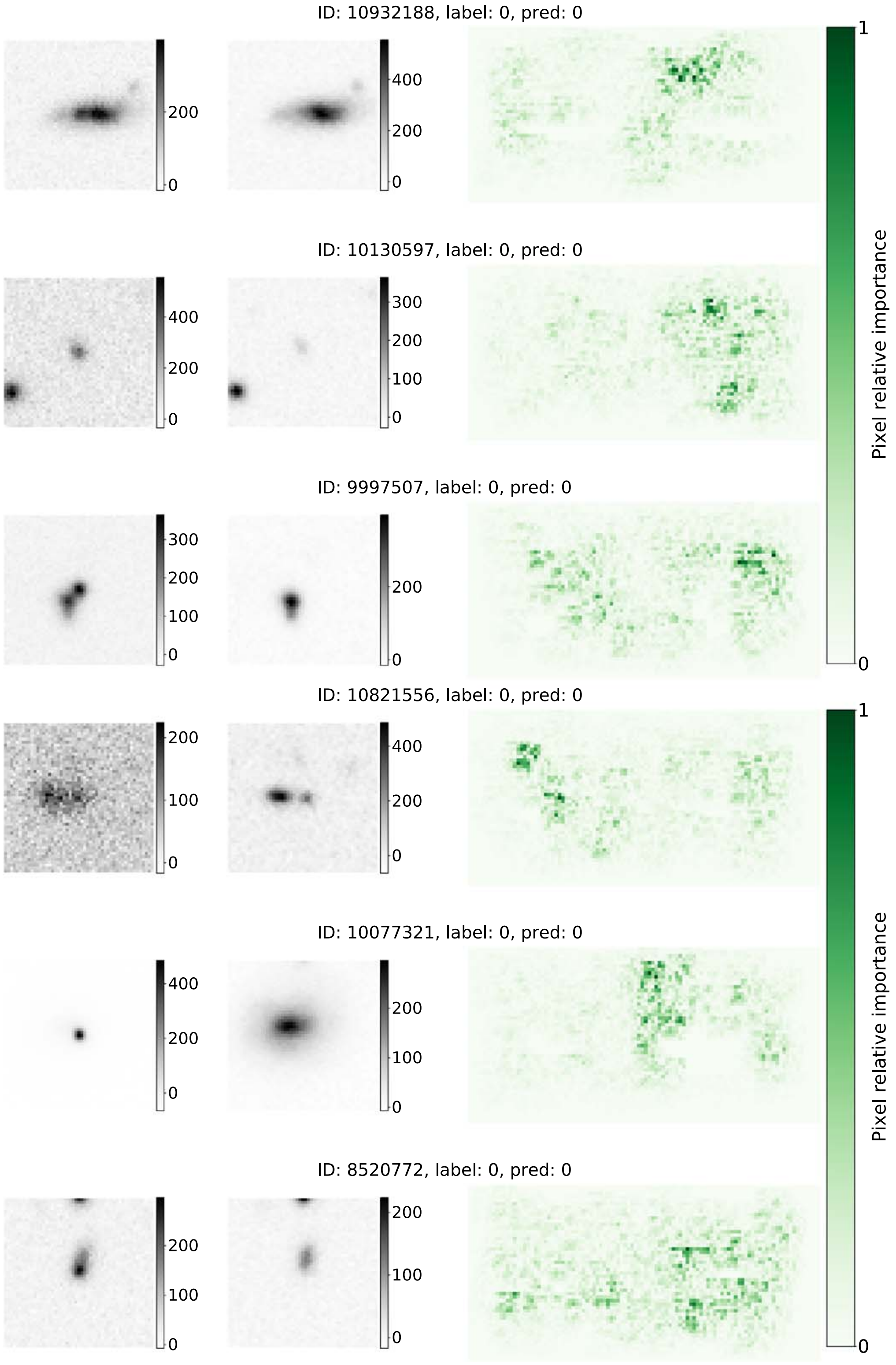

Figure 18. Transients (srch–tmpl) and their respective saliency maps for the *noDIA* model TPs (correctly identified real astrophysical transients). Important pixels are found everywhere in the image, as the CNN learns how to compare the diff and tmpl by taking a synoptic look at the properties of each image component.

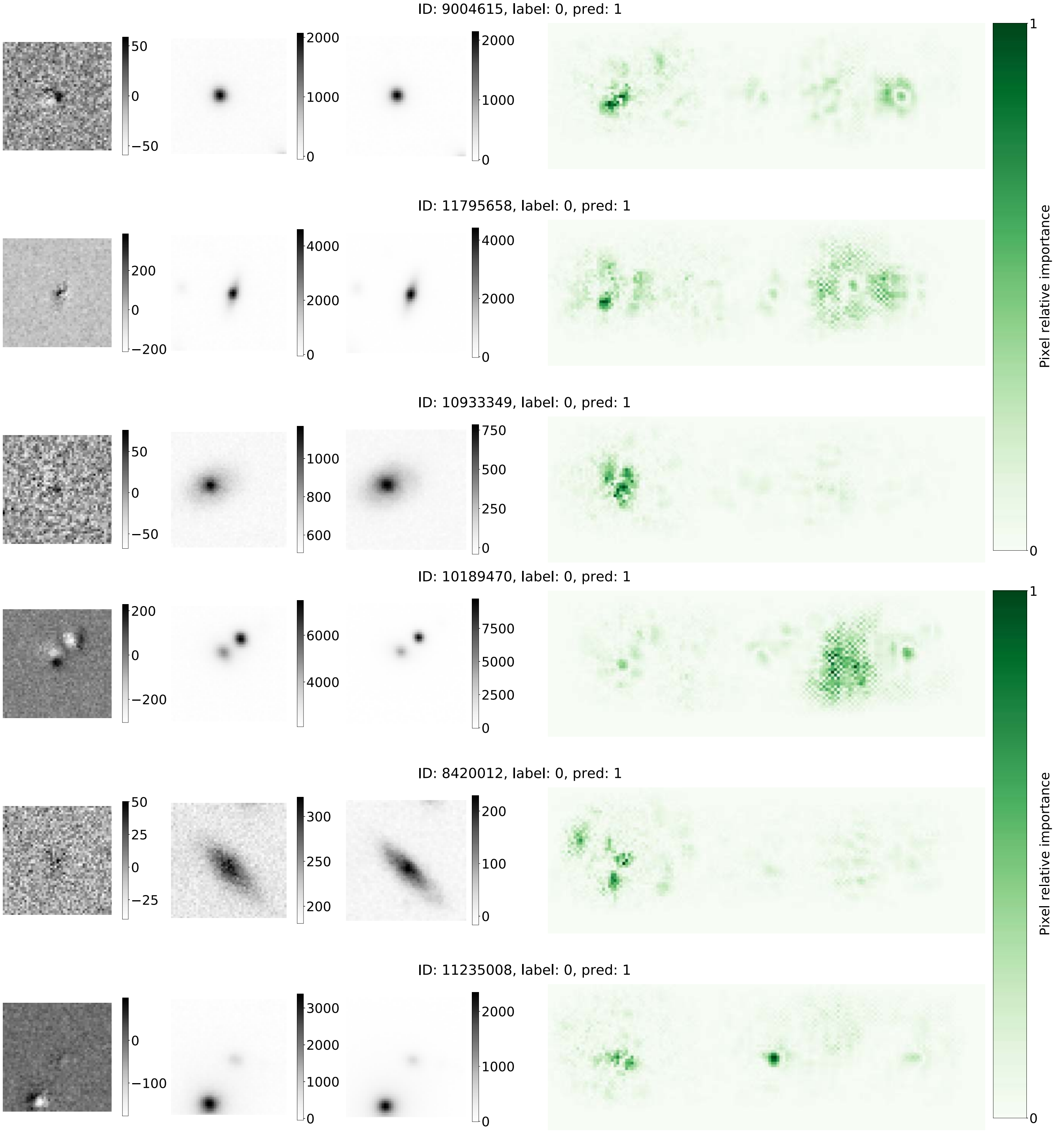
ID: 9004615, label: 0, pred: 1
50
0
−50
2000
1000
0
2000
1000
0
ID: 11795658, label: 0, pred: 1
200
0
−200
4000
2000
0
4000
2000
0
ID: 10933349, label: 0, pred: 1
50
0
−50
1000
800
600
750
500
250
0
ID: 10189470, label: 0, pred: 1
200
0
−200
6000
4000
7500
5000
2500
0
ID: 8420012, label: 0, pred: 1
50
25
0
−25
300
250
200
200
100
0
ID: 11235008, label: 0, pred: 1
0
−100
3000
2000
1000
0
2000
1000
0
1
0
Pixel relative importance
1
0
Pixel relative importance

Figure 19. Transients (diff–srch–tmpl) and their respective saliency maps for *DIA-based* model FNs (real transients identified as bogus). We remind the reader that the labels are inherited from Goldstein et al. (2015) and cannot be verified. Some level of label inaccuracy is expected. The real transients in this data set are implanted SNe onto real DES images. However, in this collection, several transients display DIA inaccuracies (rows 1, 2, 4, and 6 show dipoles; see Section 2.1) that likely lead to the incorrect classification. Two very low signal-to-noise ratio detections are missed (rows 3 and 5) by our model. Important pixels are more commonly found in the diff portion of the image. In the srch saliency maps, we see again that the core of the central source is used in the classification, as well as the pixels that surround the source, but these two sets of important pixels are separated by a gap, again reminiscent of the typical aperture photometry technique (top two panels).

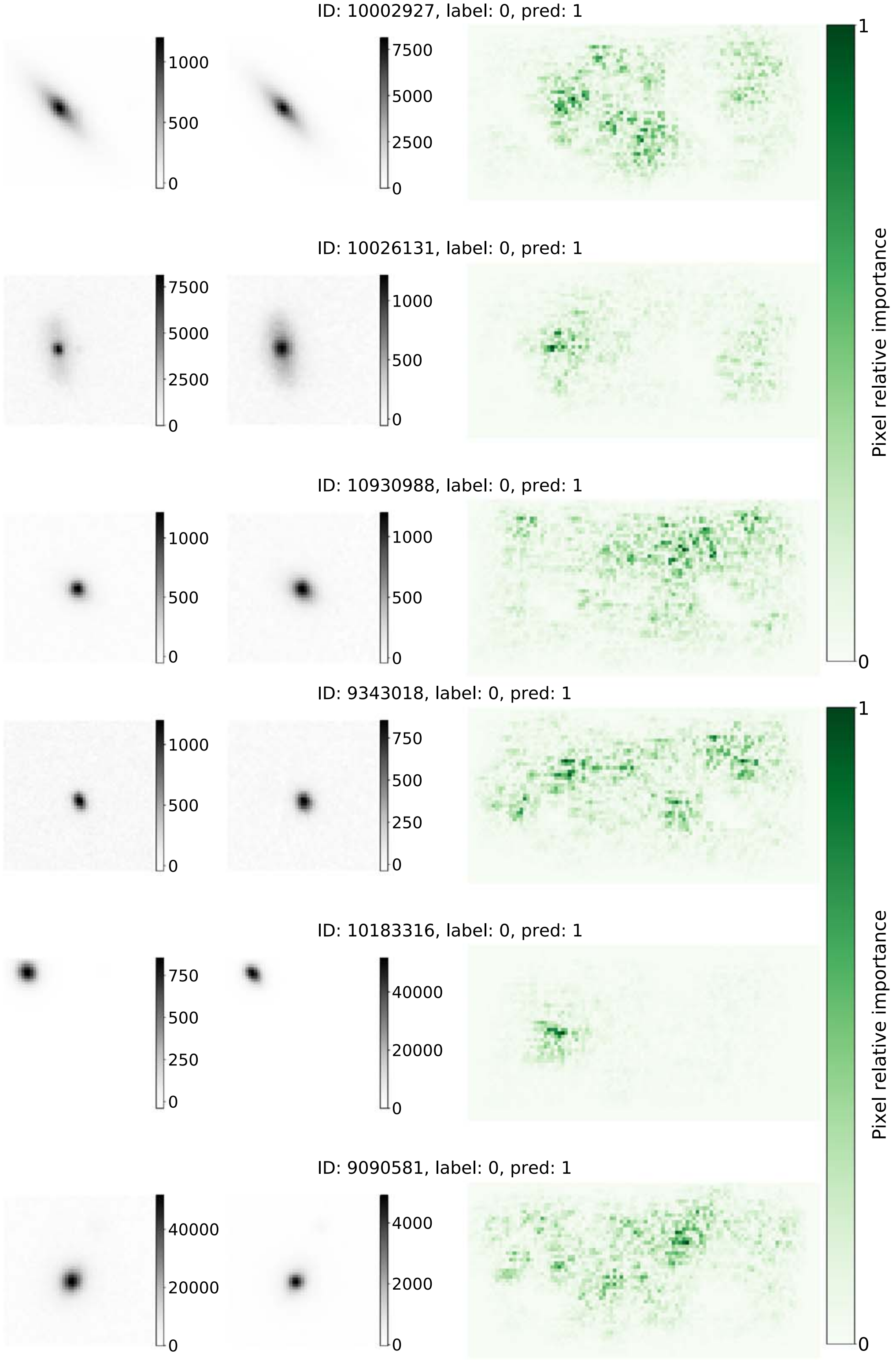

Figure 20. Transients (srch–tmpl) and their respective saliency maps for the *noDIA* model FNs (astrophysical sources classified as bogus). In all cases but row 5, it is not clear why the classification fails. In row 5, another source dominates the image scaling (and preprocessing), reducing the visibility of the transient (that is completely missed by human inspection). We remind the reader that the real transients in this data set are implanted SNe onto real DES images. Important pixels are found everywhere in the image, as the CNN learns how to compare the diff and tmpl by taking a synoptic look at the properties of each image component.

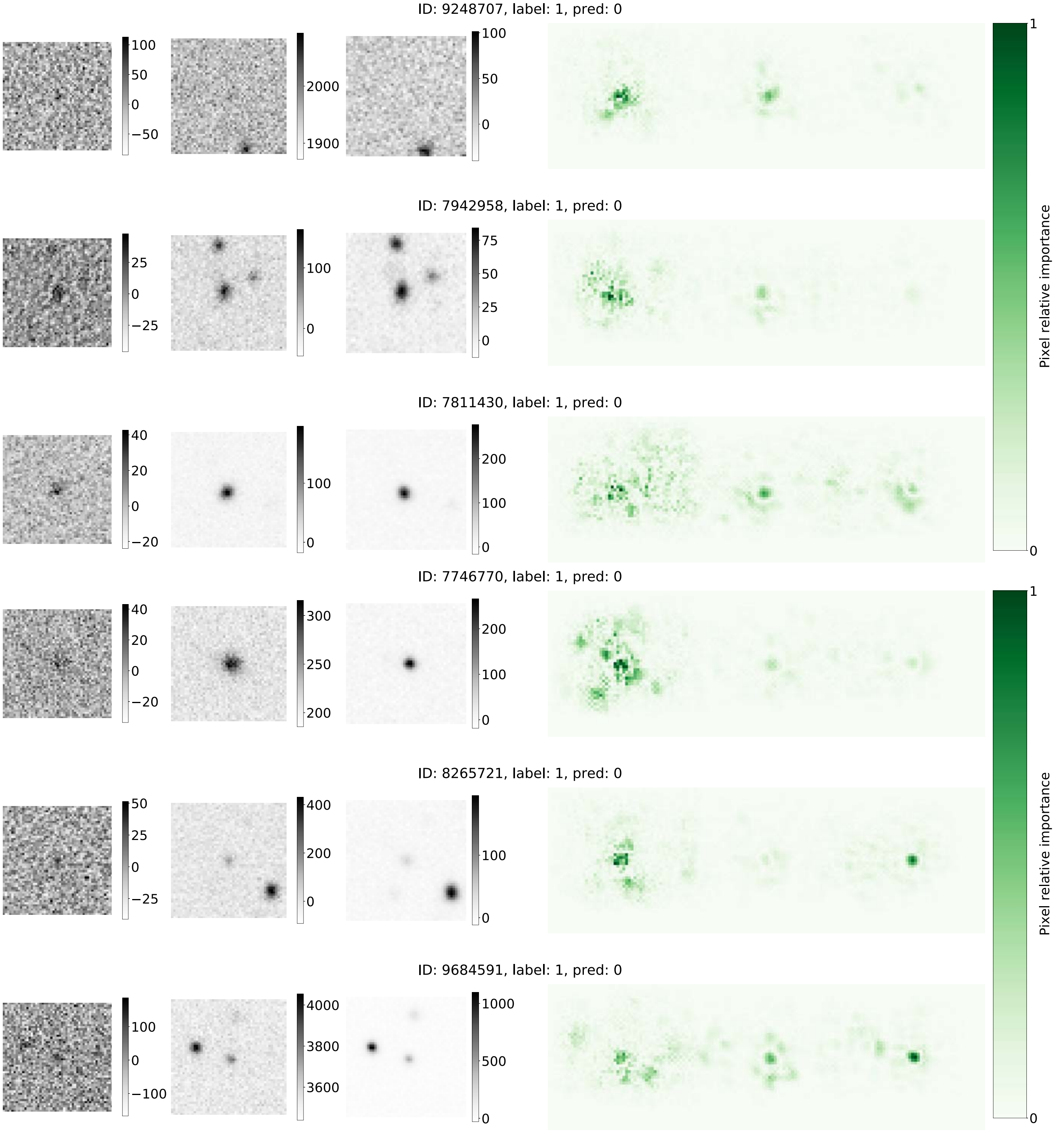

Figure 21. Transients (diff–srch–tmpl) and their respective saliency maps for *DIA-based* model FPs (bogus predicted as real). We remind the reader again that the labels are inherited from Goldstein et al. (2015) and cannot be verified. Some level of label inaccuracy is expected. Bogus transients were labeled by human scanners among astrophysical images with detection. However, in this collection, we cannot verify the nature of the transient, and we argue that in the cases presented here, there is no obvious evidence of its bogus nature. Important pixels are most commonly found in the diff, but in tmpl and srch, we again see that the CNN analyzes the central source and its surroundings but avoids the tail of the central source in a way similar to traditional aperture photometry techniques.

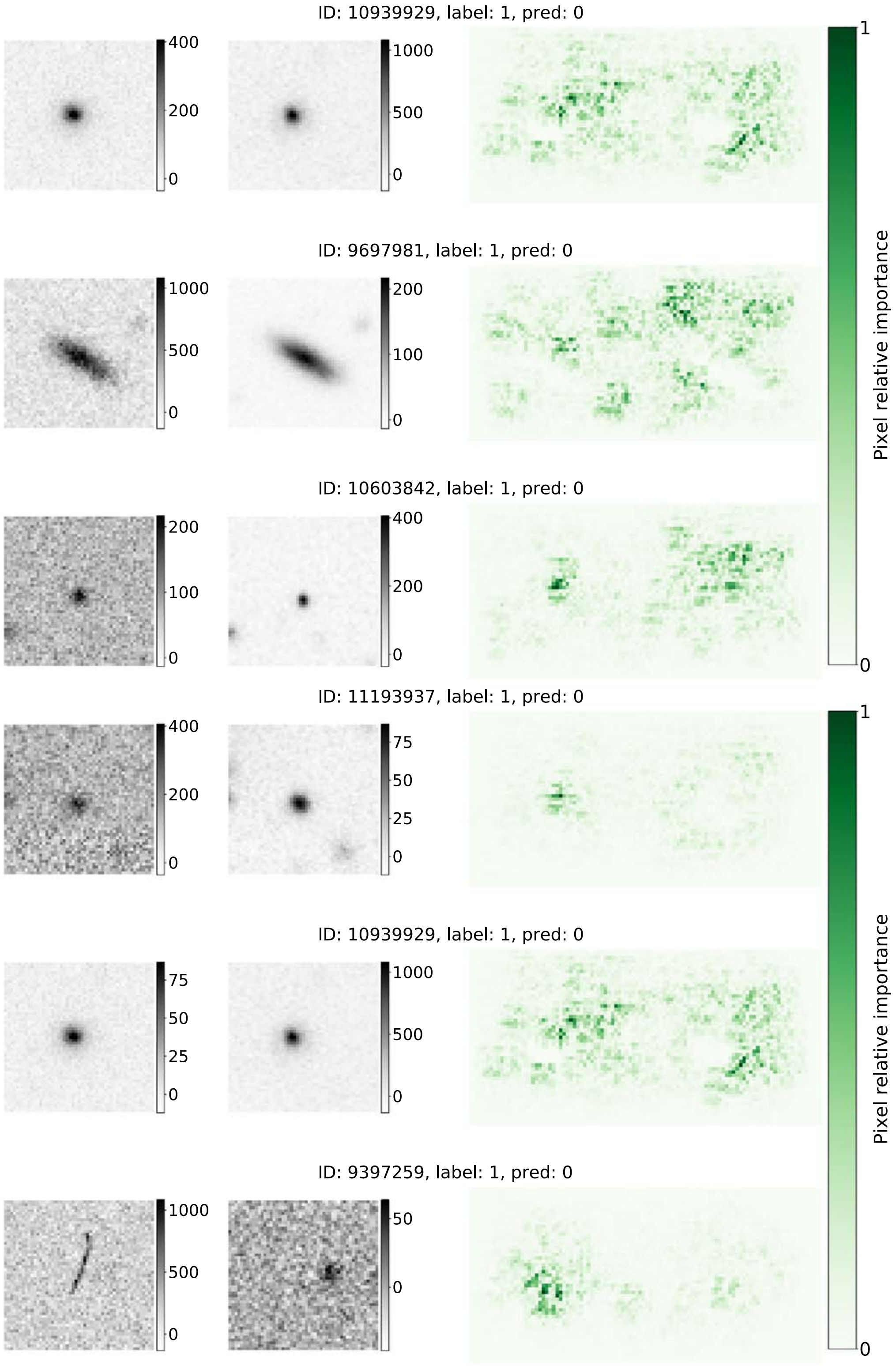

Figure 22. Transients (srch–tmpl) and their respective saliency maps for the *noDIA* model FPs (bogus classified as real). Important pixels are found everywhere in the image as the CNN learns how to compare the diff and tmpl by taking a synoptic look at the properties of each image component.

Appendix D
Visual Inspection of Bogus Images

A total of 300 images originally labeled as bogus were visually inspected by our team; each was inspected by one to five people. It was found that ∼3% of them should be relabeled as real, and a classification disagreement persisted for ∼10% of them.

Some of the models' FPs were also visually inspected. A few examples are shown in Figure 23.

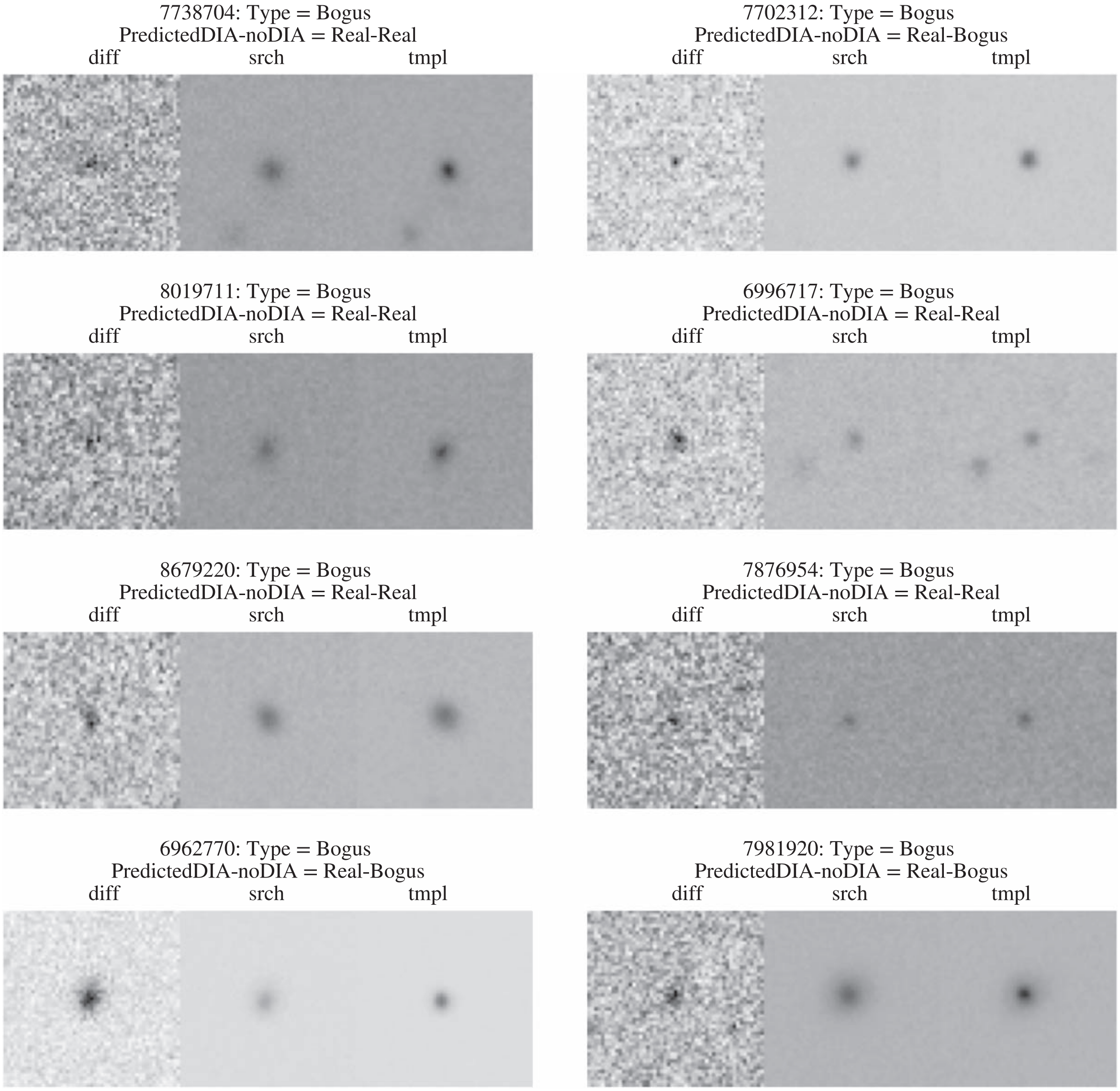

Figure 23. Visual inspection of some objects originally labeled as bogus that were classified as real for *DIA-based* or *noDIA*.

ORCID iDs

Tatiana Acero-Cuellar https://orcid.org/0000-0002-5947-2454
Federica Bianco https://orcid.org/0000-0003-1953-8727
Gregory Dobler https://orcid.org/0000-0002-9276-3261
Masao Sako https://orcid.org/0000-0003-2764-7093
Helen Qu https://orcid.org/0000-0003-1899-9791

References

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

Abbott, T. M. C., Abdalla, F. B., Allam, S., et al. 2018, ApJS, 239, 18

Agarap, A. F. 2018, arXiv:1803.08375

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

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33

Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2018, PASP, 131, 018002

Bloom, J., Richards, J., Nugent, P., et al. 2012, PASP, 124, 1175

Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2016, in 2016 Int. Joint Conf. Neural Networks (IJCNN) (New York: IEEE), 251

Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, ApJ, 836, 97

Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., et al. 2019, PASP, 131, 108006

Carrasco-Davis, R., Reyes, E., Valenzuela, C., et al. 2021, AJ, 162, 231

Chollet, F., et al. 2015, Keras, GitHub, https://github.com/fchollet/keras

Crotts, A. P. S. 1992, ApJL, 399, L43

Deng, J., Dong, W., Socher, R., et al. 2009, in 2009 IEEE Conf. Computer Vision and Pattern Recognition (New York: IEEE), 248

Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441

Drake, A. J., Djorgovski, S. G., Mahabal, A., et al. 2009, ApJ, 696, 870

Duev, D. A., Mahabal, A., Masci, F. J., et al. 2019, MNRAS, 489, 3582

Förster, F., Maureira, J. C., Martín, J. S., et al. 2016, ApJ, 832, 155

Gabbard, H., Williams, M., Hayes, F., & Messenger, C. 2018, PhRvL, 120, 141103

Gieseke, F., Bloemen, S., van den Bogaard, C., et al. 2017, MNRAS, 472, 3101

Goldstein, D. A., D'Andrea, C. B., Fischer, J. A., et al. 2015, AJ, 150, 82

Gunn, J. E., Siegmund, W. A., Mannery, E. J., et al. 2006, AJ, 131, 2332

Hambleton, K., Bianco, F., Clementini, G., et al. 2020, RNAAS, 4, 40

Hanley, J. A., & McNeil, B. J. 1983, Radiol., 148, 839

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Natur, 585, 357

Harvey, D. C., Lintott, T. K., Marshall, P., Willett, K., & Zoo, G. 2013, Galaxy Zoo—The Galaxy Challenge, Kaggle, https://kaggle.com/competitions/galaxy-zoo-the-galaxy-challenge

He, K., Zhang, X., Ren, S., & Sun, J. 2016, arXiv:1603.05027

Hernández-Orallo, J., Flach, P., & Ferri Ramírez, C. 2012, JMLR, 13, 2813, http://jmlr.org/papers/v13/hernandez-orallo12a.html

Ho, T. K. 1995, in Proc. 3rd Int. Conf. Document Analysis and Recognition, 1 (New York: IEEE), 278

Hunter, J. D. 2007, CSE, 9, 90

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

Kessler, R., Marriner, J., Childress, M., et al. 2015, AJ, 150, 172

Kim, E. J., & Brunner, R. J. 2016, MNRAS, 464, 4463

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Proc. 25th Int. Conf. Neural Information Processing Systems—Vol. 1, NIPS'12 (Red Hook, NY: Curran Associates Inc.), 1097

LeCun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436

LeCun, Y. 1989, Connectionism in Perspective (Amsterdam: Elsevier), 143

Lee, G., Tai, Y.-W., & Kim, J. 2016, in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (New York: IEEE), 660

Lee, K. H., Park, C., Oh, J., & Kwak, N. 2021, CoRR, arXiv:2105.00937

LeNail, A. 2019, JOSS, 4, 747

Liu, W., Zhu, M., Dai, C., et al. 2019, RAA, 19, 042

Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, PASP, 131, 038002

McKinney, W. 2010, in Proc. 9th Python in Science Conf., ed. S. van der Walt & J. Millman (Austin, TX: SciPy), 56

Mong, Y.-L., Ackley, K., Galloway, D. K., et al. 2020, MNRAS, 499, 6009

Nielsen, M. A. 2015, Neural Network and Deep Learning, Vol. 25 (San Francisco, CA: Determination Press)

pandas development team 2020, pandas-dev/pandas: Pandas, 1.3.5, Zenodo, doi:10.5281/zenodo.3509134

Reyes, E., Estevez, P. A., Reyes, I., et al. 2018, in 2018 Int. Joint Conf. Neural Networks (IJCNN) (New York: IEEE), 1

Richards, G., Peters, C., Martin, B., & Bauer, F. E. 2018, https://docushare.lsstcorp.org/docushare/dsweb/Get/Document-30573/richard_dcr_wfd.pdf

Sánchez, B., Domínguez, R. M., Lares, M., et al. 2019, A&C, 28, 100284

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

Simonyan, K., Vedaldi, A., & Zisserman, A. 2014, arXiv:1312.6034

Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556

The Dark Energy Survey Collaboration 2005, arXiv:astro-ph/0510346

Tomaney, A. B., & Crotts, A. P. S. 1996, AJ, 112, 2872

Wardega, K., Zadrożny, A., Beroiz, M., Camuccio, R., & Díaz, M. C. 2021, MNRAS, 507, 1836

Waskom, M. L. 2021, JOSS, 6, 3021

Xin, B., Roodman, A., Angeli, G., Claver, C., & Thomas, S. 2016, Proc. SPIE, 9906, 99064J

Zackay, B., & Ofek, E. O. 2017, ApJ, 836, 188