RAS Techniques and Instruments

RASTAI **3,** 385–399 (2024) https://doi.org/10.1093/rasti/rzae027
Advance Access publication 2024 July 12

Training a convolutional neural network for real–bogus classification in the ATLAS survey

J. G. Weston [1]★ K. W. Smith,[1] S. J. Smartt [1,2] J. L. Tonry[3] and H. F. Stevance [2]

[1]*Astrophysics Research Centre, School of Mathematics and Physics, Queen's University, Belfast BT7 1NN, UK*
[2]*Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK*
[3]*Institute for Astronomy, University of Hawai'i, 2680 Woodlawn Drive, Honolulu, HI 96822, USA*

Accepted 2024 July 8. in original form 2024 April 16

ABSTRACT

We present a convolutional neural network (CNN) for use in the real–bogus classification of transient detections made by the Asteroid Terrestrial-impact Last Alert System (ATLAS) and subsequent efforts to improve performance since initial development. In transient detection surveys, the number of alerts made outstrips the capacity for human scanning, necessitating the use of machine learning aids to reduce the number of false positives presented to annotators. We take a sample of recently annotated data from each of the three operating ATLAS telescope with ∼340 000 real (known transients) and ∼1030 000 bogus detections per model. We retrained the CNN architecture with these data specific to each ATLAS unit, achieving a median false positive rate (FPR) of 0.72 per cent for a 1.00 per cent missed detection rate. Further investigations indicate that if we reduce the input image size it results in increased FPR. Finally architecture adjustments and comparisons to contemporary CNNs indicate that our retrained classifier is providing an optimal FPR. We conclude that the periodic retraining and readjustment of classification models on survey data can yield significant improvements as data drift arising from changes in the optical and detector performance can lead to new features in the model and subsequent deteriorations in performance.

Key words: Machine Learning – Data Methods – Algorithms – Convolutional Neural Networks – Classification – Supernovae.

1 INTRODUCTION

Improvements to transient survey methods in the past two decades have led to the emergence of wide-field, all-sky surveys of the local Universe allowing for large regions of the night sky to be studied at short cadences at any given time for supernovae or other transients. The resulting growing and diverse population of discoveries has revealed previously unstudied transient behaviour from unusual supernovae to gravitational wave merger sources (Abbott et al. 2017; Taubenberger 2017; Inserra 2019). As we continue to establish a view of transient behaviour, our focus shifts towards understanding the entire evolution of these objects from the initial outburst to their fading below our survey detection limits. Such a goal requires rapid data processing and analysis to select the best candidates for follow-up observations, for which resources are limited. While techniques have been applied in previous generations of time-domain surveys to success, the recent advancements in machine learning over the past several years have accelerated our ability to sweep large sets of data for relevant information in a short timespan (Bloom et al. 2012; Mahabal et al. 2019; Acero-Cuellar et al. 2023).

The increased depth and scope of wide-field imaging allow surveys to generate tens of thousands of difference images for human scanners to inspect. In difference imaging, a reference image of the sky is matched and scaled to align with the processed sky target image and it is subtracted. In principle, the difference image should be a pixel array representing the combined noise of the two frames and any variable, moving or otherwise astrophysical real transient source. In practice the difference image is dominated by positive and negative artefacts which by far outnumber the real astrophysical sources. This is a common problem in surveys that use difference imaging, and an automated method of removing the majority of false positives (FPs) is required. Following this, human scanners distinguish true 'real' transient detections from the remaining false 'bogus' artefacts (Mahabal et al. 2019; Carrasco-Davis et al. 2021; Duev et al. 1999). Bogus detections make up the majority of the detection population and are produced by multiple factors such as detector defects, saturated sources, and data processing issues (Wright 2015; Gieseke et al. 2017). To reach detection numbers that can be processed by human scanners, we require filtering via machine learning models that are capable of identifying and discarding the majority of bogus detections while maintaining completeness in the real transient population. A machine learning approach, as opposed to one in which feature identification is hardcoded by astronomers, allows for a greater complexity and efficiency of feature analysis. Models must be capable of identifying these transients with early, often incomplete data to ensure that the most interesting candidates can be selected for follow-up observations as soon as possible (Villar et al. 2019; Miranda et al. 2022; Russeil et al. 2024).

Machine learning techniques utilized in transient detection are broadly divided into two categories – supervised (primarily used) and unsupervised (minimally used) methods (Hastie, Friedman & Tisbshirani 2017). In transient astronomy, supervised learning is commonly used for classification and regression, where the model

★ E-mail: jweston04@qub.ac.uk

is required to learn or approximate a function mapping the input to one of multiple classes by looking at a data set consisting of labelled examples. In the context of transients, this can include classifying a given detection as real, classifying the object as one of a list of source types (Leoni et al. 2022; Sheng et al. 2024), or predicting its behaviour over time (Mahabal et al. 2019).

The Asteroid Terrestrial-impact Last Alert System (ATLAS; Tonry et al. 2018b), consists of four telescopes reaching $c \lesssim 19.5$ magnitude (given a 30-s exposure time) in dark skies with the primary aim of detecting potentially hazardous near-Earth objects (Heinze et al. 2021; Reddy et al. 2022). With four operational telescopes, it has the capability of surveying the whole visible sky, multiple times, every 24 h, and hence the data are scientifically useful for a range of variable and transient sources. ATLAS has produced large catalogues of variable stars (Heinze et al. 2018), an all-sky stellar reference catalogue (Tonry et al. 2018a) and tens of thousands of extragalactic transients and supernovae (Smith et al. 2020). Among the highlights are the discovery of luminous fast blue optical transients (the prototype being AT2018cow, Prentice et al. 2018), luminous fast coolers (Nicholl et al. 2023), detections of GRB afterglows (Stalder et al. 2017), and constraints on the counterparts of gravitational wave sources (Smartt et al. 2017, 2023; Ackley et al. 2020). The rapid 1- to 2-d cadence has allowed many precursor emission features of supernovae to be discovered (e.g. Anderson et al. 2018; Srivastav et al. 2023a, b) and early detections of the most luminous (Schulze et al. 2024) and faintest supernovae (Srivastav et al. 2022).

In order to enable these scientific discoveries and light-curve measurements, ATLAS utilizes convolutional neural networks (CNNs) trained on the difference image data through supervised learning to filter out a majority percentage of bogus detections. Initially, there were two ATLAS units operating on Haleakala and Mauna Loa. We trained two bespoke CNN classifiers on data from each telescope, with the Mauna Loa Observatory (MLO) classifier being applied to data from the later operating Sutherland and El Sauce telescopes. Meanwhile many detector and optical refinements were made which affected the delivered point-spread function (PSF). While the telescopes are designed to be identical, such changes to hardware over time have resulted in the image quality (and detector characteristics) to diverge. While we have had good operational performance using a single classifier trained on an all-encompassing but outdated data set, one might expect that CNNs could do better when trained on up-to-date image data from each of the four ATLAS units with their current detector and optical systems.

This paper describes our computational experiments to improve our transient detection methods for the ATLAS survey through improvements to the CNN real–bogus (RB) classifier, particularly paying closer attention to matching the training data to the current telescope performance. Following an introduction to the ATLAS survey and classifier architecture in Sections 2 and 3, we describe how we retrained our classifiers for each individual telescope with recent data in Section 4. Following this, we investigate the effect of varying image size on our performance metrics (Sections 5 and 6). Lastly, we compare the performance of our classifiers with other model architectures (Section 7).

2 ATLAS

2.1 ATLAS and chronology

The ATLAS (Tonry et al. 2018b) was initially developed by the University of Hawai'i and funded by NASA with the intent of detecting hazardous near-Earth objects (e.g. asteroids) which are submitted to the Minor Planet Center (Heinze et al. 2021). This requires multiple telescopes of wide-field design and rapid data processing. As of 2023, the survey consists of four identical 0.5 m telescopes which are a variation of a 'Wright Schmidt' design utilizing a 0.65 m spherical primary mirror and three-lens field corrector. ATLAS initially became operational on the Hawaiian island of Maui with one telescope at the Haleakala Observatory (HKO) in 2015. This was later joined in 2017 by a second telescope on the neighbouring Big Island's MLO, a third in 2021 at the Sutherland Observatory in South Africa (STH), and a fourth in 2022 at the El Sauce Observatory in Chile (CHL). Each telescope uses two filters in normal survey operations (Tonry et al. 2018b). The first is a cyan filter (c) used during dark time which is roughly analogous to a composite PS1 $g + r$ (420–650 nm). In bright time, an orange filter (o) roughly equivalent to $r + i$ (560–820 nm) is used. The survey is capable of surveying the entire sky between a declination of $\pm 50^\circ$ with a 1-d cadence and the polar regions with a 2-d cadence (Tonry 2010; Tonry et al. 2018b) with a repeat coverage of 4 times each night, with 30-s exposures.

2.2 Difference imaging and transient detection

Each telescope camera contains an STA $10\,560^2$ pixel single CCD detector with a 30 pixel border and scale of 1.86 arcsec per pixel. The detector and telescope optics deliver a $5.4^\circ \times 5.4^\circ$ field of view. The current survey strategy employs 30-s exposures on all units (Tonry et al. 2018b). Transients are detected via a process known as difference imaging (Tomaney & Crotts 1996; Alard & Lupton 1998). As outlined in Tonry et al., a matching reference image is constructed from the ATLAS wallpaper, which we rebuild periodically. We use a modified version of the HOTPANTS image subtraction algorithm (Becker 2015), itself an implementation of the Alard & Lupton algorithm for difference imaging, which photometrically aligns the input image with a reference following astrometric alignment (Alard & Lupton 1998; Alard 2000).

The complete ATLAS reduction pipeline can be found outlined in Tonry et al. (2018b). The pipeline has a custom PSF fitting routine called TPHOT that produces flux measurements of all difference image sources that are detected at $> 5\sigma$ above the background noise, which are outputted in a text file marked with the .ddc file. The .ddc files are transferred to a machine at Queen's University Belfast and ingested into a transient object data base (Smith et al. 2020) alongside the reduced and calibrated images with their associated subtracted frames. Detections are first cone searched against previously ingested objects, with those within 3.6 arcsec of an existing object being ingested and associated with that object to contribute to its light curve. New objects are created from detections that have no pre-existing objects within the given proximity. An object is defined from a set of three or more good quality, co-spatial detections that all occur on the same night more than 100 pixels away from the detector edge. The detection closest to the mean coordinates of the object is identified as the representative detection for the object.

Following the extraction of detections in Hawai'i, basic quality control checks are carried out as outlined in Tonry et al. (2018b). For detections that pass this process 'stamps' are created of the target, reference, and difference images measuring 6.2 arcmin (200×200 pixels). The 20×20 grey-scale pixel cores of these difference image stamps are passed to the machine learning model for classification.

Figs 1 and 2 display examples of real and bogus transient detections. The left frame is the target image of the detection, while the middle is the relevant space cut from the ATLAS reference wallpaper. The right frame is the resulting subtraction which should only contain the net flux of the detected object. This is not the case for the bogus detections contained in Fig. 2.

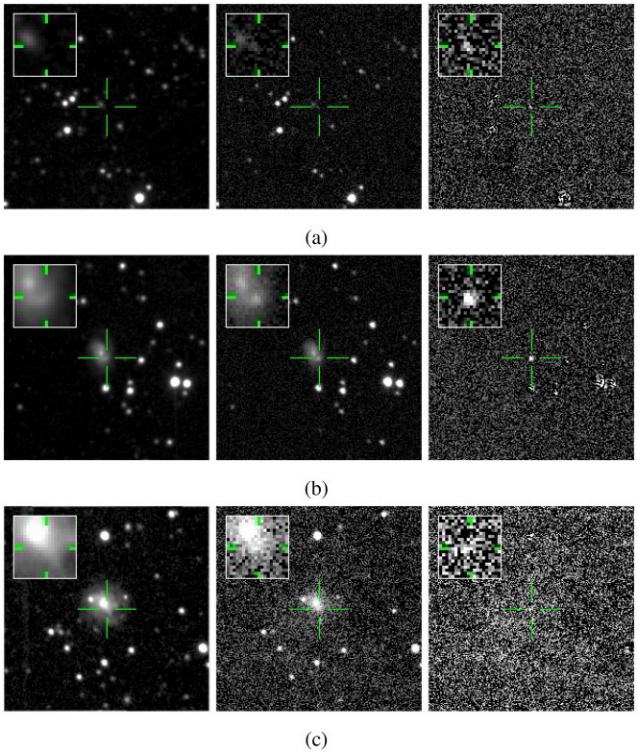

Figure 1. Real astrophysical transients detected via ATLAS difference imaging. From left to right: detection image, reference image, and difference image. 20 x 20 pixel stamps are displayed in the top left. In classification, we only utilize the 20 x 20 core of the difference image.

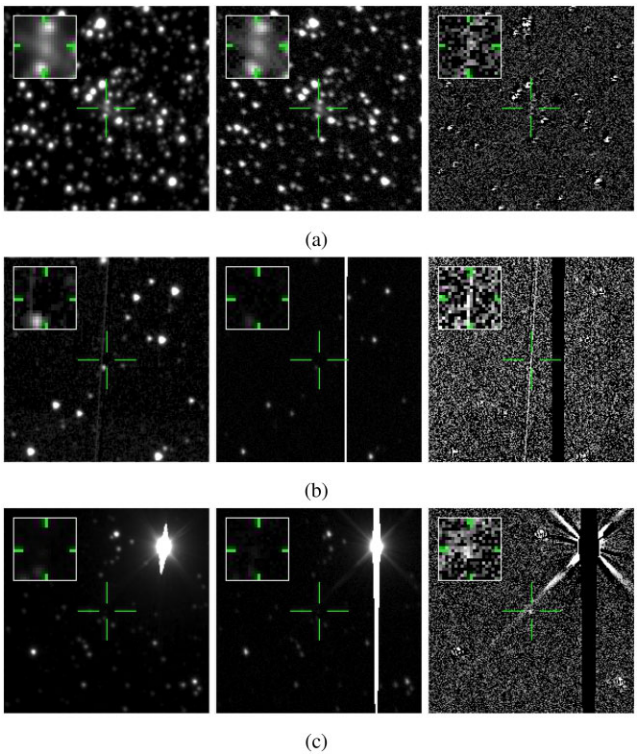

Figure 2. Bogus astrophysical transients detected via ATLAS difference imaging. From left to right: detection image, reference image, and difference image. 20 x 20 pixel stamps are displayed in the top left. In classification, we only utilize the 20 x 20 core of the difference image.

3 THE ATLAS CLASSIFIER

3.1 Convolutional neural networks

In the early stages of transient searches with ATLAS, we utilized a random forest and sparse-filtered neural network algorithm as outlined in Wright (2015) and Wright et al. (2017), before moving in 2018 September to a Keras CNN (Chollet 2017) which other transient detection surveys had utilized successfully (Cabrera-Vives et al. 2017; Reyes et al. 2018). The CNN in use by ATLAS is a modification of an existing example model used to classify images of handwriting digits in the MNIST data base, which we have repurposed for use on our difference images (Smith et al. 2020). The model architecture, summarized in Fig. 3 and A1 in the appendices, is as follows:

(i) A two-dimensional (2D) convolutional layer with a 2×2 kernel size and 16 filters utilizing rectified linear units (ReLUs). The purpose of the convolutional layer is to perform a dot product between the kernel, which is a matrix of the model's learnable parameters, and the input image. The kernel 'passes' across the full height and width of the image to produce a 2D activation map providing a response of the kernel at each position of the input matrix. The ReLU sets the negative activation map values to zero while leaving positive values unchanged. This transformed activation map is passed on as the input to subsequent layers. ReLUs allow us to achieve non-linear data transforms with the aim of making the transformed data linearly separable. The simplicity of the function allows fast computation in a many-layered CNN with the added benefit of maintaining a constant gradient of 1 for its positive inputs, preventing the degradation or vanishing of gradients or features through the multiple layers. In removing less informative or negatively influencing input from the activation map ReLUs introduce sparsity that allows for a more computationally efficient CNN. We use 'Same' padding throughout each convolutional layer in our model (Dumoulin & Visin 2016). Without padding, the kernel is only applied to the centre pixels of the input, producing an activation map with a smaller size than the input. In Same padding additional pixels or padding is generated around the input data before convolution, to ensure that the kernel can be centred over each input pixel. The purpose of the padding is to preserve the complete spatial information of the image through multiple convolutional layers. With 16 filters we obtain 16 distinct activation maps corresponding to individual recognized features within the input.

(ii) A max pooling layer for the 2×2 pixel regions (Yamaguchi et al. 1990). The pooling layer reduces the spatial size of the representation and thus the required amount of computational power. In the case of max pooling the maximum output from the (2×2) kernel size is reported.

(iii) A repeated 2D convolutional layer with the same kernel size but 32 filters, followed by an identical max pooling layer. In introducing multiple convolutional layers, we permit a hierarchical feature extraction wherein the CNN learns progressively complex features with the addition of each layer, beginning with features such as edges and gradients before moving to individual objects or object parts within the image and their corresponding individual features. An increased number of filters allows us to capture more of these features while maintaining a high level of dimensionality and complexity. It should be noted that introducing a greater number of filters and convolutional layers, and therefore a greater number of abstract features, diminishes our ability to interpret the model. As the level of abstraction increases and the CNN generates a larger number of activation maps it becomes difficult to analyse output layer-by-layer, leaving us instead to focus on the machine's final decision.

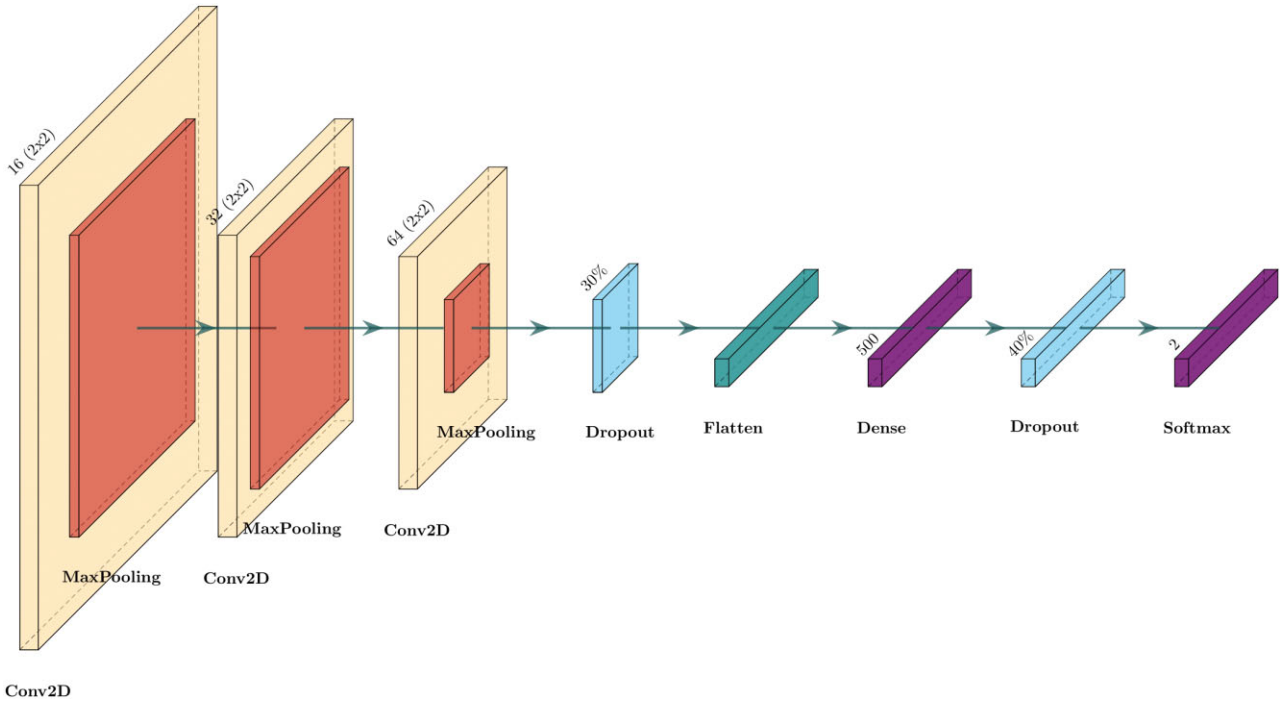

Figure 3. Utilized CNN architecture for the ATLAS RB classifiers.

(iv) A third convolutional layer, now with 64 filters, and a third identical pooling layer.

(v) A 'flattening' layer which transforms the 2D tensor output of the third layer to a 1D tensor. Subsequent fully connected layers require a 1D input where each element corresponds to a separate neuron.

(vi) A dropout layer with a dropout rate of 0.3. A random 30 per cent of inputs are set to 0 at each step during training to prevent overfitting. Inputs not set to 0 are scaled up by $1/0.7 = 1.43$ such that the sum of the input layer remains identical. With the goal of a CNN to be to optimize the weights of the neural network to accurately represent features provided in the training data, it is possible for all neurons in a given layer to synchronize their weights and converge to the same goals. To prevent overfitting, dropout prevents this synchronous optimization and encourages learning based on a more diverse pool of features. The reduction in training required at each iteration can assist in the development of more efficient and less computationally intense machines.

(vii) A dense or fully connected layer with 500 ReLUs. In this context, fully connected refers to each neuron of the layer being connected to every neuron of the preceding layer. In this, we combine the features extracted by our previous layers and allow the CNN an opportunity to examine further complex patterns within the data.

(viii) A second dropout layer with rate 0.4.

(ix) A final dense output with a softmax classifier (Bridle 1990). This second dense layer can refine the features identified by the first to focus on the most relevant neurons for classification and, as with the convolutional layers, identify more abstract features within the initial input image. The softmax activation function scales the output logits, or unnormalized predictions, to a vector with the probability of the input image belonging to each outcome. The first vector component is the probability of a given input being a real detection; the second is the probability of that same input being bogus.

We train the model with categorical cross-entropy loss (Hastie et al. 2017) and utilize the 'Adam' optimization routine which is recognized to outperform other optimization techniques for large data sets and achieves fast optimization (Kingma & Ba 2014; Ruder 2016). The purpose of the cross-entropy loss is to measure the distance between the output prediction values S and the true values T:

$$L_{\mathrm{CE}} = \sum_{i=1} T_i \log(S_i).$$

With the aim of minimizing the loss, the softmax function is continuously differentiable, allowing us calculating the loss function derivative with respect to every weight in our CNN.

3.2 The RB factor

RB populations are separated through the use of the machine learning classifier score or a RB factor. Models are trained such that a low RB factor denotes a high probability of a detection being spurious, while a high factor denotes a high probability of a detection being genuine, with the overlap between the two populations in the centre of the score range being smaller for better-trained classifiers. In our context, any objects with a score above a set threshold are passed on to humans for validation. This threshold is set to balance the completeness of the real population with the purity of the subdata; in our current working system, we are set to select objects with RB ≥ 0.2 to provide 96 per cent completeness (Wright et al. 2017; Smith et al. 2020).

In this report, we refer to the false positive rate (FPR) and the missed detection rate (MDR, or false negative rate) in reference to score threshold (Wright et al. 2015):

$$FPR = \frac{FP}{FP + TN} \times 100 \text{ per cent}$$
$$MDR = \frac{FN}{FN + TP} \times 100 \text{ per cent}.$$

The former refers to the percentage of bogus detections [FPs and true negatives (TNs)] selected as real by the classifier for scanning,

while the latter refers to the percentage of real detections [false negatives (FNs) and true positives (TPs)] erroneously selected by the machine as bogus. In comparing the performances of our classifiers, it is important to refer to a figure of merit, whether by fixing our FPR or MDR and observing changes to the other rate (Brink et al. 2013). A lower FPR at a fixed MDR brings the benefit of reducing the number of bogus images passed to human scanners, reducing the workload provided.

3.3 Data

We use the PSAT-ML Python repository[1] which creates a pipeline that connects to the ATLAS data base to build a set of difference images and train a corresponding classifier. We seek to build data sets for the HKO, STH, and CHL telescopes. The eruption of Mauna Loa in 2022 November resulted in a pause in MLO telescope operations for 12 months, and as a result we do not retrain the relevant classifier at this time.

We built a data set for each telescope with an initial 1:3 ratio of real to bogus objects. Ten nights typically provide a substantial number of real detections ($\sim$300 000). We selected 10 nights from the ATLAS data store to equally cover orange and cyan filter observing five nights each and obtain known real objects for each. Known real objects refer to a population of known asteroids identified via ephemeris checks. We augment this population with objects identified as real, stationary, transient sources by human scanners (although these are relatively small in number compared with the known asteroids, with the total number of confirmed detections totalling 33 000). To meet our requirement for bogus objects, we pull data from 200 nights between 2022 May and October. These 'bogus' objects consist of detections rejected by human annotators and those rejected by the previous classifier, having fallen below the RB threshold.

All 'bogus' detections have undergone the same selection criteria as 'real' ones. They must be 100 pixels or more from the chip edge, be positive flux, part of a triplet of co-spatial detections, be regarded as variable or transient, not crosstalk, not a known mover and not a burn or a scar.

After the selection of our real and bogus detections, the distribution of these on each camera is shown Fig. 4. It demonstrates that there are detector-dependent spurious detections and that the real objects have no spatial dependence. The plots (which we internally refer to as heat maps) show that the good sources are distributed uniformly across the detectors and the bogus objects, we used, have strong detector-dependent characteristics.

We are left with a data set that for each telescope contains $\sim$ 25 per cent asteroids, $<$ 1 per cent real stationary transient objects, and 75 per cent bogus detections. We use a split of 75 per cent when building our training and testing sets, ensuring that detections from the same object are not divided between the sets, and are left with a data set containing approximately 1300 000 detections for training (see Table 1). We highlight the use of known, catalogued asteroids which serve as a large sample of astrophysically real transient sources which have single PSFs. They do not move substantially within the 30-s exposures to affect the PSF shape.

4 RETRAINING

We train the classifiers with the same CNN architecture as outlined in Section 3. We train over 20 epochs, selecting this as an optimal

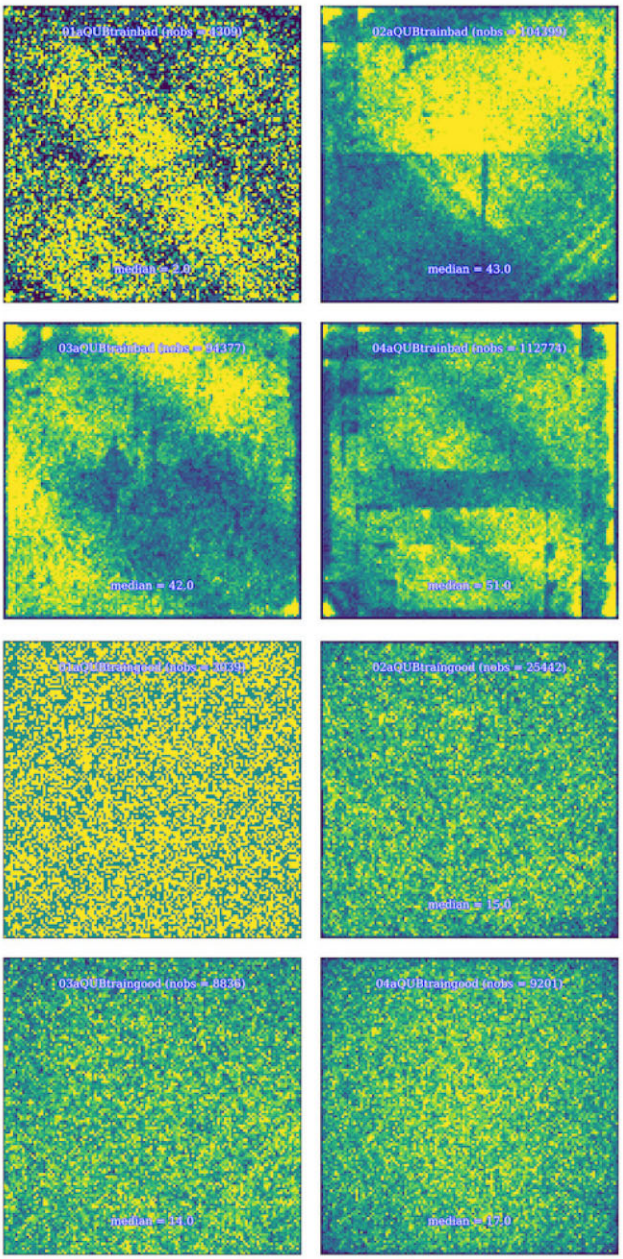

Figure 4. Heat maps showing the spatial distribution of the bogus detections used for our training on each of the detectors (top four) compared with the astrophysical real detections used as good detections (bottom four).

Table 1. Object composition of classifier training sets. Transients refer to astrophysically real, stationary transient sources vetted by human scanners.

Data set	Transients	Asteroid	Bogus
23HKO	27 567	302 433	990 000
23STH	4239	315 761	960 000
23CHL	1428	378 572	1140 000

[1] https://github.com/genghisken/psat-ml

number beyond which the categorical cross-entropy loss plateaus for the validation data set (itself a 25 per cent subset of the training data, maintained for each epoch of training). We save the optimal model of these 20 epochs based on the minimum loss. When utilizing the resulting classifier, we output two figures of merit for each set (training, testing, and validation) of each classifier data set: first, we calculate the FPR corresponding to the operating point where the MDR is equal to 1 per cent on the receiver operating characteristic (ROC) curve (Brink et al. 2013; Hastie et al. 2017). Inversely, we also calculate the MDR corresponding to the point on the ROC curve where the FPR is equal to 1 per cent.

Retraining took place in the first half of 2023. We therefore refer to the new classifiers and their corresponding training data as 23HKO in the case of the Haleakala telescope, 23CHL in the case of the Chilean telescope, etc. We seek to compare the latest classifier performance with that of the classifiers from the last retraining epoch in 2020. We refer to the two classifiers and their corresponding training data as 20HKO and 20MLO.

A complete set of training, validation, and testing metrics for each telescope's optimal classifier can be found in Tables A1, A2, and A3 (see Appendices). We find improvements in both the FPR and MDR across all data subsets for each classifier compared with our old models. Promisingly we already see a marked improvement in our STH and CHL classifiers which previously lacked bespoke models. In particular, we examine our FPRs for a 1 per cent fixed MDR. In our testing data, we obtain (relative) decreases to the FPR of 75.8 per cent, 77.5 per cent, and 86.3 per cent for the 23HKO, 23STH, and 23CHL classifiers, respectively, in comparison with the 20HKO classifier, the 20MLO classifier on 23STH data, and the 20MLO classifier on 23CHL data.

We have used hold-out validation in our retraining with a single training and testing set for each classifier (Arlot & Celisse 2010). Although cross-validation is a valuable technique for enhancing performance stability and accuracy, the data set's size, in this instance, offers a reliable estimate of classifier performance or does it?.

We compare the performances of the old and retrained classifiers on recent data in Fig. 5 and Table 2, using our figure of merit, the FPR for a fixed MDR of 1 per cent. In addition to this, we measure the area under the receiving operating curve (ROC-AUC); a plot of the TP rate against the FPR. As the AUC tends to be 1, we see the TP rate approaching 1 and the FPR tends to 0; thus, the higher the AUC, the better the performance. We should expect to see our greatest improvements in the CHL and STH classifiers given the previous lack of a specialized classifier for these telescopes:

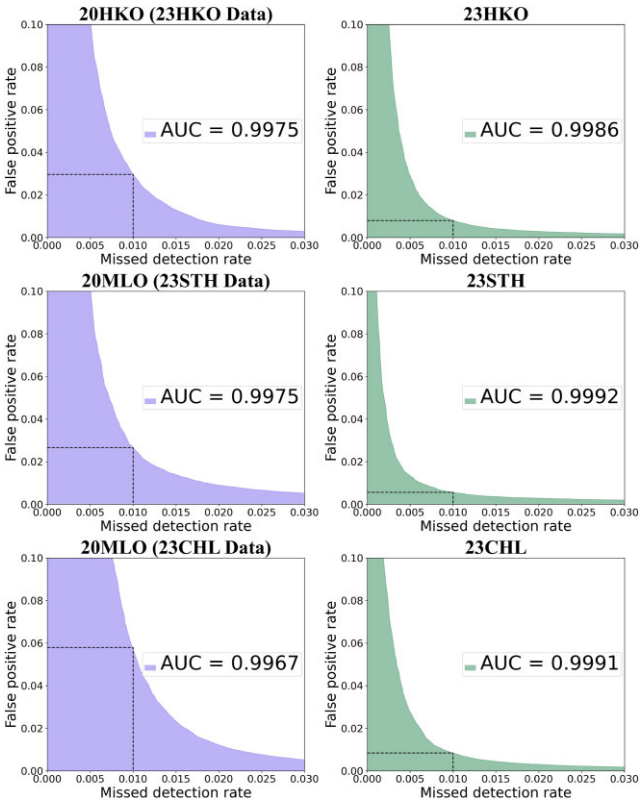

Figure 5. FPRs for 2020 classifiers (left panel) and 2023 classifiers (right panel) for a 1 per cent MDR.

Table 2. Performance metrics of classifiers on complete data sets. FPR refers to FPR value when we fix the MDR at 1.0 per cent.

Telescope	Classifier	Data	FPR (per cent)	ROC-AUC	RB threshold
HKO	2020	2020	4.16	N/A	0.055
	2020	2023	2.97	0.997 487	0.074
	2023	2020	12.8	N/A	0.040
	2023	2023	0.72	0.998 640	0.182
STH	2020	2023	2.67	0.997 511	0.157
	2023	2023	0.60	0.999 172	0.316
CHL	2020	2023	5.79	0.996 669	0.060
	2023	2023	0.79	0.998 919	0.202

(i) For HKO, we see an improvement in the FPR from 2.97 per cent (for 20HKO) to 0.72 per cent (for 23HKO). Being the only operating telescope that has had a previously specialized classifier trained on its data, we should not expect too significant improvement in performance; however, HKO provides us with a means to inspect our new 23HKO model performance on the older 20HKO data. On the 20HKO data, our new model has an FPR of 12.8 per cent, in comparison to the 20HKO model's FPR of 4.16 per cent (see Table 2). As the telescope hardware (detector, Schmidt corrector, and focus model) has evolved over time, we might expect the old classifier to outperform the new one when applied to the older data. It is encouraging to see that the 23HKO model applied to contemporary data shows a much lower FPR than the old model on old data. This may be in part attributable to the previously mentioned improvements to the training data through the addition of corrected classifications made by human scanners. The ROC-AUC has seen an increase from 99.7 per cent to 99.9 per cent.

(ii) For STH, we see an improvement in the FPR from 2.67 per cent (for 20MLO) to 0.60 per cent (for 23STH). The Sutherland telescope had previously seen good performance using the 20MLO classifier (having the lowest FPR of the three telescopes currently in operation) and has maintained its lead following retraining. The AUC sees an increase from 99.8 per cent to 99.9 per cent.

(iii) For CHL, we see an improvement in the FPR from 5.79 per cent (for 20MLO) to 0.79 per cent (for 23CHL). As expected, this is a significant improvement in comparison to the HKO classifiers. The AUC sees an increase from 99.7 per cent to 99.9 per cent.

Our ultimate aim is reducing the number of bogus images passed on to human scanners while maintaining a fixed MDR (Fig. 6). For

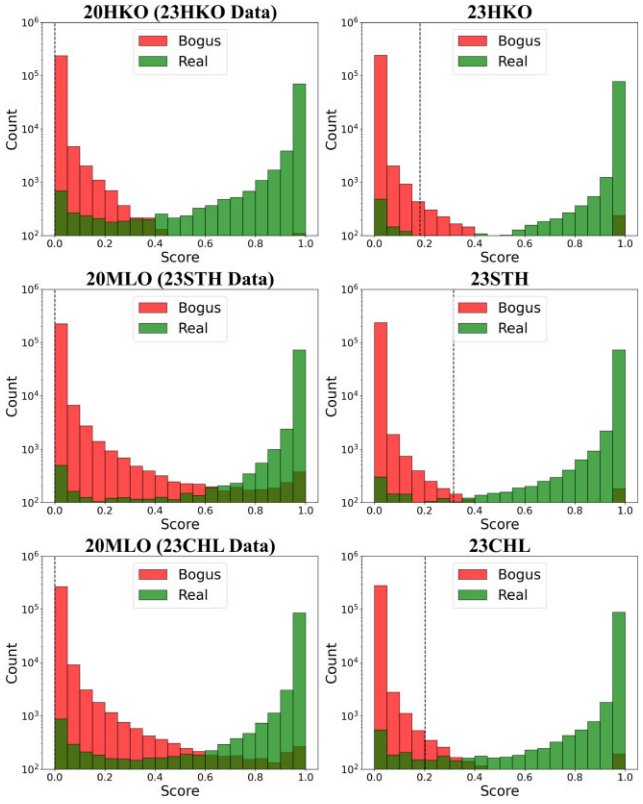

Figure 6. Class distributions for each classifier. The new classifiers separate the distributions to a greater extent than their older versions. Corresponding changes to the RB factor threshold are minimal.

our CHL classification, we have reduced the FPR by almost a factor of 4. For the 1140 000 bogus detections in our training data, we would have previously seen 66 006 passed on for human scanning by our old classifier given a score threshold of 0.2. For our new classifier that number is reduced to 9006; a significant real-term improvement.

It is worth investigating the performance of the new classifiers on corresponding objects in the good list (= 27 174). Of these objects, 112 fall below the old RB threshold of <0.2. Ignoring objects classified by the 20MLO model, we note 14 significantly misclassified detections. Of these, five were misclassified by the old models and subsequently recovered by human scanners following detections by other surveys. Several of the object scores may also be impacted by nearby bad pixels in the stamps. Inspecting 20 000 recent bogus objects, we find 215 with RB factors above the threshold, eight of which are scored >0.9 by the new classifiers. Of these, two are real and six are due to residuals from bright star subtraction. We can expect low 'swap rates' between the real and bogus populations owing to the nature of the training set building process. Our training bogus objects are bootstrapped by the existing machine learning scores; objects that are real in nature but scored below the threshold by our classifiers and are not subsequently recovered will encourage our new classifiers to disregard similar detections in the future. Likewise the classifier weighing of features when selecting good objects is biased to those the previous classifiers placed importance on. In our Appendix, Tables A1, A2, and A3 display training, testing, and validation set metrics. We do not see evidence of overfitting when comparing our FPRs between the training and testing sets, with higher rates in the latter. We can see this is an acceptable degree of overfitting; the test set errors remain lower for the new models, but it is important to note the impact our data selection has had on training.

Table 3. Retraining times for each classifier.

Classifier	User time	System time	CPU (per cent)	Total
23HKO	16 745	22 189	250	4:19:03
23STH	16 778	23 127	458	2:25:01
23CHL	20 899	28 312	456	2:59:39

The computational expense of retraining each classifier is low. On an 8-core Apple M1 Pro CPU, each neural network can be trained in as few as 4 h as shown in Table 3. Incorporating into this process the steps of selecting nights for data, building a data set, and implementing the new model the timeframe of retraining and implementation for each classifier can be under a day. This as well does not consider the possibility of retraining multiple classifiers at once on a larger machine.

With the benefits of regular model retraining established it is important to consider the frequency used. In the case of the ATLAS-CNN suite, previous models show a reasonable performance prior to retraining. At current FPRs, it is unlikely that the use case of the classifiers will require further improvements to performance unless significant divergence from the expected FPR is observed in live. While it is feasible to retrain the classifiers as frequently as every 6 months, the trade-off between labour and performance improvement should be taken into account. As a result, we elect to examine the need for retraining on an ad hoc basis following changes to optical hardware for each telescope.

5 IMPROVEMENT TESTS: VARIATION OF INPUT

5.1 Variation of image size

While the use of recent, camera-specific data in the retraining of the classifiers already yields decrements in FPRs, it is worth also considering changes to the input data as a means of improving performance. We identify two significant opportunities for improvement. The first is the reduction of the image dimensionality, which may carry several benefits: first, the associated reduction in the amount of data yields a greater computational efficiency in training and classification. In addition to this, the reduction in data may reduce the complexity of the model image, thus decreasing the risk of overfitting and allowing the classifier to more easily generalize to new data. Perhaps most significant in the transient detection use case is the reduction in noise: in difference images, we find artefacts and blurring where the image subtraction was poor. While poor difference images may be indicative of a bogus detection, the associated artefacts may be found in images containing real objects, which could lead the classifier to incorrectly associate such features with transients. Supernova-like transients may be near bright galaxies leaving subtraction residuals in their core upon creation of the difference image. In reducing image dimensionality, we also reduce the number of such artefacts in each input. While the ATLAS data pipeline also allows us to increase the dimensionality of the input, the size of the associated detection is small in comparison to the size of the entire image (∼5 pixels). In increasing the number of pixels, we increase the likelihood that the classifier will learn fewer features from the detection itself. There is also the associated decrease to computational efficiency. Our classifier must be capable of processing thousands of detections each night. Our current classifier input consists of 20 × 20 pixel images – in increasing the size to 30 x 30 pixels, we more than double the size of our input for sky data that will not contain information

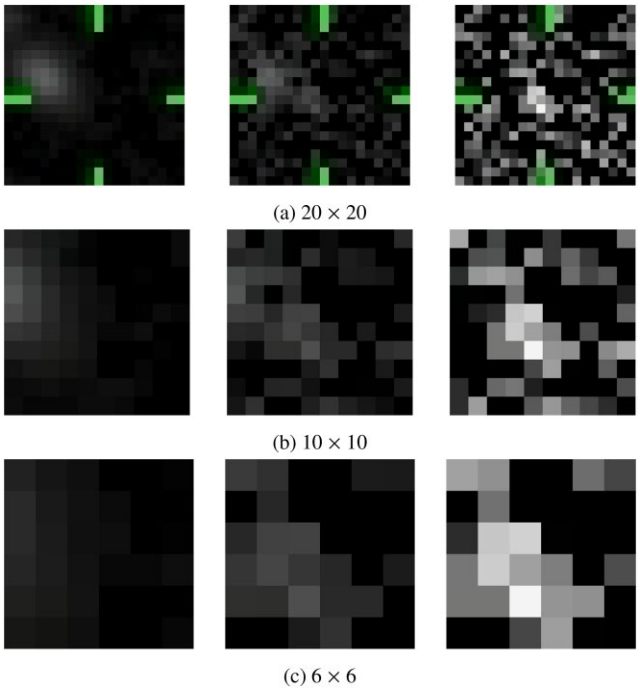

Figure 7. Varying image size for a real astrophysical transient. From left to right: detection image, reference image, and difference image.

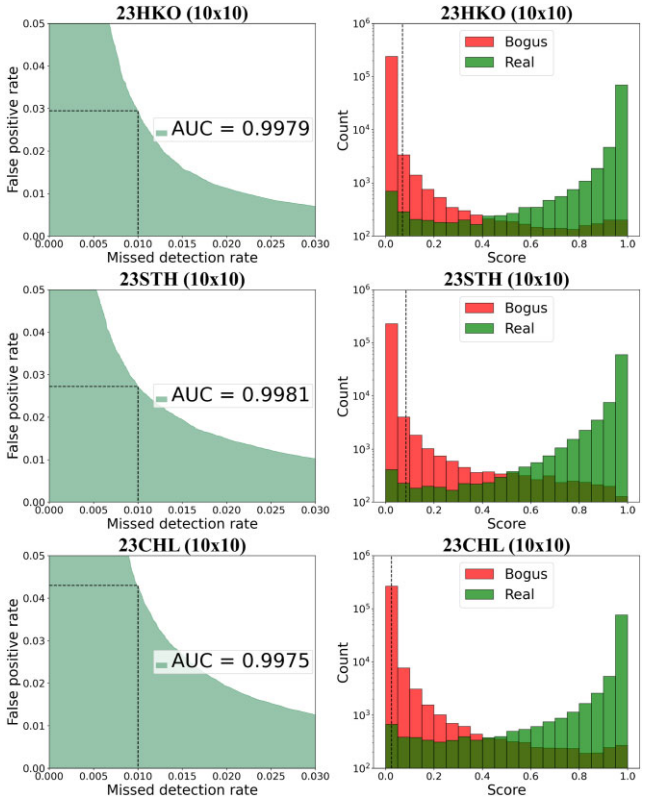

Figure 8. FPRs for the 10 × 10 2023 image classifiers for a 1 per cent MDR.

about the detection itself. As such it is sensible to decrease the image size to 10 × 10, which focuses the model to learn more from the region immediately surrounding the potential transient and quarter the number of pixels inputted to improve computational efficiency (Fig. 7).

We attempt to retrain, in parallel, our 2023 classifiers with 10 × 10 and 6 × 6 pixel images and inspect for improvements in the FPR and ROC-AUC (Table 4, Fig. 8). We find that for 10 × 10 images, in the case of all three classifiers, the FPR is significantly worse with a median increase by a factor of 3.55. We can consider the typical use case of CNNs as to why this may be. Typically CNNs are not designed for inputs smaller than 28 x 28 pixel images found in common data sets such as MNIST, CIFAR-10, and CIFAR-100 (LeCun & Cortes 2005; Krizhevsky 2009; Sharma, Jain & Mishra 2018). Indeed such data sets contain RGB images, greatly increasing dimensionality against our 20 × 20 grey-scale images. The smallest commonly used grey-scale image data set, and indeed the one used to build the example classifier we base our own model architecture from, is the MNIST data set of handwritten digits in 28 x 28 pixel images. A 28 x 28 grey-scale pixel image provides approximately twice as much information for a CNN compared with a 20 × 20 input. A 10 × 10 image provides four times less. We can consider the lost information from any noise or artefacts by revisiting our RB ratio as described in the initial retraining. We select a ratio of 1:3 real to bogus detections for two primary reasons: first, we reflect the in-live discrepancy between the proportions of each detection class while also maintaining a high enough percentage of real detections for the CNN to learn their associated features. Secondly, in maintaining a higher proportion of bogus detections we train the classifier to tend towards more conservative classification. The purpose of each classifier is to reduce the number of FPs passed to human annotators. By providing the classifiers with more bogus data, we aim to create a model that is better trained to identify features of bogus detections.

Table 4. FPRs of classifiers on larger images versus smaller images. FPR refers to the FPR value when we fix the MDR at 1.0 per cent.

Data set	20 x 20 (per cent)	10 x 10 (per cent)
23HKO	0.72	2.95
23STH	0.60	2.73
23CHL	0.79	4.30

In reducing our image size, we limit the ability of a classifier to learn or identify these features.

For training the CNN on 6 × 6 pixel images, we find the use of successive convolutional layers by the classifier reduces the input for each layer to such a degree that the model is unable to fully process the detection. The output shape of each maxpooling layer is halved – in the case of the 20 × 20 input, we have a 10 × 10 output from the first instance, 5 × 5 from the second, and 2 × 2 from the third. For an initial 6 × 6 pixel input, we then have a 3 × 3 output following the first instance of maxpooling and 1 × 1 following the second, leaving us unable to continue any feature extraction. When using CNNs on small images, it is vital to take into consideration the reduction in dimensionality that takes place within the model layers. In the case of our 10 × 10 pixel images, the output following the convolution and maxpooling layers (and preceding the dropout and fully connected layers) consists of 64 single pixel pooled feature 'maps'.

6 IMPROVEMENT TESTS: VARIATION OF THE MODEL

We have so far examined avenues for performance improvement via the updating of our training data and adjusting our input images.

A third possibility is to modify the model architecture itself. The reduction in the FPR following our initial training indicates a change to the data or optical performance over time, including new features that may not have been easily learned by the previous classifier. Improvements made to the data reduction pipeline prior to classification may have introduced more complex (or more simple) features. It is worth investigating if a different model architecture could extract such features more easily.

We compare the FPRs regarding the 20 × 20, 10 × 10, and 6 × 6 pixel image 23HKO data sets for five different classifiers, using the FPR of our default retrained classifier architecture (D0, as used in 23HKO) as a benchmark (1.11 per cent). Following the discussions regarding dimensionality reduction within the model in Section 5, we attempt three variations of our default model. The first drops the final 64-filtered convolutional layer and subsequent pooling layer (D1). The second drops the preceding 32-filtered convolutional layer and subsequent pooling layer (D2). Having seen the effect of dimensionality reduction through successive convolutional and pooling layers for small image inputs it is worth examining shallower models with fewer reductions. While any loss of dimensionality will be reduced for the former, and even more so for the latter, a reduction in convolutional layering inhibits a CNN's ability to extract complex features. While the use of additional convolutional and pooling layers was explored, we decided that the loss of information was too great for the resulting classifier to be of any benefit. We also compare performance with a non-CNN; we utilize the fully connected dense layer followed by our second dropout layer (D3). In such a classifier, we should expect to see little loss in information but a significant decrease in the model's ability to extract complex features. Finally, we compare our model performance to two well-known lightweight CNNs. MobileNets are a series of CNNs developed by Google, Inc. that utilize depthwise convolutional layers (Sifre & Mallat 2014; Howard et al. 2017). A depthwise convolutional layer carries output dot products between kernels and each individual channel of the input image. For example, in the case of an RGB image, different activation maps are provided for each kernel for the red, green, and blue channels. For a grey-scale image, a convolutional filter is applied to the single channel as with our Conv2D layer. A depthwise layer also utilizes a pointwise convolution which applies a 1 x 1 filter to combine the outputs from each channel. In our case, this pointwise convolution simply applies a 1 x 1 filter to the output. We utilize a variation of the MobileNet architecture as follows:

(i) A Conv2D layer with a 3 × 3 kernel size and 32 filters.
(ii) A DepthwiseConv2D layer with a 3 × 3 kernel size. This layer neglects the pointwise convolution, so we use a Conv2D layer with a 1 x 1 kernel size and 64 filters.
(iii) A dropout layer with a dropout rate of 0.25.
(iv) Another DepthwiseConv2D and Conv2D pairing with 128 filters, and another dropout layer with a dropout rate of 0.25.
(v) A GlobalAveragePooling2D layer (Lin, Chen & Yan 2014). Global average pooling differs from maxpooling in that it takes the mean value for a given pixel in each feature map; i.e. the average across the channel. This pooling technique is useful in cases where the spatial position of features in the image is less important, and serves to increase the efficiency of the classifier.
(vi) A final dense output with a softmax classifier.

We also utilize the LeNet-5 architecture developed by LeCun et al. in 1998 (Lecun et al. 1998, 1995). This LeNet-5 model is one of the oldest lightweight CNNs developed and was trained on the MNIST handwritten digits image data set (LeCun & Cortes 2005). The model consists of two Conv2D layers with 3 × 3 kernel sizes, the first with six filters and the second with 16. After each Conv2D layer, we use an AveragePooling layer. Rather than outputting the average of the entire activation map, AveragePooling pools across 2 × 2 pixel regions. The model then flattens the output in advance of two dense layers, the first with 120 ReLUs and the second with 84. The output is provided by a third dense layer with a softmax classifier.

Finally, we examine a MiniVGGNet architecture (Simonyan & Zisserman 2015; Rehemtulla et al. 2024). With this architecture, we duplicate each convolutional layer prior to pooling for three successive blocks of two convolutional layers and one max pooling layer. We remove the dropout layer prior to the flattening and dense layers, keeping one 50 per cent dropout layer prior to the softmax classifier.

We train the five models with each of our three data sets, providing fifteen classifiers in total.

Table 5. Performance metrics of classifiers on 23HKO data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent.

Classifier	Image size	FPR (per cent)	ROC-AUC	RB threshold
D0	20 x 20	0.72	0.998 640	0.182
	10 x 10	2.95	0.997 859	0.069
	6 x 6	N/A	N/A	N/A
D1	20 x 20	0.75	0.998 754	0.196
	10 x 10	3.47	0.997 680	0.073
	6 x 6	19.4	0.986 944	0.066
D2	20 x 20	1.05	0.998 592	0.259
	10 x 10	2.81	0.997 666	0.160
	6 x 6	18.8	0.986 830	0.057
D3	20 x 20	4.42	0.995 773	0.045
	10 x 10	6.31	0.995 254	0.038
	6 x 6	21.5	0.984 561	0.024
MobileNet	20 x 20	1.47	0.998 268	0.142
	10 x 10	3.23	0.997 731	0.097
	6 x 6	N/A	N/A	N/A

6.1 Model comparison

For our variation on D0 with two convolutional layers, respectively, we see an increase in the FPR to 0.75 per cent, 3.47 per cent, and 19.4 per cent for the 20 × 20, 10 × 10, and 6 × 6 23HKO data sets (Table 5, Fig. 9). While only a slight difference is observable between the 20 x 20 images, a reduction in the classifier ability to extract useful classification features can be inferred; as well as this, the increase in FPR as the image size decreases indicates that the loss in features is more greatly felt as they are cropped from the data. Likewise for the variation with one convolutional layer, we see an increase in the FPR to 1.05 per cent, 2.81 per cent, and 18.8 per cent. In the case of our lone fully connected layer model the performance is again impacted by an inability to extract complex features, with a median FPR of 6.31 per cent. Notable is the comparison of our 23HKO classifier performance with that of MobileNet. For the data set, we find an FPR of 1.47 per cent – a rate approximately twice as great as our own. There are several possible explanations for our classifier to outperform MobileNet. While both are structured to contain three consecutive convolutional layers, MobileNet neglects to pool until after all convolution has completed. The resulting loss of spatial information may then be responsible for inefficient feature extraction. Alternatively, the loss of dimensionality in the application of a 3 x 3 kernel size rather than 2 x 2 as in the default model may be to blame. The output feature maps the following: the final convolutional layers

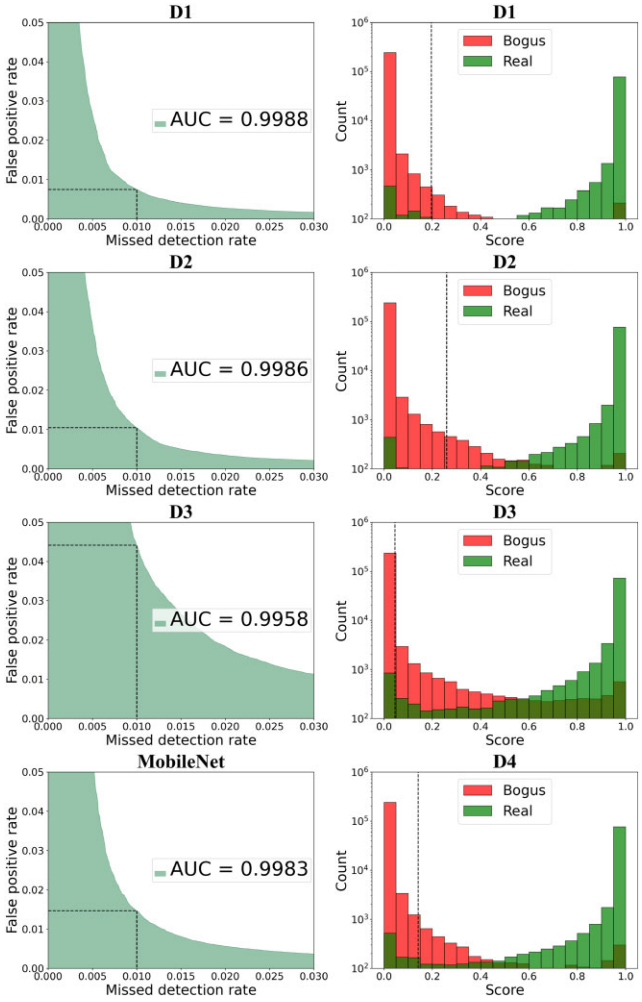

Figure 9. FPRs for various classifiers on the 20 × 20 23HKO data, for a 1 per cent MDR.

are 5 x 5; while these are the same dimensions as our classifier, same padding is used in both. A larger kernel size for a small input will reduce the ability to capture smaller features and detail in the output. The ability of the current ATLAS classifiers to outperform a known well-performing CNN is promising; however, MobileNet is designed with larger multi-channel images in mind, and it is important to make comparisons with a more similar classifier.

The performance of LeNet-5 on the 20 × 20 23HKO data set is similar to our default classifier with an FPR of 0.87 per cent (Table 6, Fig. 10). Having been developed with 32 × 32 grey-scale input images, it is natural to expect good performance on our own 20 × 20 grey-scale input. The ability of LeNet-5 to approach D0 on 23HKO data warrants exploration with the 23STH and 23CHL data, providing us again with higher FPRs for each (0.76 per cent for the former and 1.02 per cent for the latter). LeNet-5 uses fewer filters than D0 in its feature maps, which may result in the selection of more prominent features. Further to this, the use of an additional fully connected layer allows the model to further refine the combined features interpreted by the first dense layer. Finally the use of average pooling as opposed to our MaxPooling layer provides 3 x 3 feature maps prior to flattening rather than 2 x 2, maintaining a greater level of spatial information.

The comparable performance but differing approaches in D0 and LeNet-5 provide us with an opportunity for improvement. The application of a second dense layer of 250 units to D0 achieves for 23HKO an FPR of 0.77 per cent, for 23STH an FPR of 0.59 per cent, and for 23CHL an FPR of 0.79 per cent, respectively. These are an improvement on the LeNet-5 FPRs and almost achieve parity with the default model performance.

Use of the MiniVGGNet architecture yields the strongest performance on the HKO data outside the default model (FPR = 0.73 per cent) and also warrants further investigation (Fig. 11, Tables A4, A5 and A6 in the appendices). We see on 23STH (0.59 per cent) and 23CHL (0.75 per cent) that the MiniVGGNet model outperforms the default model. This comes with some caveats: first, the greater number of layers without pooling increases the training time to approximately 18 h. The greater number of features and data processed by the model may impact operation of the transient detection pipeline and warrants further investigation prior to implementation. Secondly, the use of successive convolutional layers and subsequent increase in the number of features decreases the interpretability of the model. We elect to keep the D0 architecture rather than utilize the MiniVGGNet in our classification pipeline. While continued adjustments to architecture may yield improvements, the benefits observed are telescope-specific and minimal when compared with the improvements already achieved in the retraining. Further refinement runs the risk of diminishing returns where effort could be expended on other approaches such as contextual classification and non-pixel learning. The increased number of layers for little gain also runs the risk of overfitting on current data behaviours. We have seen that our data changes over time owing to changes made in the optics, detector, or data reduction processing. Use of a lightweight CNN provides agility and adaptability to new trends, which may allow for a greater period of time before retraining is necessary. From comparison with these other models, we may state that the D0 architecture provides a competitive performance to other common CNNs while maintaining a robustness against overfitting.

7 OPERATIONAL PERFORMANCE

The 2023 classifiers were implemented within the ATLAS pipeline on 2023 September 6, replacing the previous 20HKO and 20MLO models. Fig. 12 displays the volume of detections processed by the implemented classifiers over time, with 2020 classifiers processing data prior to the implementation of the 2023 classifiers. We also examine the volumes and rates of FPs, missed detections, and TPs over time.

Prior to reimplementation, we maintained a score threshold of 0.2 for objects, for a corresponding expected MDR of 2.0 per cent. While some classifiers in the suite outperform others, an overarching threshold is currently applied across all models as detections are combined from multiple telescopes to create an object. In selecting a new threshold, we examined individual classifier performance on the training and testing data for expected MDRs of 1.0 per cent and 0.5 per cent (Table 7). The threshold for the latter = 0.038 demonstrates the improved performance of the retrained classifiers; significantly decreasing both the FPR and MDR at this end of the score range. For 1.0 per cent MDR, it is sensible to maintain the current threshold of 0.2. We see improvements in both the FPRs and MDRs at this threshold. While the 0.038 threshold sees strong improvements between 2020 and 2023, the FPR remains higher than our current implemented pipeline. To avoid the increase in the workload of the human annotators, we maintain a threshold of 0.2 in the implementation of the new classifiers.

Summary statistics for Fig. 12 can be found in Table 8 owing to the noise of the plotted data. The date utilized is the date the

Table 6. Performance metrics of LeNet-5 and MiniVGGNet Classifiers. FPR refers to the FPR value when we fix the MDR at 1.0 per cent.

Model	Data set	Image size	FPR (per cent)	ROC-AUC	RB threshold
LeNet-5	23HKO	20 x 20	0.87	0.998 717	0.209
	23STH	20 x 20	0.76	0.999 172	0.304
	23CHL	20 x 20	1.02	0.998 895	0.229
MiniVGGNet	23HKO	20 x 20	0.73	0.998 764	0.288
	23STH	20 x 20	0.59	0.999 284	0.456
	23CHL	20 x 20	0.75	0.998 798	0.242

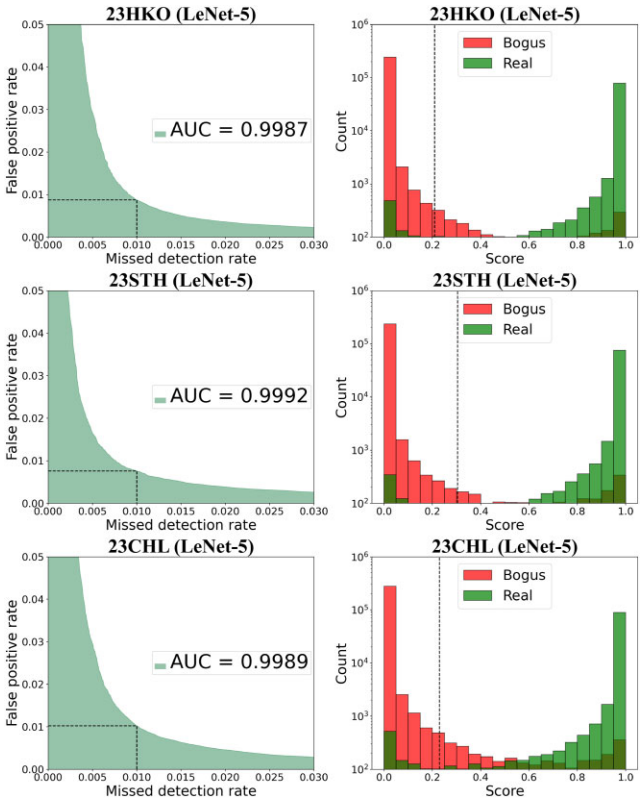

Figure 10. FPRs for the LeNet-5 classifier on 20 × 20 2023 data, for a 1 per cent MDR.

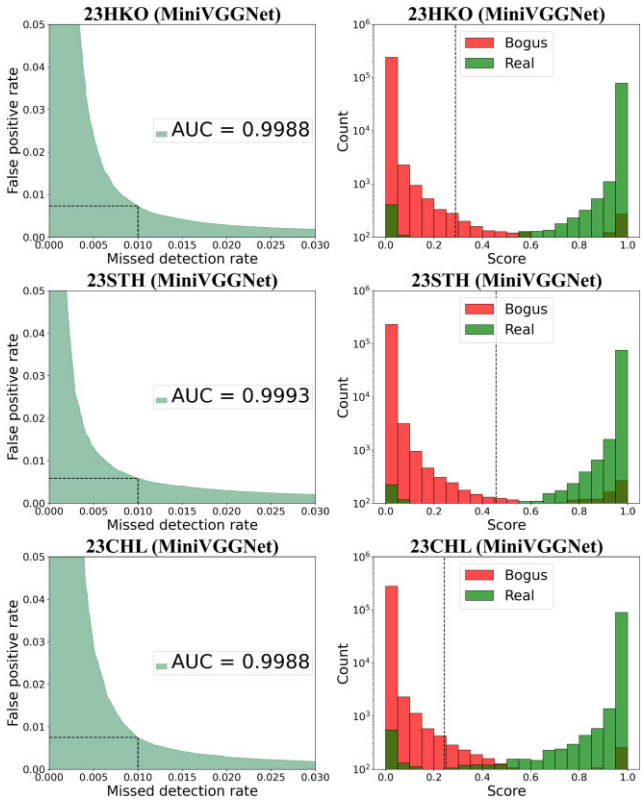

Figure 11. FPRs for the MiniVGGNet classifier on 20 × 20 2023 data, for a 1 per cent MDR.

object was processed rather than the date of first detection, which provides a more erratic short-term behaviour. Seasonal trends and weather patterns can also impact both the quality and amount of data recorded on a given night. For ease of examination, we take a 3-d rolling average of the metrics in our plots.

Prior to the implementation date, we find an average FPR of 2.52 per cent, with a high standard deviation indicating a non-robustness to varying observing conditions. Subsequently the performance improves in both quality and consistency in the following the implementation, with a mean FPR of 1.41 per cent and a standard deviation of 1.07 per cent. A larger training data size may have provided this improved generability to different nights.

Data reception from the MLO telescope began again on the 2024 November 22, after which we see increases in the FPR and number of counts. This is to be expected owing to the performance of the 20MLO model, which makes lower quality decisions compared with the 2023 classifiers. The performance remains above the levels seen prior to the implementation, and should decrease following the training of a 23MLO classifier.

The operational 'MDR' is difficult to measure. A missed detection would only be labelled as such if the object were to be first scored below the threshold by the classifier and subsequently rescued by human annotators. Missed detections therefore are only found following either investigation as a result of a detection made by other surveys or by eyeballing of the garbage data. The latter is not done as a routine process and subsequently the missed detection volume is lower than expected, with only three being recovered in the past 12 months. The true MDR is expected to be higher than this.

We note that we do not examine the classifier output against data quality. To understand the full performance of the classifiers through seasonal trends and weather patterns would require extensive long-term monitoring. From the past several months of data there is an improvement to classifier performance that indicates significant improvement over the previous models.

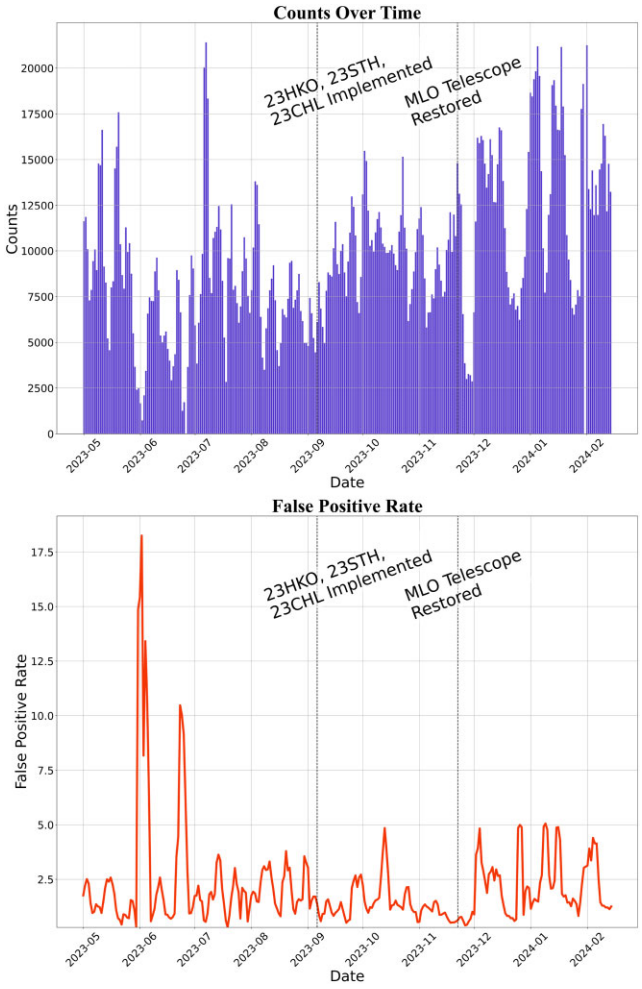

Figure 12. Performance of the ATLAS CNNs in operation, 3-d rolling averages, at a score threshold of 0.2.

Table 7. Performance metrics of the 2020 and 2023 classifiers at different thresholds.

Model	Threshold	Metric	HKO (per cent)	STH (per cent)	CHL (per cent)
2020	0.200	FPR	0.96	2.17	1.93
		MDR	1.70	1.11	1.64
	0.038	FPR	5.23	8.12	8.49
		MDR	0.72	0.55	0.83
2023	0.200	FPR	0.77	0.67	0.67
		MDR	1.03	0.86	1.14
	0.038	FPR	2.61	2.37	2.78
		MDR	0.53	0.33	0.50

It takes on average approximately 145 ms to produce images for each object. Each object consists of an average of 12 stamps (at least nine stamps to a maximum of 18), giving us a production time of 12 ms per cutout on a machine with 56 hyperthreaded cores. The evaluation time per object, which is carried out purely on the difference images, requires an additional 2.5 ms.

Table 8. Operational performance metrics of the 2020 and 2023 classifiers at different thresholds.

Threshold	Start date	End date	Mean FPR (per cent)	STD FPR (per cent)
0.2	01/05/2023	06/09/2023	2.52	4.63
	06/09/2023	22/11/2023	1.41	1.07
	22/11/2023	01/02/2023	1.89	2.07
0.038	01/05/2023	06/09/2023	8.24	7.16
	06/09/2023	22/11/2023	4.36	1.69
	22/11/2023	01/02/2023	5.38	3.22

8 CONCLUSIONS

In this paper, we have investigated machine learning algorithms applied to an image recognition problem on the currently operating ATLAS telescopes (HKO, CHL, and STH). For each detection from the difference images, we use a 20 × 20 pixel image stamp, centred on the detection, for RB classification. Using our previous CNN architecture, we train three independent classifiers for use in transient detection for our telescopes. Using a figure of merit outlined in Brink et al. (Brink et al. 2013; Wright et al. 2017) we selected the classifier decision boundaries with an MDR of 1 per cent, which resulted in FPRs of 0.72 per cent, 0.79 per cent, and 0.60 per cent for our 23HKO, 23CHL, and 23STH models, respectively. In tandem with our high ROC-AUC values for each classifier, we can state that these models outperform our previously used 20HKO and 20MLO classifiers. We can consider the use case of transient detection classifiers. At its base level, we aim to select a low number of real objects from a large population. While both our old and new classifiers have good performance as evidenced by their ROC-AUC values, the lower FPR and MDR for the latter indicate a better ability to ignore bogus objects without sacrifices to the MDR.

We tested our 2023 classifiers further on our list of 27 174 real astrophysical transients verified by human scanners and found that 112 were reclassed as bogus, of which 14 were notably misclassed. Five of these objects had also been misclassed by the previous classifiers. We also found that among 20 000 recent bogus objects, 215 are reclassed as real, with eight being given an RB factor >0.9, and two of these being genuine real objects.

We have attempted the use of smaller image stamps in our training data and found the resulting information loss negatively impacts the FPR. From comparisons with alternate CNNs, we can state that the 2023 ATLAS classifiers maintain a competitive performance while maintaining adaptability to new data trends.

We have noted bias in our data collection. Our data sets consist of four sub-populations. The real astrophysical transients are composed of known asteroids (slow moving and not trailed) and stationary transients promoted and selected by human scanning. Of bad objects, we have objects rejected by the RB threshold and objects rejected by human scanners. Of these, only the known real object population is independent of our 2020 model classifications. Objects promoted to good via scanning are pre-selected based on the RB threshold, meaning that real objects the model discards are less likely to be promoted, and as such any relevant features will not be weighed heavily by our new classifiers. The impact of this is doubled when we consider those ignored real objects will instead be in our bad population, where retrained classifiers may learn to treat these features negatively rather than ignore them. While we could incorporate checks for discoveries made by other surveys, differences to telescope hardware and image subtraction techniques mean that there is no guarantee an external discovery could have been made

independently from the corresponding ATLAS difference stamp. Such cases could be passed on for human scanning but labelling them could introduce some level of bias – the scanner already knows that the object is real regardless of data quality. An alternate solution would be to have human scanners parse each individual detection, both real and bogus, in our training and testing data. These are however large data sets (∼1000 000 objects) and it may not be expedient for a team of scanners to inspect each stamp and any associated data. For future retraining it may be worth considering human scanning on a sample of the training and testing data and inspect the impact on model fitting.

DATA AVAILABILITY

The model training and evaluation code utilized in this paper is available in a public github repository at https://github.com/genghisken/psat-ml. ATLAS science data are available on the forced photometry server. Training data sets are available upon reasonable request.

ACKNOWLEDGEMENTS

JW acknowledges a studentship funded by the Leverhulme Trust through the Leverhulme Interdisciplinary Network on Algorithmic Solutions (LINAS) at Queen's University Belfast. SJS and KS acknowledge STFC grants ST/X006506/1, ST/T000198/1, ST/S006109/1, and ST/X001253/1. SJS acknowledges a Royal Society Research Professorship. This work has made use of data from the Asteroid Terrestrial-impact Last Alert System (ATLAS) project. ATLAS is primarily funded to search for near-Earth asteroids through NASA grants NN12AR55G, 80NSSC18K0284, and 80NSSC18K1575; byproducts of the near-Earth object (NEO) search include images and catalogues from the survey area. The ATLAS science products have been made possible through the contributions of the University of Hawai'i Institute for Astronomy, the Queen's University Belfast, the Space Telescope Science Institute, and the South African Astronomical Observatory.

REFERENCES

Abbott B. P. et al., 2017, ApJ, 848, L12
Acero-Cuellar T., Bianco F., Dobler G., Sako M., Qu H., LSST Dark Energy Science Collaboration, 2023, AJ, 166, 115
Ackley K. et al., 2020, A&A, 643, A113
Alard C., 2000, A&AS,144, 363
Alard C., Lupton R. H., 1998, ApJ, 503, 325
Anderson J. P. et al., 2018, A&A, 620, A67
Arlot S., Celisse A., 2010, Stat. Surv., 4, 40
Becker A., 2015, Astrophysics Source Code Library, record ascl:1504.004
Bloom J. et al., 2012, PASP, 124, 1175
Bridle J. S., 1990, in Soulié F. F., Hérault J., eds, Neurocomputing. Springer, Berlin, p. 227
Brink H., Richards J. W., Poznanski D., Bloom J. S., Rice J., Negahban S., Wainwright M., 2013, MNRAS, 435, 1047
Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J.-C., 2017, ApJ, 836, 97
Carrasco-Davis R. et al., 2021, AJ, 162, 231
Chollet F., 2017, Keras: Deep Learning for Humans. GitHub. Available at: https://github.com/fchollet/keras
Duev D. A.et al., 2019, MNRAS, 489, 3582
Dumoulin V., Visin F., 2016, preprint (arXiv:1603.07285)
Gieseke F. et al., 2017, MNRAS, 472, 3101
Hastie T., Friedman J., Tisbshirani R., 2017, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Berlin
Heinze A. N. et al., 2018, AJ, 156, 241
Heinze A. N. et al., 2021, Planet. Sci. J., 2, 12
Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., 2017, preprint (arXiv:1704.04861)
Inserra C., 2019, Nat. Astron., 3, 697
Kingma D., Ba J., 2015, Proc. 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA
Krizhevsky A., 2009, Learning Multiple Layers of Features from Tiny Images, https://api.semanticscholar.org/CorpusID:18268744 (Accessed January 2024)
LeCun Y., Cortes C., 2005, The MNIST Database of Handwritten Digits, https://api.semanticscholar.org/CorpusID:60282629 (Accessed January 2024)
Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, Proc. IEEE, 86, 2278
Lecun Y. et al., 1995, in Oh J. H., Kwon C., Cho S., Neural Networks: The Statistical Mechanics Perspective. World Scientific, p. 261
Leoni M., Ishida E. E. O., Peloton J., Möller A., 2022, A&A, 663, A13
Lin M., Chen Q., Yan S., 2014, preprint (arXiv:1312.4400)
Mahabal A. et al., 2019, PASP, 131, 038002
Miranda N. et al., 2022, A&A, 665, A99
Nicholl M. et al., 2023, ApJ, 954, L28
Prentice S. J. et al., 2018, ApJ, 865, L3
Reddy V. et al., 2022, Planet. Sci. J., 3, 123
Rehemtulla N. et al., 2024, preprint (arXiv:2401.15167)
Reyes E., Estövez P. A., Reyes I., Cabrera-Vives G., Huijse P., Carrasco R., Forster F., 2018, Proc. 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, Queensland, Australia 1
Ruder S., 2016, preprint (arXiv:1609.04747)
Russeil E. et al., 2024, A&A, 683, A251
Schulze S. et al., 2024, A&A, 683, A223
Sharma N., Jain V., Mishra A., 2018, Procedia Comput. Sci., 132, 377
Sheng X. et al., 2024, MNRAS, 531, 2474
Sifre L., Mallat S., 2014, preprint (arXiv:1403.1687)
Simonyan K., Zisserman A., 2015, preprint (arXiv:1409.1556)
Smartt S. J. et al., 2017, Nature, 551, 75
Smartt S. J., et al., 2023, GCN Circ., 33278, 1
Smith K. W. et al., 2020, PASP, 132, 085002
Srivastav S. et al., 2022, MNRAS, 511, 2708
Srivastav S. et al., 2023a, ApJ, 943, L20
Srivastav S. et al., 2023b, ApJ, 956, L34
Stalder B. et al., 2017, ApJ, 850, 149
Taubenberger S., 2017, Handbook of Supernovae. Springer International Publishing, Berlin, p. 317
Tomaney A. B., Crotts A. P. S., 1996, AJ, 112, 2872
Tonry J. L., 2010, PASP, 123, 58
Tonry J. L. et al., 2018a, ApJ, 867, 105
Tonry J. L. et al., 2018b, PASP, 130, 064505
Villar V. A. et al., 2019, ApJ, 884, 83
Wright D. E., 2015, Department of Physics and Astronomy, Queens University Belfast
Wright D. E. et al., 2015, MNRAS, 449, 451
Wright D. E. et al., 2017, MNRAS, 472, 1315
Yamaguchi K., Sakamoto K., Akabane T., Fujimoto Y., 1990, Proc. First International Conference on Spoken Language Processing (ICSLP 1990). Kobe, Japan, p. 1077

SUPPORTING INFORMATION

Supplementary data are available at *RASTAI* online.

Retraining and Improvements to the ATLAS Real-Bogus Classifier RASTI (v1.04) Source.zip

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.

APPENDIX A: ADDITIONAL FIGURES

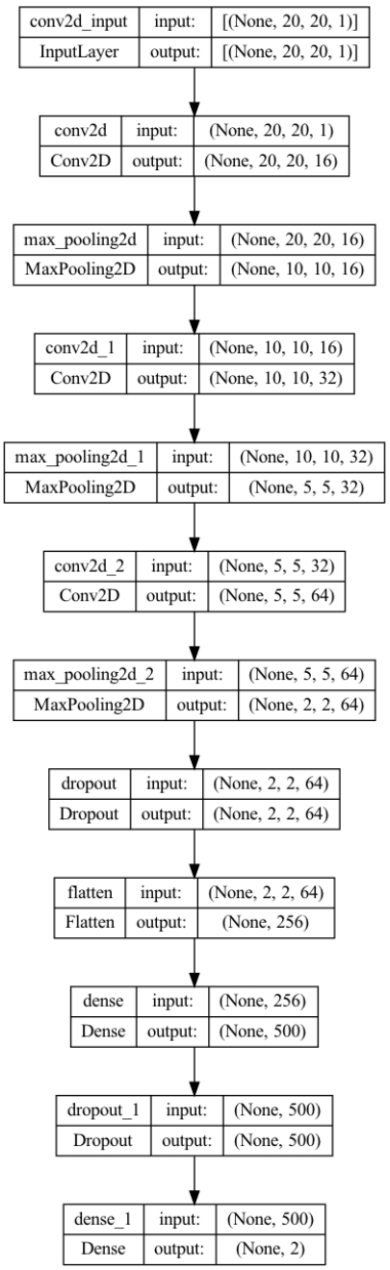

Figure A1. Utilized CNN architecture for the ATLAS RB classifiers.

Table A1. Performance metrics of classifiers on training data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent. Likewise MDR refers to the MDR value when FPR = 1.0 per cent.

Telescope	Classifier	Data	FPR	MDR
HKO	2020	2020	0.266 570	0.022 122
	2020	2023	0.088 106	0.035 112
	2023	2020	0.135 781	0.059 280
	2023	2023	0.004 400	0.006 118
STH	2020	2023	0.042 186	0.033 847
	2023	2023	0.004 271	0.005 097
CHL	2020	2023	0.093 318	0.048 290
	2023	2023	0.006 087	0.007 234

Table A2. Performance metrics of classifiers on validation data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent. Likewise MDR refers to the MDR value when FPR = 1.0 per cent.

Telescope	Classifier	Data	FPR	MDR
HKO	2020	2020	0.036 578	0.032 807
	2020	2023	0.090 623	0.036 630
	2023	2020	0.131 285	0.060 703
	2023	2023	0.007 324	0.008 656
STH	2020	2023	0.039 039	0.032 556
	2023	2023	0.005 597	0.006 094
CHL	2020	2023	0.090 077	0.048 056
	2023	2023	0.008 709	0.008 961

Table A3. Performance metrics of classifiers on testing data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent. Likewise MDR refers to the MDR value when FPR = 1.0 per cent.

Telescope	Classifier	Data	FPR	MDR
HKO	2020	2020	0.041 387	0.033 573
	2020	2023	0.091 014	0.035 842
	2023	2020	0.128 480	0.062 107
	2023	2023	0.007 164	0.008 364
STH	2020	2023	0.038 700	0.034 425
	2023	2023	0.005 967	0.007 916
CHL	2020	2023	0.094 411	0.047 768
	2023	2023	0.007 916	0.008 800

Table A4. Performance metrics of MiniVGGNet classifiers on training data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent. Likewise MDR refers to the MDR value when FPR = 1.0 per cent.

Data set	FPR	MDR
23HKO	0.004 601 235	0.006 198 403
23STH	0.003 517 338	0.004 558 503
23CHL	0.004 601 235	0.006 198 403

Table A5. Performance metrics of MiniVGGNet classifiers on validation data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent. Likewise MDR refers to the MDR value when FPR = 1.0 per cent.

Data set	FPR	MDR
23HKO	0.007 151 589	0.008 364 748
23STH	0.005 147 055	0.005 876 755
23CHL	0.007 151 589	0.008 364 748

Table A6. Performance metrics of MiniVGGNet classifiers on testing data sets. FPR refers to the FPR value when we fix the MDR at 1.0 per cent. Likewise MDR refers to the MDR value when FPR = 1.0 per cent.

Data set	FPR	MDR
23HKO	0.007 284 848	0.008 169 697
23STH	0.005 887 500	0.006 350 000
23CHL	0.007 284 848	0.008 169 697

This paper has been typeset from a TEX/LATEX file prepared by the author.