
Article:
Mong, Y.-L., Ackley, K., Killestein, T.L. orcid.org/0000-0002-0440-9597 et al. (46 more authors) (2023) Self-supervised clustering on image-subtracted data with deep-embedded self-organizing map. Monthly Notices of the Royal Astronomical Society, 518 (1). pp. 752-762. ISSN 0035-8711

https://doi.org/10.1093/mnras/stac3103

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **518,** 752–762 (2023) https://doi.org/10.1093/mnras/stac3103
Advance Access publication 2022 November 2

Self-supervised clustering on image-subtracted data with deep-embedded self-organizing map

Y.-L. Mong,[1,2★] K. Ackley,[1,2,3★] T. L. Killestein,[3] D. K. Galloway,[1,2,4] C. Vassallo,[5] M. Dyer,[6] R. Cutter,[3] M. J. I. Brown,[1] J. Lyman,[3] K. Ulaczyk,[3] D. Steeghs,[3] V. Dhillon,[6,7] P. O'Brien,[8] G. Ramsay,[9] K. Noysena,[10] R. Kotak,[5] R. Breton,[11] L. Nuttall,[12] E. Pallé,[7] D. Pollacco,[3] E. Thrane,[1,2] S. Awiphan,[10] U. Burhanudin,[6] P. Chote,[3] A. Chrimes,[3] E. Daw,[6] C. Duffy,[9] R. Eyles-Ferris,[8] B. P. Gompertz,[3] T. Heikkilä,[5] P. Irawati,[10] M. Kennedy,[11] A. Levan,[3] S. Littlefair,[6] L. Makrygianni,[6] T. Marsh,[3] D. Mata Sánchez,[7,13] S. Mattila,[5] J. R. Maund,[6] J. McCormac,[3] D. Mkrtichian,[10] J. Mullaney,[6] E. Rol,[1,2] U. Sawangwit,[10] E. Stanway,[3] R. Starling,[8] P. Strøm,[3] S. Tooke[8] and K. Wiersema[3]

[1]*School of Physics & Astronomy, Monash University, Clayton VIC 3800, Australia*
[2]*OzGrav: The ARC Centre of Excellence for Gravitational Wave Discovery, Clayton VIC 3800, Australia*
[3]*Department of Physics, University of Warwick, Coventry, West Midlands, CV4 7AL, UK*
[4]*Institute for Globally Distributed Open Research and Education (IGDORE)*
[5]*Department of Physics and Astronomy, University of Turku, FI-20014 Turun yliopisto, Finland*
[6]*Department of Physics and Astronomy, University of Sheffield, Sheffield, S3 7RH, UK*
[7]*Instituto de Astrofísica de Canarias, E-38205 La Laguna, Tenerife, Spain*
[8]*School of Physics and Astronomy, University of Leicester, University Road, Leicester, LE1 7RH, UK*
[9]*Armagh Observatory & Planetarium, College Hill, Armagh, BT61 9DB, County Armagh, Northern Ireland, UK*
[10]*National Astronomical Research Institute of Thailand, 260 Moo 4, T. Donkaew, A. Maerim, Chiangmai, 50180, Thailand*
[11]*Department of Physics and Astronomy, University of Manchester, M13 9PL, UK*
[12]*Institute of Cosmology and Gravitation, University of Portsmouth, Dennis Sciama Building, Burnaby Road, Portsmouth, PO1 3FX, UK*
[13]*Departamento de Astrofísica, Universidad de La Laguna, E-38206 La Laguna, Tenerife, Spain*

Accepted 2022 October 17. Received 2022 September 4; in original form 2022 February 12

ABSTRACT

Developing an effective automatic classifier to separate genuine sources from artifacts is essential for transient follow-ups in wide-field optical surveys. The identification of transient detections from the subtraction artifacts after the image differencing process is a key step in such classifiers, known as real-bogus classification problem. We apply a self-supervised machine learning model, the deep-embedded self-organizing map (DESOM) to this 'real-bogus' classification problem. DESOM combines an autoencoder and a self-organizing map to perform clustering in order to distinguish between real and bogus detections, based on their dimensionality-reduced representations. We use 32 × 32 normalized detection thumbnails as the input of DESOM. We demonstrate different model training approaches, and find that our best DESOM classifier shows a missed detection rate of 6.6 per cent with a false-positive rate of 1.5 per cent. DESOM offers a more nuanced way to fine-tune the decision boundary identifying likely real detections when used in combination with other types of classifiers, e.g. built on neural networks or decision trees. We also discuss other potential usages of DESOM and its limitations.

Key words: methods: observational.

1 INTRODUCTION

Time-domain astronomy has risen in popularity among the astronomical research fields during the past decade. It is particularly important for the study of gamma-ray bursts (GRBs, Piran 2004; Zhang et al. 2006; Li et al. 2012; Berger, Fong & Chornock 2013; Jin et al. 2013; Tanvir et al. 2013; Berger 2014; Cenko et al. 2015; Kumar & Zhang 2015; Kasliwal et al. 2017; Lamb et al. 2019; Coughlin et al. 2020; Ho et al. 2020; Andreoni et al. 2021; Mong et al. 2021) and gravitational-wave (GW) events (Abbott et al. 2016, 2017; Blanchard et al. 2017; Chornock et al. 2017; Coulter et al. 2017; Cowperthwaite et al. 2017; Goldstein et al. 2017; Hallinan et al. 2017; Margutti et al. 2017; Savchenko et al. 2017; Gompertz et al. 2020). These events require prompt (time-scales of hours) follow-up observations in order to identify their nature, before they become too faint to be detected (see e.g. Rau et al. 2009). Information about the event's origin can often be relatively poor, with uncertainty regions of hundreds of square degrees typical. To this aim, facilities with a large field of view (FoV) enable the follow-up observations of these transient events.

★ E-mail: yik.mong@monash.edu (Y-LM); kendall.ackley@warwick.ac.uk (KA)

The Gravitational-wave Optical Transient Observer (GOTO)[1] is a robotic optical telescope designed to search for the counterparts of GW events (Dyer et al. 2020; Steeghs et al. 2022). GOTO presently consists of two telescope arrays, each equipped with 8×40 cm unit telescopes and a total FoV of $\sim$80 square degrees per pointing. Other than following up GRBs and GW events, GOTO also performs regular all-sky survey observations in order to explore the transient sky with serendipitous searching. It can reach depths of $\approx$20 mag in the adopted Baader *L*-band filter (400–700 nm), with a set of 3×90 s exposures. The image size of each camera is 8176×6132 pixels with a pixel scale of $\approx$1.2 arcsec.

Difference imaging is commonly used to identify transient objects on an image (Alard & Lupton 1998; Becker 2015). With a reference image, taken during a historical visit of the same field as the input image (also called the science image), the two images can be aligned by performing an affine transformation (e.g. with a custom Python package `spalipy`). The aligned reference frame is then subtracted from the science frame (e.g. using `HOTPANTS`) in order to generate a difference image (Becker 2015). An ideal subtraction helps to remove the vast majority of the objects that do not vary in intensity. Transients which appear only on the science frame should appear on the difference image after the image subtraction process.

'Real-bogus' classification is the process of separating real objects from 'bogus' detections, including instrumental artifacts, subtraction residuals, bad pixels, etc., on the difference image (Bloom et al. 2012). Due to imperfections in the difference imaging method, $\sim 10^4$ subtraction artifacts can be identified as detections by `SExtractor` per GOTO image (Bertin & Arnouts 1996). A large number of subtraction artifacts make it impossible to manually separate real transients from bogus detections. As a result, an automatic real-bogus classifier is required. Supervised machine learning models are commonly used to construct the real-bogus classifier. Among all of the supervised learning algorithms, the convolutional neural network (CNN) model shows the best performance on solving computer vision problem (Cabrera-Vives et al. 2016, 2017; Gieseke et al. 2017). The current real-bogus classifier of GOTO is built based on the VGG16 CNN model (hereafter GOTO-VGG, Simonyan & Zisserman 2014; Duev et al. 2019; Killestein et al. 2021).

There are two major disadvantages of using supervised machine learning to solve real-bogus classification problems. First, since the training process of the model is supervised, all training instances must be labelled as real or bogus. This labelling process requires a high cost of expert human labour. This challenge has been previously addressed by using detections on the science frame to train the classifier (Mong et al. 2020) or building a training set with synthetic transients (Smith et al. 2020; Killestein et al. 2021). Second, during the actual prediction process, supervised learning models act like 'black box' models. The algorithm is typically too complicated for a human to visually understand how the prediction is made. Therefore, improving the model is challenging.

Using unsupervised learning models to build our classifier, avoids the shortcomings of supervised learning models. Since the clustering model groups similar data together, it is easy to understand that two inputs with the same prediction means that they are close to each other in the parameter space, and hence likely arising from a similar origin.

In this paper, we apply unsupervised machine learning to solve the real-bogus classification problem. In Section 2, we introduce the learning algorithms used to construct our classifier. In Section 3, we discuss how to extract and pre-process our data before performing further analyses. In Section 4, we discuss how we train our model and report the result of our best model. We also compare the model performance with the GOTO-VGG model. In Section 5, we discuss the advantages and the shortcomings of the model. Finally, we conclude our results in Section 6.

[1] http://goto-observatory.org

2 LEARNING ALGORITHMS

We employ the deep-embedded self-organizing map (DESOM, Forest et al. 2019; Teimoorinia et al. 2021) as the unsupervised learning model to build our real-bogus classifier. DESOM consists of two parts, the autoencoder (AE) and the self-organizing map (SOM). In this section, we will be introducing the basic concept behind these algorithms.

2.1 Autoencoder

The AE model is a variant of the neural network model, with the main objective of reconstructing the input data via dimensionality reduction (Baldi 2012; Wang et al. 2014; Bank, Koenigstein & Giryes 2020). The architecture of the AE is usually symmetric, i.e. the output has the same dimension as the input. Due to the objective of AE, the input X is also used as the target in the training process. Therefore, AE is considered to be a 'self-supervised' learning algorithm.

The AE architecture consists of three parts, the encoder, the 'bottleneck', and the decoder. The bottleneck represents the middle layer of the AE. AE can generally be divided into two types, undercomplete and overcomplete. For the undercomplete AE, the number of neurons in the bottleneck layer is smaller than that of the input layer. On the other hand, an overcomplete AE has a bottleneck with more neurons than the input layer. In this work, we use undercomplete AE to construct our model in order to capture the most important features from the raw input data.

The first half of the AE is defined as the encoder. It maps the input X to the output parameter space, also called the 'latent space'. Since the bottleneck has a smaller size than the input layer, the output of the bottleneck is considered to be a compressed representation of the raw input. Therefore, the encoder can be used to perform dimensionality reduction. The second half of the AE is the decoder. It takes the compressed representation $\phi(X)$ as the input and attempts to reconstruct the original input X. The mathematical operation of an AE can be written as

$$\hat{X} = \psi(\phi(X)), \tag{1}$$

where ϕ and ψ are the operators of the encoder and the decoder, respectively. The reconstructed output of the AE is denoted by $\hat{X}$.

The encoder part of the AE is usually identical to an artificial neural network (ANN, Daniel 2013) or a CNN (O'Shea & Nash 2015) model. The choice of using the convolutional (`Conv2D`) layers or the fully connected (`dense`) layers depends on the type of the problem. The `Conv2D` layer identifies local patterns under translation and rotation invariance. For computer vision problems, `Conv2D` usually performs better in general. In this work, we use a CNN model to construct our AE model.

2.2 Self-organizing map

A SOM is a clustering algorithm consisting of only two layers, the input layer and the output layer (Kohonen 1990, 2001). Each layer consists of several nodes, and each input node is connected to all output nodes with a corresponding weight. Since

those two layers are fully connected, SOM is a variant of an ANN model. The number of output nodes, which represents the number of desired clusters, is the most important hyperparameter of SOM.

Each cluster in the input parameter space is characterized by a weight vector, which is called the prototype vector (PV), of an output node. The SOM nodes usually form a two-dimensional (2D) grid, also called a SOM map. The input vector $\mathbf{x}$ is connected with an output node with the weights $\mathbf{w}_{ij}$, where i, j indicates the position of the node in the output layer. Therefore, the PV of an $m \times m$ SOM map can be expressed as

$$\mathbf{W} = \begin{pmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} & \cdots & \mathbf{w}_{1m} \\ \mathbf{w}_{21} & \mathbf{w}_{22} & \cdots & \mathbf{w}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{m1} & \mathbf{w}_{m2} & \cdots & \mathbf{w}_{mm} \end{pmatrix}. \tag{2}$$

The prediction output of the SOM model is the cluster with the minimum Euclidean distance between $\mathbf{w}_{ij}$ and $\mathbf{x}$. We call the selected $\mathbf{w}_{ij}$ the 'winner PV' of $\mathbf{x}$.

In the SOM training process, $\mathbf{W}$ is updated iteratively with three steps: competition, cooperation, and adaptation. In the competition step, the input $\mathbf{x}$ searches for the winner PV $\mathbf{w_k}$ with the minimum Euclidean distance $|\mathbf{w_k} - \mathbf{x}|$ among $\mathbf{W}$. In the cooperation step, the distances between $\mathbf{w_k}$ and other PVs $\mathbf{w}_{ij}$ are calculated. The winner PV $\mathbf{w_k}$ will then be updated by reducing the Euclidean distance between the $\mathbf{w_k}$ and the $\mathbf{x}$. However, $\mathbf{w_k}$ is not the only PV being updated. In fact, all $\mathbf{w}_{ij}$ will be updated in the final adaptation step depending on their distances from $\mathbf{w_k}$.

The update of any $\mathbf{w_i}$ is directional pointing towards $\mathbf{x}$ in the latent space, and the step size is controlled by three factors: the Euclidean distance between $\mathbf{w_k}$ and $\mathbf{x}$, the Euclidean distance between $\mathbf{w_k}$ and $\mathbf{w_i}$, and the learning rate η. The step size of the update also decays with the number of the training iterations t. Therefore, the algorithm of the update is written as

$$\mathbf{w_i}' = \mathbf{w_i} + \eta(t) h_{\mathbf{ik}}(t) |\mathbf{x} - \mathbf{w_k}|, \tag{3}$$

where

$$h_{\mathbf{ik}}(t) = \exp\left(-\frac{|\mathbf{w_i} - \mathbf{w_k}|^2}{T(t)^2}\right) \tag{4}$$

is the kernel function. We use the Gaussian neighbourhood function to be the kernel here. The temperature parameter,

$$T(t) = T_0 \exp\left(-\frac{t}{\tau_T}\right), \tag{5}$$

determines the kernel size at the training epoch t. The maximum temperature and the decay constant are denoted by T_0 and τ_T. The same form is used to describe the t-dependent learning rate,

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_\eta}\right). \tag{6}$$

2.3 Deep-embedded self-organizing map

The DESOM model is constructed by combining an AE and SOM. The AE achieves the dimensionality reduction, while the SOM performs the actual clustering process on the dimensionality-reduced input. Therefore, the SOM layer is attached to the bottleneck of the AE. With this model architecture, DESOM can be divided into two types depending on how the model is trained, the 'combine-trained' DESOM, the 'separate-trained' DESOM.

2.3.1 *Combine-trained DESOM*

The combined training of DESOM was first demonstrated by Forest et al. (2019). In the training process, the parameters in all layers are trainable. In the other words, both AE and SOM are trained at the same time. A combine-trained DESOM was also used to build an image-quality-based recommendation system (Teimoorinia et al. 2021). Fig. 1 illustrates the architecture of DESOM. As we can see that DESOM generates two separate outputs from the decoder and the SOM layer with their corresponding losses, the least-squares reconstruction loss $L_{\rm dec}$ and the SOM loss $L_{\rm som}$ indicating the Euclidean distance between the input $\mathbf{x}$ and the winning PV in the latent space. The total loss of each run can be defined as the weighted sum of the two losses:

$$L_{\rm tot} = L_{\rm dec}(\boldsymbol{\theta}_{\rm e}, \boldsymbol{\theta}_{\rm d}) + \gamma L_{\rm som}(\boldsymbol{\theta}_{\rm e}, \boldsymbol{\theta}_{\rm som}), \tag{7}$$

where γ is the weight of the SOM loss. The trainable model parameters of the encoder, the decoder, and the SOM are denoted by $\boldsymbol{\theta}_{\rm e}, \boldsymbol{\theta}_{\rm d}$, and $\boldsymbol{\theta}_{\rm som}$. All parameters connected from the output layer to the input layer are updated in each training iteration through a numerical algorithm called back-propagation (Munro 2010). However, since we have two output layers, there are two different paths in the back-propagation to update the parameters (see the *red* lines in Fig. 1). Therefore, the back-propagations of both decoder and SOM also contribute to the update of the encoder, and this is the reason why both losses also depend on $\boldsymbol{\theta}_{\rm e}$.

There are two major disadvantages of using this training approach. First, DESOM has many adjustable hyperparameters such as the number of layers, the number of neurons, and the size of the SOM map, among others, which makes the model evaluation with respect to different hyperparameter set-ups very complicated. Blindly searching for the best hyperparameter set is very time consuming. Second, this approach provides relatively ineffective SOM training. At the beginning of the SOM training phase, the kernel size τ_T is large, such that the update of the SOM map is more global. As a result, the early training phase of SOM is more effective to roughly map the SOM map on to the latent space of AE. However, the AE is undertrained during the early training phase. If we train both SOM and AE together, the SOM layer will learn the unuseful information from the undertrained AE latent space effectively due to a large $T(t \sim 0)$. On the other hand, once the AE has been well-trained, the SOM kernel $h_{\mathbf{ik}}(t \gg \tau_T)$ has converged to a small size. At that time, the training process would only fine-tune the SOM layer, which means that the final SOM is mainly trained on an undertrained AE. To solve this problem, we are motivated to explore the following alternative training approach.

2.3.2 *Separate-trained DESOM*

Here, we instead break down the DESOM training into two separate processes. First, we train the AE individually. Then, we freeze both $\boldsymbol{\theta}_{\rm e}$ and $\boldsymbol{\theta}_{\rm d}$ (by manually setting them to be 'untrainable' parameters). Finally, we insert the SOM layer to the bottleneck of the AE and train for it with the frozen, trained AE.

Training the AE and SOM individually can effectively speed up the hyperparameter searching process. Considering that AE and SOM have M and N hyperparameter configurations, respectively, the grid searching finds the best combined configuration among $M \times N$ configurations. By separating the training processes of AE and SOM, we first search for the best AE among those M configurations. Then, we implement the best AE to search for the best SOM configuration. Therefore, the number of trials reduces

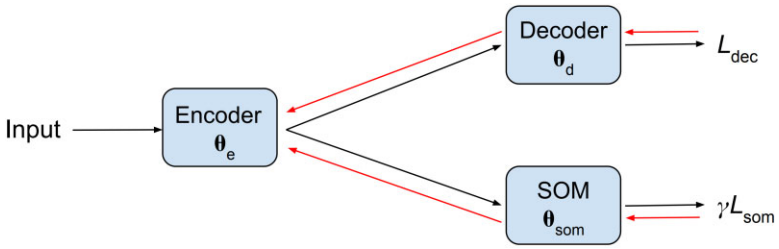

Figure 1. The propagation flows of DESOM training (Forest et al. 2019). The *black* lines indicate the direction of the forward propagation. And the *red* lines indicate the direction of the back-propagation.

down to $M + N$, which significantly speeds up the grid searching process.

Since the SOM training process commences after obtaining the best AE configuration, the SOM layer is directly trained on the well-established latent space of the AE bottleneck. Unlike the combine-trained DESOM, which the SOM is trained on an unlearnt bottleneck of the AE during the early training phase, the training process of the SOM layer in the separate-trained DESOM is more effective. Using this approach, all the shortcomings of the combined training approach can be addressed.

2.4 GOTO-VGG classifier

To provide a direct comparison to the DESOM architecture presented in this work, we used the pre-trained CNN classifier currently being used in the live GOTO pipeline, details of which are presented in full in Killestein et al. (2021). This model is a 330 000 parameter, eight layer deep CNN that was trained on around 400 000 labelled examples. The model inherits the broad structure of the GOTO-VGG network of Simonyan & Zisserman (2014) by utilizing 'conv-conv-pool' blocks, but is significantly downscaled owing both to the smaller scale and overall lower complexity of astronomical images, compared to the data sets used in the computer science literature.

3 DATA EXTRACTION AND PROCESSING

We randomly gathered 719 stacked images taken by GOTO between 2021 April 16 and 2021 June 12 to perform analyses in this work. Each of these images includes science, reference, and subtracted frames. All images are processed through the GOTO standard image processing pipeline (Steeghs et al. 2022).

We use two different approaches, the difference-coordinate (DC) approach and the science-coordinate (SC) approach (see Sections 3.1 and 3.2 for more details), to extract the coordinates of the detections from the difference images. Once we obtain the detection coordinates, we use a 32 × 32 pixel cutout centred on the coordinates to be the input thumbnail (see Fig. 2 for some examples extracted with the DC approach). Since DESOM is an unsupervised learning model, labelling process is not required to build our training set.

3.1 Difference-coordinate approach

The DC approach is the usual approach of extracting the detection coordinates of candidate sources from the difference image. The coordinates are extracted by running the source extraction software `SExtractor` on the difference images. We randomly extracted 1000 000 detection cutouts from 719 difference images to build our data set (DC data set). We further split the data set into training and test sets, which contain 800 000 and 200 000 detections, respectively.

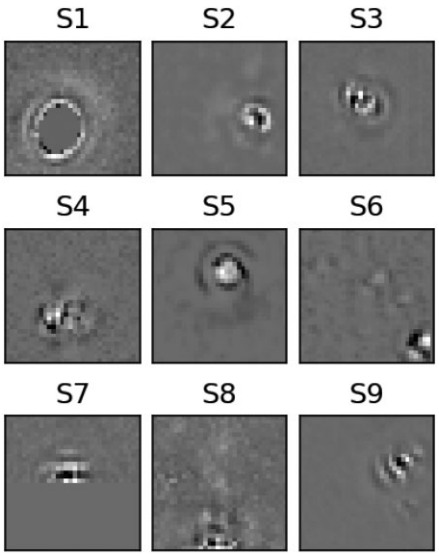

Figure 2. Examples of 32 × 32 pixel cutouts from GOTO prototype images extracted with the DC approach. S1 is an example of masked subtraction of a bright source. S6, S8, and S9 are likely due to the statistical fluctuation of the background. S7 is a detection lying on the edge of the difference image. These examples show that the DC approach usually does not centre the subtraction residual within the thumbnail.

There are two obvious problems of using this approach to create inputs. First, as we can see in Fig. 2, the central points of most of the cutouts are offset from the original positions of the sources on the science images. This issue is caused by the fact that `SExtractor` tends to identify the subtraction residuals surrounding the actual position of the source, as candidate detections. The root of this issue resides in the following `SExtractor` performance method. During the subtraction process, pixel discretization and fractional shifts may result in point spread function (PSF) kernel mismatches between the science and reference frames. As a result, the science detections are segmented into multiple peaks extracted by `SExtractor` on the difference image (see Fig. 3). This effect substantially increases the number of bogus detections on the difference image. Thus, even a classifier with a low false-positive rate (FPR) could result in many false positives. The disadvantages of this approach motivate us to develop another approach, the SC approach, to perform source extraction.

3.2 Science-coordinate approach

Here, we run `SExtractor` on the science frame instead of the difference frame, to extract the detection coordinates. Those coordinates will then be used to generate cutouts from the difference image. The

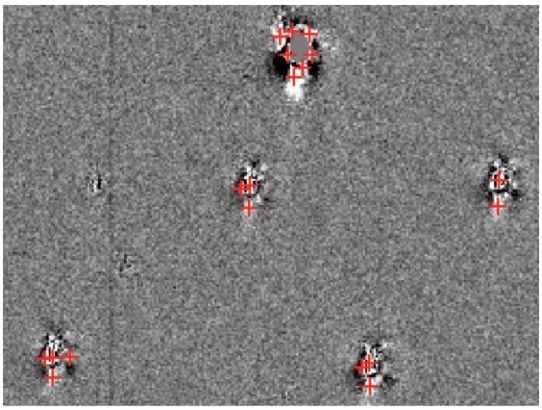

Figure 3. Example of the segmentation caused by difference imaging. The extracted detections recovered by `SExtractor` are indicated with *red* crosses. The segmentation of the image subtraction can substantially increase the number of the bogus detections on a difference image.

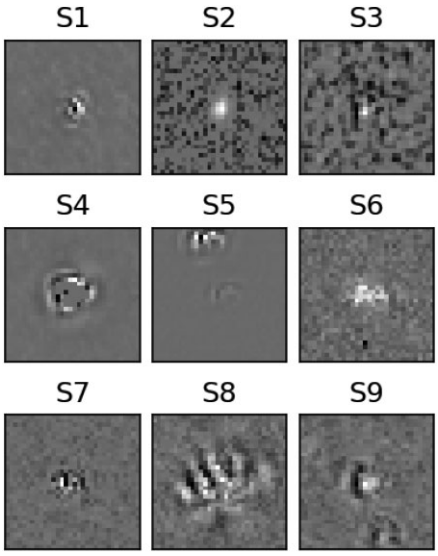

Figure 4. Examples of 32 × 32 pixel cutouts from GOTO prototype images extracted with the SC approach. Unlike the ones extracted with the DC approach (see Fig. 2), the subtraction residuals are centred at the middle of the thumbnails.

thumbnail generated with the SC approach is centred at the original position of the source on the science image, as the difference image is astrometrically aligned to the science frame before the image subtraction (see Fig. 4). In addition, we can effectively eliminate the majority of the segmented detections which might arise from the subtraction of a single source. Therefore, the SC approach solves both problems of the DC approach.

Nevertheless, the SC approach has its own issues. Using the coordinates of all science detections to extract the cutouts is extremely inefficient, since the subtraction process can otherwise effectively reject persistent objects. To solve this problem, we cross-match all detections on the science image with the ones on the difference image, and select only those science detections with a cross-matched result within a radius defined in equation (8). If the same detection appears on both science image and difference image, the offset in the cross-match result is defined as the 'subtraction offset'. However, it is very challenging to conclude that every detection on the science

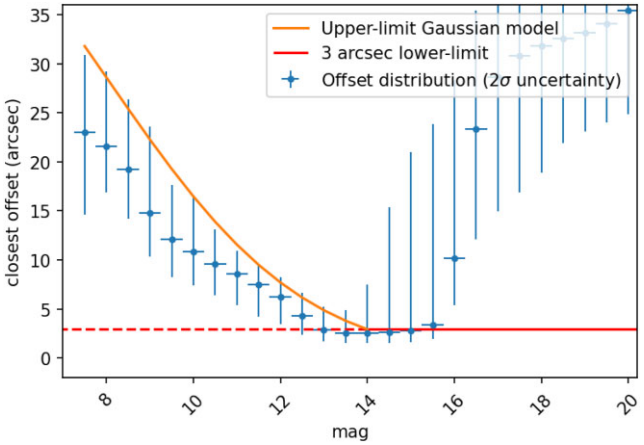

Figure 5. The offset between the detection on the science image and its closest detection on the difference image against its magnitude. The *y*-axis error-bar indicates the 2σ confidence level of the offset distribution. The *orange* curve represents the best Gaussian function fitting the 2σ upper limits of the data with $m < 13.5$. The *red* line indicates the lower limit of $f(m)$ in equation (8) at 3 arcsec. For the science detection with $m > 13.5$ and an offset smaller than 3 arcsec, the closest detection on the difference image is considered as the same source.

image is the same source of another detection on the difference image even if the subtraction offset is small.

Although we can only conclude that a smaller offset implies a higher chance of association, we have to set a critical offset to claim the association of the detections on the science image and the difference image. We plot the offsets of the detections on the science images from the closest detections on difference images against the magnitudes of the detections on the science image (Fig. 5). Depending on the brightness of the source, the closest offset could be a good representative of the subtraction offset. For the bright sources with $m \lesssim 10$, the subtraction offset can go up to $\sim$30 arcsec (2σ confidence level), due to the effect of masking saturated pixels. For the fainter detections with $m \gtrsim 15$, the offset gets larger with decreasing brightness. This can be explained by the fact that the subtraction on the faint stable source is usually cleaner. If the subtraction residual left on the difference image is negligible, it cannot be detected by `SExtractor`. Therefore, in the cross-matching process, the science detection actually matches with another completely different source. Hence, the cross-matched offset becomes larger. We conclude that the closest offset beyond $m > 13.5$ in Fig. 5 does not reflect the subtraction offset of the same source.

Since the subtraction offset is magnitude-dependent, we obtain the offset threshold as a function of magnitude by fitting a phenomenological piece-wise Gaussian function,

$$f(m) = \max\left[3,\, A\exp\left(-\frac{(m-m_0)^2}{2\sigma^2}\right)\right], \tag{8}$$

to the data in Fig. 5. We restrict our fitting at 2σ upper limit of the subtraction offset and $m < 13.5$. The best-fitting parameters are $A = 46.3$, $m_0 = 3.,7$ and $\sigma = 4.4$. We also set the lower limit of the offset threshold at 3 arcsec. Any detection with magnitude m on the science image which has a closest offset smaller than $f(m)$ is considered as having a cross-matched result, and it is included in the data set.

With this approach, the number of detections that need to be analysed is greatly reduced. In Fig. 6, we can see that a science frame usually has $\sim 10^4$ detections. After the difference imaging, only $\sim 10^{3-4}$ detections are left on a difference image. With the

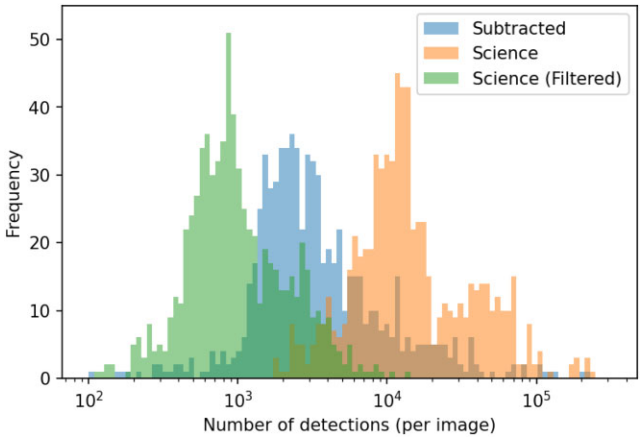

Figure 6. Distributions of numbers of science detections (*orange*), subtracted detections (*blue*) and extracted detections with the SC approach (*green*) per image. With the SC approach, the number of detections needed to be analysed drops to $\sim 10^{2-3}$ per image.

SC approach, $\sim 10^{2-3}$ detections per image have to be analysed, which is an order of magnitude smaller than the original number of science detections. However, this filtering step has to be done carefully in order to prevent overfiltering. To evaluate how many detections are overfiltered, we estimate the fraction of the 'real-labelled' detections which are not filtered in the SC approach. This estimation is done by manually inspecting 200 random detections from those overfiltered detections. We find that the SC approach can only recover $\approx$80 per cent of those 'real-labelled' detections. We also find that $\approx$50 per cent of those overfiltered detections are only statistical fluctuations instead of genuine real detections. Therefore, the overfiltered rate of the SC approach is $\approx$10 per cent. For those overfiltered detections, most of them are filtered by mistake due to a slightly larger subtraction offset. This overfiltered rate is roughly consistent with the fact that we obtain our offset threshold $f(m)$ by fitting to the 2σ confidence level of the subtraction offset and set the lower limit at 3 arcsec.

We further exclude all faint detections ($m > 21$) and edge detections (lying within 50 pixels from the edge of the image) to obtain our second data set, the SC data set, which then contains 637 937 detection thumbnails. This set is further split into training and test sets, which contain 574 143 and 63 794 detections, respectively.

3.3 Normalization

Image normalization is an important data pre-processing procedure to make sure that all inputs are consistent and all pixels have the same range of values. Due to different observing conditions, detections having the same intrinsic brightness but observed at different epochs usually have different background levels and peak values of the signals. Since the real-bogus classification is usually based only on the morphology of the detection on the difference image, we need to normalize the input (the pixel values of the subtracted cutouts) in order to avoid the classification being affected by these variations.

We normalize each input by multiplying each difference cutout pixel value by

$$f(p) = \begin{cases} [(p-\bar{p})/(p_{1.00}-\bar{p})+1], & p > \bar{p} \\ [\max(-|\bar{p}-p|/|\bar{p}-p_{0.05}|, -1)+1]/2, & p < \bar{p}, \end{cases} \quad (9)$$

where $\bar{p}$ and $p_{1.00}$ are the median pixel value and the peak value of the thumbnail, respectively. $p_{0.05}$ is the 5- percentile value of the pixels which lie below $\bar{p}$. In this normalization algorithm, we set $\bar{p}$ to be the background level with a normalized value of 0.5. We then linearly normalize the above-background pixels ($p > \bar{p}$) and below-background pixels ($p < \bar{p}$) with different normalization constants. We normalize the peak value $p_{1.00}$ to be 1 and the 5-percentile $p_{0.05}$ to be 0. Since the subtraction process might generate some outliers with extremely negative values, in order to avoid the normalization scale being affected by those outliers, we use $p_{0.05}$ instead of the minimum pixel value to normalize the below-background pixels.

3.4 Minor planet test set

To generate a test set for the classifiers we use the code of Killestein et al. (2021) to extract a set of stamps centred on known minor planets (MPs). Positions of MPs are queried using `SkyBoT` (Berthier et al. 2006) and cross-matched to difference image sources to yield a confirmed set of genuine examples. We extract stamps of size 32×32 pixels centred on the difference image detections (the DC approach described in Section 3.1), equivalent to extracting stamps centred on science sources given the lack of underlying template source. This approach yields 50 279 real detections, spanning a wide range of sky conditions, PSFs, and source magnitudes.

We also randomly gathered 92 901 human-reviewed bogus detections from the GOTO detection data base to form a bogus test set. Combining with the 50 279 real detections, the entire test set contains 143 180 detections.

4 ANALYSES AND RESULTS

4.1 Model training

We start our analysis by training our DESOM model with the DC data set (see Section 3.1). We apply different training approaches (see Section 2.3) and perform grid searching on the hyperparameter space of the DESOM model to obtain the best model configuration.

The AEs constructed with different training approaches and complexities share similar averaged reconstruction loss $L_{\rm dec}$ indicating similar performance. The loss can be used to perform model comparison. However, we cannot conclude whether the AE performs well based on its loss value alone. Additionally, we manually compare the reconstructed outputs and the raw inputs to see if the AE performance is reasonable. Fig. 7 shows some examples illustrating the AE performance. Since the main objective of the AE in this work is to generate a compressed representation for the input to train the SOM layer, the AE does not need to be perfect but has to be able to pick up enough details from the input for classification purposes. In fact, a more complex AE can improve the image reconstruction, however it also results in a larger latent space which requires a more complex SOM to learn. Therefore, we should always keep our AE simple but just good enough to reproduce the main details of the inputs in order to maximize the efficiency of the SOM training.

Although the AE trained with the DC data set provides a reasonable performance, the performance from the SOM layer is unexpectedly bad. We expect that the subtraction artifacts of a similar type should group together in the latent space. With this assumption, we should be able to conclude the different subtraction issue based on the predicted PVs. However, the predicted PVs generated by the DC-trained DESOM show a completely different morphology from the original inputs (see Fig. 8).

The poor performance of the DC-trained DESOM can be explained by the fact that the SOM prediction is not transformation (rotation and/or translation) invariant (Polsterer, Gieseke & Igel 2015;

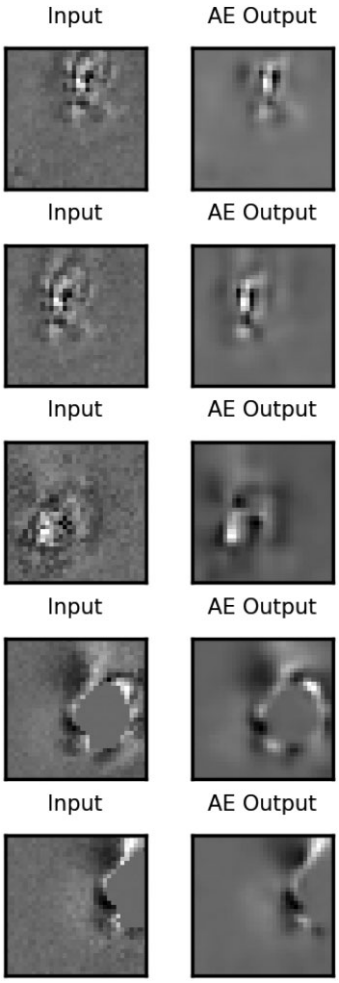

Figure 7. Examples of some cutout inputs from GOTO prototype images and their reconstructed outputs generated by the DC-trained AE. These examples show that the AE performs well on denoising with preserving most of the characteristics from the original inputs.

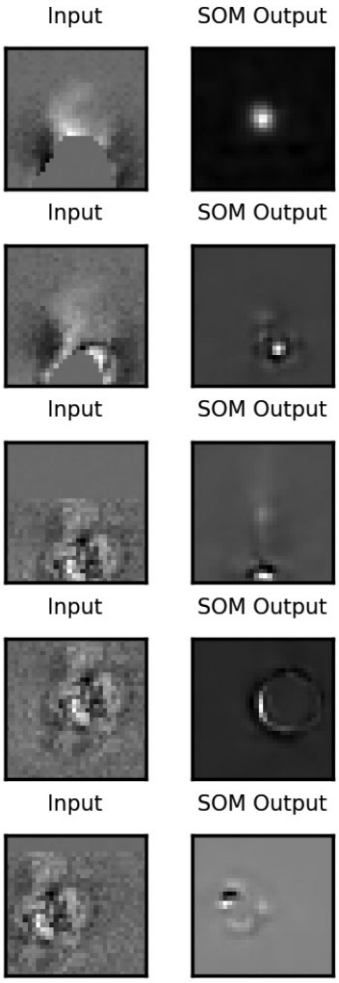

Figure 8. Examples of some cutout inputs and their decoded PVs showing the poor performance of the DC-trained DESOM. In the last three examples, the SOM layer classifies the same pattern located at different positions into three different PVs, which indicates that the prediction of SOM is not transformation invariant.

Teimoorinia et al. 2021). Thus, two inputs with the same pattern could possibly be separated far apart from each other in the latent space if only the location or the orientation of the pattern is different. As we can see in Fig. 8, while the last three inputs have identical patterns but with offsets, their PVs look completely different, and none of them is a good representation of that pattern. It implies that the DC-trained DESOM fails to group them together in the latent space. Therefore, we are motivated to train our DESOM with the SC data set (see Section 3.2).

We train our DESOM model with the SC data set using different training approaches and find that the separated training approach (Section 2.3.2) provides the best performance. We perform grid searching on the hyperparameter space of the DESOM. We compare $L_{\rm dec}$ and $L_{\rm som}$ of different model configurations. The best model configuration is shown in Table 1. The encoder in our best DESOM consists of five hidden layers, three `Conv2D` layers, and two `dense` layers. The decoder is symmetric to the encoder built with `dense` layers and `Conv2DTranspose` layers.

We also show the 30 × 30 DESOM output map in Fig. 9. The separation between two PVs on the SOM map is proportional to the Euclidean distance between those corresponding clusters in the latent space. However, we can see that the point-like PVs concentrate at both top-left and bottom-left corners, which are supposed to stay close with each other. We randomly pick one of the PVs at the top-left corner and calculate the Euclidean distance between that PV and those at the bottom-right corner. The distance is comparable to the intra-distance of the top-left PV clusters. This result indicates that the two corners are very close to each other. The DESOM

Table 1. Best DESOM model configuration.

Parameter	Value
Training set	SC data set
Training approach	Separate-trained
1st hidden layer	32 neurons (`Conv2D + MaxPooling2D`)
2nd hidden layer	64 neurons (`Conv2D + MaxPooling2D`)
3rd hidden layer	128 neurons (`Conv2D + MaxPooling2D`)
4th hidden layer	512 neurons (`dense`)
5th hidden layer	120 neurons (`dense`)
Decoder layers	`dense + Conv2DTranspose`
SOM map size	30 × 30
$T_{\rm max}$	10
$T_{\rm min}$	0.01
Training iterations	15 000

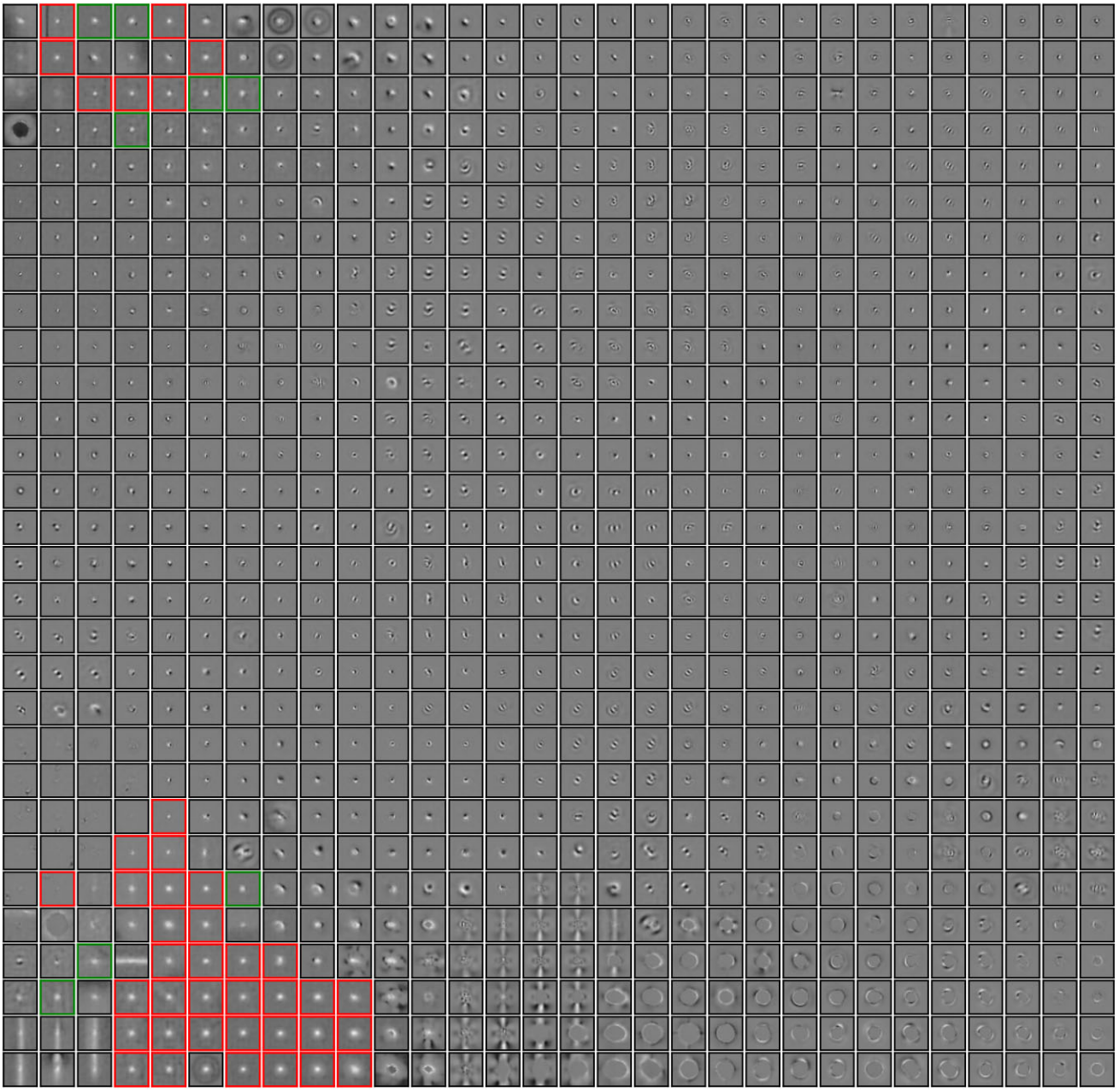

Figure 9. The decoded PVs on the DESOM map of the best model configuration showed in Table 1. The *red* bordered PVs indicate the selected PVs in the reference PV selection (see text in Section 4.2). The *green* bordered PVs indicate the PVs which are selected manually to further improve the model performance.

map shows that it successfully groups the PVs with qualitatively similar patterns together. We can see, in Fig. 10, that the PVs are able to capture the key features from those corresponding inputs.

4.2 Model evaluation

Before testing our model accuracy, we visually inspect all DESOM maps generated by the models with different training approaches and hyperparameter configurations. Since most of the DESOM maps generated with the combine-training approach (see Section 2.3.1 for more details) have two main issues, some of the PVs are random noise and some other PVs look identical, we decide to deploy the separate-training approach (see Section 2.3.2) instead. For the rest of this paper, all results are presented based on the separate-training approach.

In order to apply the DESOM model to the real-bogus classification, we need to select which PVs should be classified as real. Since each PV can be labelled as real class or bogus class, there are 2^{900} different permutations for a 30×30 DESOM map.

We compare the model performance between DESOM and GOTO-VGG by studying their receiver operating characteristic (ROC) curves. The ROC curve can be generated by gradually moving the decision boundary. However, for the DESOM model, there is no specific order to switch on and off the PVs. With different switching order, 900! ROC curves can be generated. To generate some good representatives out of all ROC curves, we decide to use GOTO-VGG predictions as the reference to assign the switching order for the

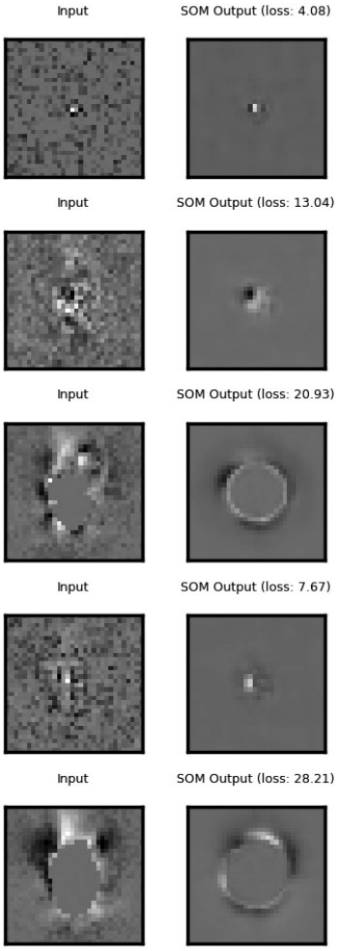

Figure 10. Examples of some cutout inputs and their decoded PVs predicted by the SC-trained DESOM. The SOM loss $L_{\rm som}$ shows a large variation depending on the complexity of the image pattern. These examples show that the DESOM model performs noise reduction and identify the general patterns of the inputs at the same time.

PVs. We apply the DESOM model to our SC test set such that each detection falls on to one of the PVs. Since each test detection comes with the real-bogus score predicted by the GOTO-VGG model, we can order the PVs by the median values of the GOTO-VGG scores. The PV with a higher GOTO-VGG median score means it is more likely to be real. To begin with, we 'switch on' all PVs. In the other words, we label all PVs to be real at first. We then 'switch off' the PVs one by one starting from the one with the lowest median VGG score. In the switch-off process, the ROC curve can be generated as the FPR decreases with the increasing missed detection rate (MDR). Here, we use our MP test set to estimate the FPR and the MDR in order to make sure that each test sample has been reviewed manually.

To obtain the best ROC curve, we experimented with switching off the PVs in different orders. We repeat the above procedure with different percentile GOTO-VGG scores instead of the median score. Fig. 12 shows that using 99- percentile GOTO-VGG score to order the PVs generates the best ROC curve. We define the reference PV selection as the one with the lowest MDR at FPR $=$ 1 per cent. The MDR at FPR $=$ 1 per cent is also called the figure of merit (FoM). With these definitions, the FoM of the reference PV selection is about 9–10 per cent (see the *red star* in Fig. 12). We also specify

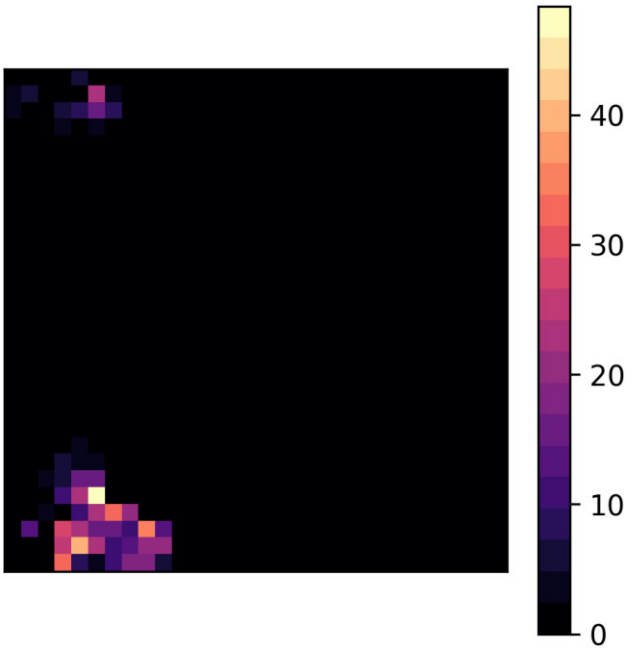

Figure 11. The heatmap presents the performance of the PVs. The colour of each cell represents the ratio between the probabilities of a real detection and a bogus detection falling into that PV.

the reference PV selection with the *red bordered* PVs in Fig. 9. By comparing with the FoM $\approx$ 6 per cent of the GOTO-VGG model, the reference PV selection performs slightly worse than the GOTO-VGG classifier.

We further modify our reference PV selection by including some extra PV manually. We inspect the selection and find that some of the unselected PVs look real. By including eight extra PVs (indicated by the *green bordered* PVs in Fig. 9) to the reference PV selection, the MDR drops to 6.6 per cent and the FPR climbs to 1.5 per cent, respectively (see the *blue star* in Fig. 12). This exercise demonstrates that further adjustments to the PV selection can improve the performance of the DESOM classifier in combination with the GOTO-VGG classifier.

In practice, since generating ROC curve is not necessary unless for model comparison, we can manually select all PVs which look like genuine detections without any help from other supervised classifiers.

In order to visualize the performance of the PVs, for each PV, we calculate the ratio between the probabilities of a real detection and a bogus detection being classified into that PV. The heat map in Fig. 11 represents the ratio values. We can see that the ratios of those selected PVs are much higher than the others, indicating that a real detection is more likely to fall into those PVs than a bogus detection does.

5 DISCUSSION

We have constructed a DESOM model to classify detections on difference images produced with the GOTO prototype instrument, with the objective of improving our ability to distinguish between real and 'bogus' detections. We compare our DESOM model with the existing GOTO-VGG classifier by discussing their strengths and weaknesses in Section 5.1. In Section 5.2, we discuss the potential usages of DESOM beyond real-bogus classification. We also discuss the limitations of the DESOM model in Section 5.3.

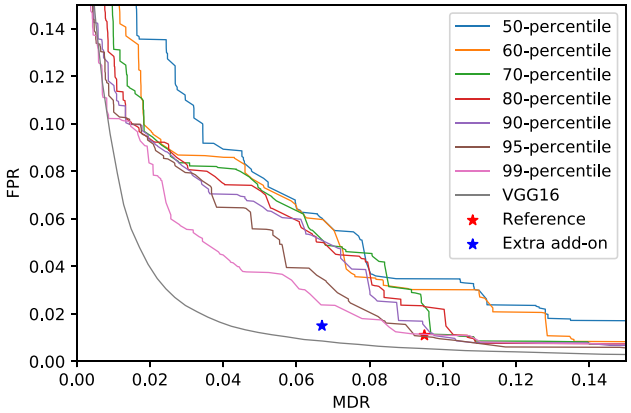

Figure 12. ROC curves generated by 'switching off' the ordered PVs on the DESOM map one by one. Different colours represent the PVs ordered by different percentile values of the GOTO-VGG score (see text in Section 4.2 for more details). The FoM of the reference PV selection is about 9–10 per cent. We also plot the ROC curve of the GOTO-VGG classifier. The *blue star* indicates the performance of the modified reference PV selection with the inclusion of eight extra PVs (see *green bordered* PVs in Fig. 9).

5.1 Comparison between DESOM and GOTO-VGG classifier

A neural network model is usually considered to be working as a 'black box'. It is very challenging to visually understand the prediction logic. Shifting the decision boundary can improve either the MDR or the FPR. However, balancing the MDR and the FPR does not fundamentally improve the classifier. To do so, we need to re-train the classifier with more data or different model configuration.

On the other hand, DESOM provides users more flexibility in the classification process. Users can simply adjust the model by selecting more or fewer real-labelled PVs on the DESOM map. Since the selection of PVs is done by visual inspection, the adjustment is usually explicable. However, unlike the GOTO-VGG classifier, DESOM is unable to generate a probability score representing how likely a detection is real.

For the GOTO-VGG classifier, we can only shift the entire decision boundary. The step size of the shift can be infinitesimal. However, we cannot control the shape of the decision boundary. In the case of a particular pattern which is always misclassified, the ideal way to improve the performance on that pattern is to only shift the corresponding part of the decision boundary, which cannot be achieved by adjusting the decision boundary of the GOTO-VGG classifier. Unlike the GOTO-VGG classifier, DESOM forms clusters to cover the entire latent space. We can simply change the label of a particular PV if we find that the corresponding PV contains too many false predictions, allowing us to fine-tune the decision boundary. On the other hand, this change is discrete rather than continuous. When we change the label of a PV, it is equivalent to swapping between the true predictions and the false predictions of the PV.

5.2 Other potential usages

The DESOM model treats real-bogus classification problem as a multiclass classification problem. With the visual inspection of the DESOM map in Fig. 9, only ∼5 per cent of the PVs should be used to represent the clusters of real detections. For the rest of the PVs, they represent the bogus detections with different morphologies and subtraction issues. With this characteristic, we can use DESOM to flag the detections based on their representative shapes.

The DESOM map can also be used to evaluate the performance of other classifiers. Different classifiers usually have different weaknesses. In order to improve the performance of a classifier, we may need to identify which types of patterns the classifier is relatively weak at. To do so, we can study the distribution function of the real-bogus score for each PV cluster. Visual comparison of the detection thumbnails located at both ends of the score distribution can help us understand what causes the confusion of the classifier within the same PV class.

5.3 Limitations of DESOM

There are two limitations using DESOM as a real-bogus classifier. First, since DESOM is a self-supervised clustering model, it cannot generate a probability score as a prediction. Second, some of the detections would be lost in the SC source extraction process (see Section 3.2).

With the above two limitations, our DESOM classifier is not ideal to replace the current GOTO-VGG classifier. However, there are two ways of implementing the DESOM output. As we have discussed in Section 5.2, DESOM can be used as a flagging system to provide extra information for each of the detections. Therefore, the original real-bogus score is preserved. Another way of implementing the DESOM model is to build a stack model with the GOTO-VGG classifier. However, a detailed discussion of this approach is outside the scope of this work.

6 CONCLUSION

In this work, we demonstrate how to apply a self-supervised learning approach to the 'real-bogus' classification problem for difference imaging. The algorithm we used is the DESOM, a combination of AE and SOM algorithms.

We use 32 × 32 normalized detection thumbnails extracted from the difference images to be the inputs of the DESOM model. We find that using the detection coordinates obtained from the science image to extract the thumbnails can significantly improve the DESOM performance.

We obtain our best DESOM model by training the AE and the SOM layer separately. The FoM of the DESOM classifier is about 9–10 per cent. Since the DESOM performance highly depends on the selection of the real PVs, we show that, by adding a few extra PVs, the DESOM classifier can further be improved (MDR = 6.6 per cent and FPR = 1.5 per cent).

The major advantage of the DESOM model is the flexibility of its PV selection, allowing the user fine-control of the shape of the decision boundary. Since the DESOM model treats the real-bogus classification problem as a multiclass problem, we suggest the best use of DESOM is to build a flagging system for detections, in combination with a probabilistic classification. On top of that, DESOM output map can be used to evaluate the performance of other typical real-bogus classifiers.

ACKNOWLEDGEMENTS

The Gravitational-wave Optical Transient Observer (GOTO) project acknowledges the support of the Monash-Warwick Alliance, Warwick University, Monash University, Sheffield University, University of Leicester, Armagh Observatory & Planetarium, the National Astronomical Research Institute of Thailand (NARIT), the Instituto de Astrofísica de Canarias (IAC), and the University of Turku. RLCS and POB acknowledge support from STFC. RB, MK, and DMS

acknowledge support from the ERC under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 715051; Spiders). VSD and MJD acknowledge the support of a Leverhulme Trust Research Project Grant. DMS acknowledges support from the Consejería de Economía, Conocimiento y Empleo del Gobierno de Canarias, and the European Regional Development Fund (ERDF) under grant with reference ProID2020010104.

DATA AVAILABILITY

Data files covering the system throughput and some of the software packages are available via public github repositories under https://github.com/GOTO-OBS/. Prototype data were mainly used for testing and commissioning and a full release of all data is not foreseen. Some data products will be available as part of planned GOTO public data releases.

REFERENCES

Abbott B. P. et al., 2016, ApJ, 826, L13
Abbott B. P. et al., 2017, ApJ, 848, L12
Alard C., Lupton R. H., 1998, ApJ, 503, 325
Andreoni I. et al., 2021, ApJ, 918, 63
Baldi P., 2012, in Guyon I., Dror G., Lemaire V., Taylor G., Silver D., eds, Proc. Mach. Learn. Res. Vol. 27, Proceedings of ICML Workshop on Unsupervised and Transfer Learning. PMLR, USA, p. 37
Bank D., Koenigstein N., Giryes R., 2020, preprint (arXiv:2003.05991)
Becker A., 2015, Astrophysics Source Code Library, record ascl:1504.004
Berger E., 2014, ARA&A, 52, 43
Berger E., Fong W., Chornock R., 2013, ApJ, 774, L23
Berthier J., Vachier F., Thuillot W., Fernique P., Ochsenbein F., Genova F., Lainey V., Arlot J. E., 2006, in Gabriel C., Arviset C., Ponz D., Enrique S., eds, ASP Conf. Ser. Vol. 351, Astronomical Data Analysis Software and Systems XV. Astron. Soc. Pac., San Francisco, p. 367
Bertin E., Arnouts S., 1996, A&AS, 117, 393
Blanchard P. K. et al., 2017, ApJ, 848, L22
Bloom J. S. et al., 2012, PASP, 124, 1175
Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J., 2016, Proc. 2016 Int. Joint Conf. Neural Netw. (IJCNN), Supernovae Detection by Using Convolutional Neural Networks. IEEE, p. 251
Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J.-C., 2017, ApJ, 836, 97
Cenko S. B. et al., 2015, ApJ, 803, L24
Chornock R. et al., 2017, ApJ, 848, L19
Coughlin M. et al., 2020, GCN Circ., 28841, 1
Coulter D. A. et al., 2017, Science, 358, 1556
Cowperthwaite P. S. et al., 2017, ApJ, 848, L17
Daniel G. G., 2013, in Runehov A. L. C., Oviedo L., eds, Artificial Neural Network. Springer, Dordrecht, p. 143
Duev D. A. et al., 2019, MNRAS, 489, 3582
Dyer M. J. et al., 2020, Proc. SPIE Conf. Ser. Vol. 11445, The Gravitational-wave Optical Transient Observer (GOTO). SPIE, Bellingham, p. 114457G
Forest F., Lebbah M., Azzag H., Lacaille J., 2019, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)
Gieseke F. et al., 2017, MNRAS, 472, 3101
Goldstein A. et al., 2017, ApJ, 848, L14
Gompertz B. P. et al., 2020, MNRAS, 497, 726
Hallinan G. et al., 2017, Science, 358, 1579
Ho A. Y. Q. et al., 2020, ApJ, 905, 98
Jin Z.-P. et al., 2013, ApJ, 774, 114
Kasliwal M. M., Korobkin O., Lau R. M., Wollaeger R., Fryer C. L., 2017, ApJ, 843, L34
Killestein T. L. et al., 2021, MNRAS, 503, 4838
Kohonen T., 1990, Proc. IEEE, 78, 1464
Kohonen T., 2001, Self-Organizing Maps, 3rd edn. Springer
Kumar P., Zhang B., 2015, Phys. Rep., 561, 1
Lamb G. P. et al., 2019, ApJ, 883, 48
Li L. et al., 2012, ApJ, 758, 27
Margutti R. et al., 2017, ApJ, 848, L20
Mong Y. L. et al., 2020, MNRAS, 499, 6009
Mong Y. L. et al., 2021, MNRAS, 507, 5463
Munro P., 2010, in Sammut C., Webb G. I., eds, Backpropagation. Springer, USA, p. 73
O'Shea K., Nash R., 2015, preprint (arXiv:1511.08458)
Piran T., 2004, Rev. Mod. Phys., 76, 1143
Polsterer K. L., Gieseke F., Igel C., 2015, in Taylor A. R., Rosolowsky E., eds, ASP Conf. Ser. Vol. 495, Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV). Astron. Soc. Pac., San Francisco, p. 81
Rau A. et al., 2009, PASP, 121, 1334
Savchenko V. et al., 2017, ApJ, 848, L15
Simonyan K., Zisserman A., 2014, preprint (arXiv:1409.1556)
Smith K. W. et al., 2020, PASP, 132, 085002
Steeghs D. et al., 2022, MNRAS, 511, 2405
Tanvir N. R., Levan A. J., Fruchter A. S., Hjorth J., Hounsell R. A., Wiersema K., Tunnicliffe R. L., 2013, Nature, 500, 547
Teimoorinia H., Shishehchi S., Tazwar A., Lin P., Archinuk F., Gwyn S. D. J., Kavelaars J. J., 2021, AJ, 161, 227
Wang W., Huang Y., Wang Y., Wang L., 2014, IEEE Conference on Computer Vision and Pattern Recognition Workshops. p. 496
Zhang B., Fan Y. Z., Dyks J., Kobayashi S., Mészáros P., Burrows D. N., Nousek J. A., Gehrels N., 2006, ApJ, 642, 354

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