Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **485,** 5831–5842 (2019)
Advance Access publication 2019 March 14

doi:10.1093/mnras/stz761

Detecting Solar system objects with convolutional neural networks

Maggie Lieu ,[1]★ Luca Conversi,[1] Bruno Altieri[1] and Benoît Carry[2]

[1]*European Space Astronomy Centre, ESA, Villanueva de la Cañada, E-28691 Madrid, Spain*
[2]*Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire Lagrange, F-06300, Nice, France*

Accepted 2019 March 11. Received 2019 February 11; in original form 2018 July 28

ABSTRACT

In the preparation for ESA's Euclid mission and the large amount of data it will produce, we train deep convolutional neural networks (CNNs) on Euclid simulations to classify Solar system objects from other astronomical sources. Using transfer learning we are able to achieve a good performance despite our tiny data set with as few as 7512 images. Our best model correctly identifies objects with a top accuracy of 94 per cent and improves to 96 per cent when Euclid's dither information is included. The neural network misses ∼50 per cent of the slowest moving asteroids ($v < 10$ arcsec h^{-1}) but is otherwise able to correctly classify asteroids even down to 26 mag. We show that the same model also performs well at classifying stars, galaxies, and cosmic rays, and could potentially be applied to distinguish all types of objects in the Euclid data and other large optical surveys.

Key words: methods: miscellaneous.

1 INTRODUCTION

The Solar system small bodies [asteroids, comets, Kuiper-belt objects (KBO)] are the remnants of the rocky and icy bodies that accreted to form the planets in the early Solar system. Their orbital size and compositional distribution are the results of the mass removal and radial mixing triggered by the planetary migration in the early history of the Solar system, and of Gyr of collisions (Bottke et al. 2002; Michel, DeMeo & Bottke 2015).

While their dynamics have provided the main constraints on the development of theoretical models over the last decade (e.g. Morbidelli et al. 2005; Raymond & Izidoro 2017) we are entering an era in which the compositional distribution of Solar systems small bodies is maybe becoming even more important (DeMeo & Carry 2013, 2014).

In particular, the populations of small to medium-sized KBO (tracers of the conditions in the outer planetary nebula) and small main belt asteroids (belonging to collisional families and hence progenitors of the near-Earth asteroids and meteorites) are too faint for current facilities (e.g. LSST Science Collaboration 2009; Spoto, Milani & Knežević 2015).

Whilst future large sky surveys such as LSST (LSST Science Collaboration 2009) will likely uncover and characterize a large proportion of these undiscovered Solar system bodies, ground-based telescopes are limited to night-time observations and good seeing conditions. With an estimated launch in 2022, ESA's upcoming visible and near-infrared space telescope Euclid (Laureijs et al. 2011) is unlike many of the current surveys that typically focus on objects within the ecliptic plane. A survey like Euclid, can expect to detect 1.4×10^5 Solar system objects (SSOs; Carry 2018), high-inclination ($i > 15°$) asteroids (for which there is currently a bias against in current census, see Mahlke et al. 2018) and possibly even some rare interstellar objects such as the recently discovered 1I/'Oumuamua (Meech et al. 2017; Katz 2018). Its simultaneous measurements in both visible and near-infrared will enable us to detect and compositionally map SSOs at the same time. It will nicely complement from the visible photometry from LSST and spectroscopy from *Gaia* (Delbo et al. 2012) in mapping the dynamics of SSOs. Identifying and removing asteroids in the data are also important for weak gravitational lensing to prevent contamination of the shear signal (Hildebrandt et al. 2017). Since Euclid will produce close to a terabyte of data per day, we need to prepare tools to deal with this big data quickly and accurately.

Machine learning is ideal approach to tackle the large data volume and the speed required to deal with upcoming Euclid data. Machine learning and neural networks have been used in astronomy for several years: Odewahn (1995) used such methods to classify the morphology types of galaxies from their properties; similarly Gulati et al. (1994) built a neural network to classify stellar spectra and on the topic of asteroids; Misra & Bus (2008) trained a neural network to predict the spectral class of asteroids from SDSS[1] data. Furthermore, convolution neural networks (CNNs) have allowed us to apply machine learning directly on astronomical images: Dieleman, Willett & Dambre (2015) used CNNs to classify galaxy morphologies, Schaefer et al. (2018) to detect strong gravitational lensing and they have even been used to estimate continuous

★ E-mail: maggie.lieu@sciops.esa.int

[1]Sloan digital sky survey https://www.sdss.org.

properties such as photometric redshifts (Pasquet et al. 2019) and galaxy cluster masses (Ntampaka et al. 2018).

In this study we apply machine learning techniques to the problem of SSO detection. The paper is structured as follows: in Section 2 we describe the data and simulations, in Section 3 we present the convolutional neural net architectures, Section 4 describes our results and we conclude in Section 5.

2 DATA

Euclid[2] is a 1.2 m optical and near-infrared space telescope that will observe ∼15 000 deg^2 of the sky (or over a third of the extragalactic sky) down to a V_{AB} magnitude ∼24.5 over its planned mission time of 6.25 yr. The Euclid survey will be carried out using the step and stare technique: the 0.5 deg^2 field of view will observe the same portion of sky four times, using an optimized dither pattern (Racca et al. 2016) and as a result of this planned mission strategy, it should be trivial to identify any moving object within the Solar system. However with the presence of cosmic rays, galaxies, and instrumental effects there is a lot of room for the misidentification of SSOs.

The Euclid payload consists of two instruments. VIS (Visible InStrument; Cropper et al. 2016) will obtain high-resolution optical imaging and NISP (Maciaszek et al. 2016) will provide photometry in three near-infrared bands as well as slitless spectroscopy measurements.

2.1 Simulations

In order to develop the science ground segment tools (pipeline, data analysis software, system infrastructure, etc.) in preparation for the Euclid launch and to assess its capabilities in meeting its scientific goals, detailed and extensive simulations of Euclid data set are carried out within the Euclid consortium. Our simulations are created using instrument simulators developed for such purposes. The simulator we use is based on the Euclid Visible InStrument Python Package (VIS-PP).[3]

We ingest simulations of SSOs to create state of the art, Euclid-like images for training our model. The simulated images are generated as follows:

(i) The simulator reads in a catalogue of objects (stars, galaxies, SSOs) with coordinates, magnitude, orientation and, for the SSOs, apparent speed.

(ii) For the objects that fall on to the CCD, the number of electrons are computed from the object's magnitude.

(iii) If the object is a galaxy it is simulated as an input snapshot taken from Hubble and then convolved with the Euclid point spread function (PSF).

(iv) If the object is an SSO, this is simulated as a trail of aligned stars. An oversampling factor of 10 is used to avoid PSF under sampling effects (e.g. if an SSO covers 10 pixels then it will be generated from 100 stars). The position of each SSO star is determined by the input speed and orientation angle, whilst the apparent magnitude of the SSO is determined by the integrated stellar flux,

$$m_* = m_{\rm SSO} + 2.5 \log_{10}(N_*)\,. \quad (1)$$

[2]http://sci.esa.int/euclid/

[3]http://www.mssl.ucl.ac.uk/~smn2/

(v) Once the noiseless image is generated, electron bleeding, ghosts, cosmic rays, charge transfer inefficiency, read-out and dark noise, conversion from electrons to ADU and additive bias are applied. We note that pointing inaccuracy and focal plane distortions are not simulated. As long as they can be fixed using the *Gaia* reference catalogue (Gaia Collaboration 2018), these effects should have no influence on the detection of SSOs.

2.2 Preparing the data

Our simulated images mimic the output of VIS – each field of view consists of four quadrants. Each quadrant consists of a CCD read-out node on the VIS instruments. The total size is 4096 × 4136 pixels, whilst each quadrant has a size 2048 × 2066 pixels. The image scale is 0.1 arcsec pixels^{-1}. Each field of view undergoes four dithering manoeuvres. The pointing displacements during the dithers have been optimized to the following, dither 2 – ΔX: 100 arcsec, ΔY: 50 arcsec, dither 3 – ΔX: 100 arcsec, ΔY: 0 arcsec, dither 4 – ΔX: 100 arcsec, ΔY: 0 arcsec. Each dither slew takes 64 s depending on the altitude and orbit, and 280 s for a field-to-field slew. We extract postage stamp cut-outs of objects of four categories; SSOs, cosmic rays, galaxies, and stars. The size of the postage stamp is chosen such that the image is filled by the object plus an additional padding value drawn from a Gaussian distribution with mean 65 pixels and a standard deviation 5 pixels. We investigate how our method performs both with (multichannel) and without (single channel) temporal information from the dithers. In the multichannel model, we combine dithers 1, 2, 3 and dithers 2, 3, 4 into two images of red giant branch (RGB) channels and extract postage stamp cut-outs centred on each object. We focus on the use of 3 out of the 4 Euclid dithers at a time due to constraints from our model (see Section 3.4 for more details), but also show that it is possible to make use of all 4 dithers (Section 4.6). For these images the pixels of galaxies and stars should appear in all three channels, whereas the pixels of cosmic rays and asteroids (aside from the very slow moving) should only appear in one channel. The differentiating characteristic feature of cosmic rays compared to SSOs are that the latter are convolved with the PSF. Examples of the postage stamps are shown in Figs 1 and 2.

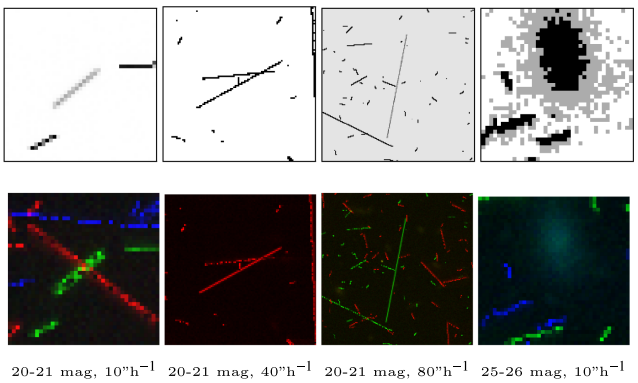

Figure 1. Examples of SSO images used in the single channel data set (top row) and those used in the three channel data set (bottom row). The SSOs are the objects closest to the centre of the image. The first three images are SSOs with magnitudes in the 20–21 mag bin and the last object is a 25–26 mag SSO. From left to right, the SSOs have speeds of 10, 40, 80, and 10 arcsec h^{-1}, respectively. The contrast levels of the images have been adjusted to enhance the appearance of object of interest, and all the images have been rescaled for illustration.

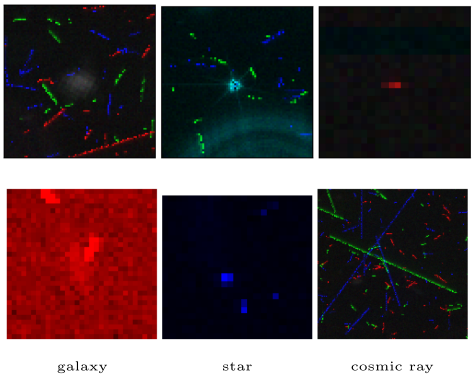

Figure 2. Examples of the other classes used in the three channel data set. The left most images are galaxies, the centre two images are stars, and the right most images are cosmic rays. The lower left image has had adjustments made to the contrast levels to enhance the appearance of the galaxy, and all the images have been rescaled for illustration.

3 METHOD

3.1 Neural networks

Artificial neural networks (ANNs) are a class of machine learning algorithm that maps some arbitrary inputs to outputs (Mcculloch & Pitts 1943). They were inspired by the visualization process of the human brain and the hierarchical perception of images. In ANNs, each neuron takes a vector of inputs $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ and applies a set of weights $\boldsymbol{w} = \{w_1, w_2, ..., w_n\}$ and a bias b to it,

$$y = \phi\left(\sum_i w_i x_i + b\right). \tag{2}$$

The input is passed to every neuron in a layer and the output of each neuron will be passed as an input to each neuron in the subsequent layer (Fig. 3). The weights and bias values are parameters of the network, and $\phi(x)$ is a non-linear activation function which determines whether or not the neuron is fired, in other words it maps the input to the response. The most commonly used non-linear activation function is the rectified linear unit (ReLU), which takes the form, $\phi(x) = \max(0, x)$. This activation function is popular since both the function itself and its derivative is quick to calculate

$$\frac{\partial \phi(x)}{\partial x} = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}. \tag{3}$$

Furthermore, unlike sigmoid or tanh activation functions, ReLU saturates only half of the time so partially solves the vanishing gradient problem.

The input ($\boldsymbol{x}$) and parameters ($\boldsymbol{w}$, b) give a predicted output $\hat{\boldsymbol{y}} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$. In supervised learning the true output $\boldsymbol{y} = \{y_1, y_2, ..., y_n\}$ is known and backpropagation (Rumelhart, Hinton & Williams 1986) is typically used to efficiently adjust the parameters to minimize a loss function $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$ (a measure of the distance between the true and predicted output). Using the gradient descent method to find the minima, the weights are iteratively updated,

$$\begin{aligned} \boldsymbol{w} &\rightarrow \boldsymbol{w} - \eta \frac{\partial L}{\partial \boldsymbol{w}} \\ b &\rightarrow b - \eta \frac{\partial L}{\partial b}. \end{aligned} \tag{4}$$

Here η is a tunable learning rate that determines the size of steps that the parameters can take on each update. A large learning rate will explore more of the parameter space but will make convergence to the minimum more difficult. A low learning rate can be more precise but will take a long time to reach the minimum. In low-dimensional parameters spaces there may be a risk of getting stuck in local minima if the learning rate is too small, however local minima are very rare in the high-dimensional parameter spaces that are common to neural networks.

The calculation of the gradients can be computationally expensive when a cycle of the entire data set is required to update the parameters (batch gradient descent) and the number of training samples is large ($\mathcal{O} \sim 10^6$). Alternatively the gradients can also be averaged over several randomly selected single samples or small samples (mini-batches) of the training data. This is known as stochastic gradient descent (SGD). It is much faster since it uses less RAM, and is better for finding multiple local-minima, however it requires the specification of the batch size. If the batch size is too small the convergence to the global minima will be slow and tends to be noisy (LeCun et al. 2012).

3.2 Convolutional neural networks

Convolutional neural networks (CNNs; LeCun et al. 1998) are deep ANNs designed for image recognition. Similarly, the CNN architecture contains multiple hidden layers, local connections between nodes and spatial invariance. The convolutional layer of a CNN takes an input image and convolves it with a small filter (kernel) whose values are weight parameters to be learnt. The filter is applied across

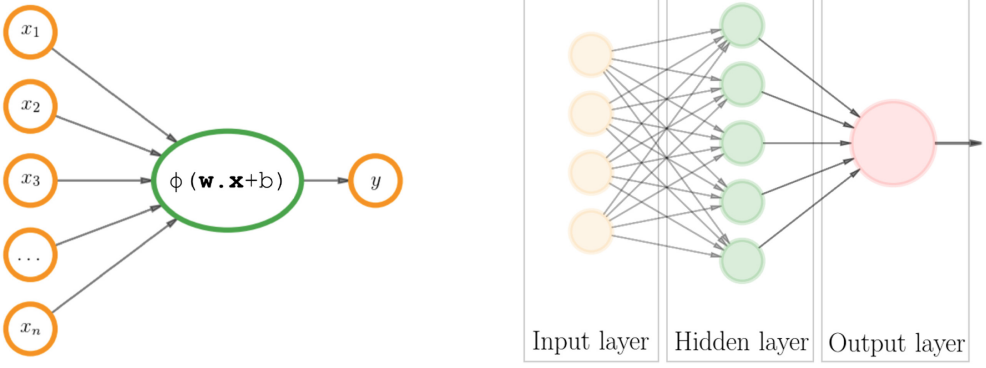

Figure 3. *Left:* Visualization of a single neuron. *Right:* Schematic diagram of an ANN with one hidden layer. Each node in the hidden layer represents a neuron. A deep neural network will have multiple hidden layers.

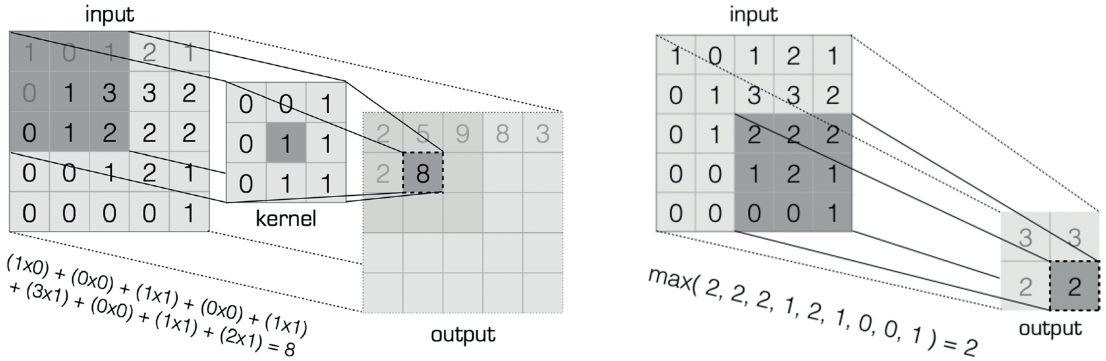

Figure 4. *Left:* Visualization of a simple convolution filter layer with a 3 × 3 convolutional filter, no padding and a stride of 1. The input represents the pixel values of an image and the kernel is applied by sliding the kernel's centre pixel value over pixels of the input. The output pixel is a weighted sum of overlapping input and kernel pixels. *Right:* Visualization of a pooling layer, with a 3 × 3 maxpool and a stride of 2. Again this is applied sliding across the input pixels. The pooling layer reduces the dimensionality of the input by mapping only the highest pixel values.

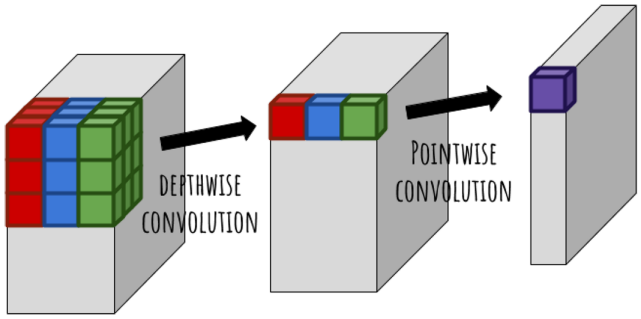

Figure 5. Depthwise separable convolution consist of first a depthwise convolution that is applied separately on each channel and then a pointwise convolution that is the same as a normal convolution (combines all channels) with a 1 × 1 kernel.

the entire image with a bias and activation function creating a feature map layer. Fig. 4 – left, shows an example of a simplified CNN. For images where there are multiple channels (e.g. RGB colour images), standard convolutional filters will combine all the channels using a weighted sum. Computationally this is not very efficient. Pooling layers are used to reduce the dimensionality of the feature maps, usually by reducing small areas to their maximum pixel value (Fig. 4 – right). An alternative to the standard convolutional filter is to use depthwise separable convolution (Chollet 2016; Fig. 5) which requires less parameters since the convolutional filters are first applied on each colour channel separately and then on each pixel across all channels. This also reduces the sensitivity to small positional shifts and distortions of a feature. Batch normalization layers are used to improve speed and generalization of the trained network by renormalizing mini-batches to their mean and variances during training, and fully connected layers connect all neurons in the previous layer and enables the mapping to a classification label.

3.3 Architectures

The layers of a neural network make up its architecture. Designing a good architecture is difficult, time consuming, and requires expert knowledge. The hyperparameters, those that need to be defined before training, include the number of convolutional filters, filter size (typically 3 × 3 pixels), padding (which defines the margin used when the convolutions are applied), stride (which determines if any pixels are skipped during the convolution), pooling layer size and dropout (a random probability of a neuron being ignored) to name a few. Whilst much research has been done to choose the best hyperparameters (e.g. Simonyan & Zisserman 2014; Szegedy et al. 2015; Murugan 2017), in practice it is trial and error, and complexity is added slowly. We use Google's Tensorflow library[4] and three different architecture models (Table 1).

Table 1. CNN architecture versions used in this paper.

Model	Number of parameters	Top-1 error[a]	Top-5 error[a]	Input size (pixel)
Inception v4	35M	80.2	95.2	299 × 299
MobileNet_v1_1.0_224	4.2M	70.7	89.5	224 × 224
NASNet-A_Large_331	88.9M	82.7	96.2	331 × 331

Note. [a]Taken from Google's internal training on the ILSVRC-2012-CLS http://www.image-net.org/challenges/LSVRC/2012/ data set using single image crop and may differ from values listed elsewhere.

Inception (Szegedy et al. 2015) is a state of the art CNN. The performance is often defined by the percentage of correct classifications featuring in the top-1 and top-5 predicted categories of each image. In 2015, `Inception_v3` was the first CNN to bypass the average human top-5 error rate of 5 per cent, obtaining a top-5 error of 3.5 per cent on the ILSVRC-2012 data set of 1000 categories. The original `Inception-v1` deep convolutional architecture was named `GoogLeNet`. It was inspired by Lin, Chen & Yan (2013)'s `Network In Network`, and was one of the first CNN architectures to implement modules of parallel operations (inception modules) instead of the traditional sequential stacking. We implement the latest version `Inception_v4` (Szegedy et al. 2016) which has since been refined to include batch-normalization, additional factorization ideas, more inception modules, and a more simplified uniform architecture.

Another CNN we use is `NASNet-A-large` (Zoph et al. 2017). Nasnet is an architecture that was created by a controller neural network `Neural Architecture Search (NAS)`. With a small data set, `NAS` uses a recurrent neural network (RNN; Zoph & Le 2016) and reinforcement learning to continuously propose

[4]https://www.tensorflow.org

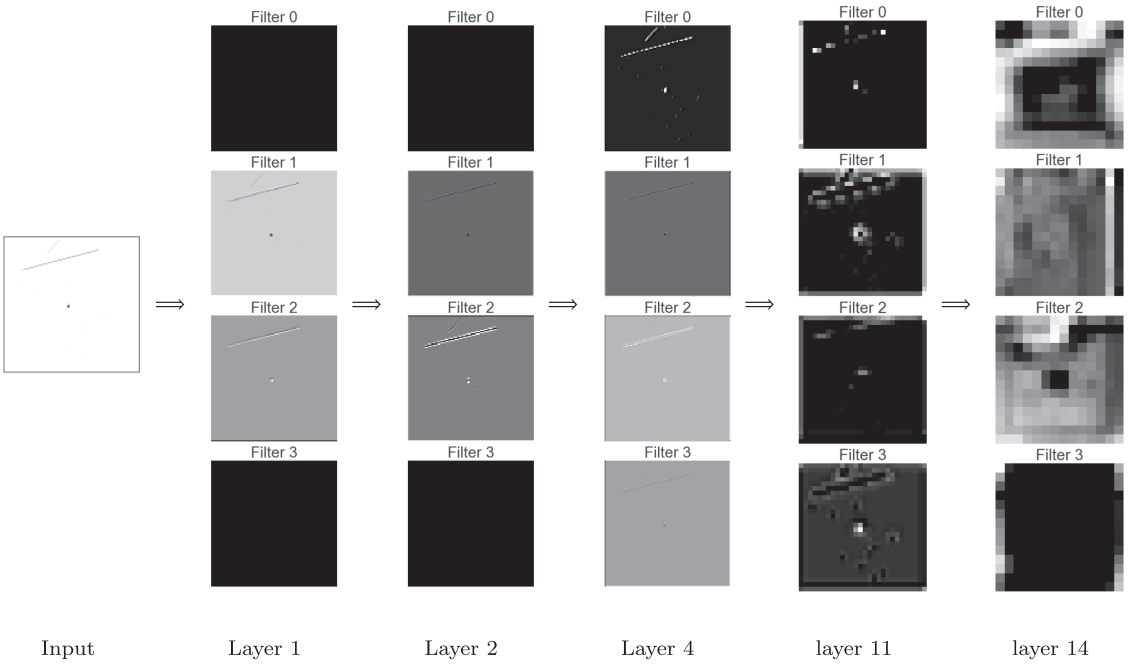

Figure 6. Activation images showing an input image (here a star) changing after convolution with the mbnet filters. Here we show only the outputs of the first four filters of each layer. Layer 1 contains 32 3 × 3 × 3 standard convolutional filters, layer 2 contains 32 3 × 3 depthwise filters, layer 4 contains 64 1 × 1 × 32 pointwise filters, layer 11 contains 256 3 × 3 depthwise filters and layer 14 contains 512 3 × 3 depthwise filters. Also note that the image size decreases with the layers due to the pooling layers.

improvements to the architecture. Since the use of `NAS` on larger data sets would be computationally expensive, it was applied to CIFAR-10, a data set of 8 × 10^7 small images in 10 classes to search for an optimal but scalable convolutional cell. Using a larger data set Imagenet 2012[5] which consists of ∼1.4 × 10^7 images and 1000 classes, `NASNet-A-large` retrained the weights of its own repetitions of these cells and filters in the penultimate layer. The final architecture consists of 18 repeated cells with 168 convolutional filters per cell. `NASNet-a-large` exceeds the performance of any human-designed model to date.

The last architecture we use is `mobilenets_1.0_224` (mbnet, Howard et al. 2017) which is the most accurate of the MobileNets architectures. MobileNets are not conventional CNNs. Traditional mobile neural networks relied on cloud computing, but the increasing computational power of mobile devices means that we no longer require dependence on an internet connection to perform deep learning tasks. Whilst other new CNNs focus on maximizing accuracy, MobileNets were designed to be very small and very fast. The standard convolutional filters in CNNs are factorized into depthwise and pointwise (1 × 1) convolutional filters which allow MobileNets to achieve competitive classification accuracies whilst optimizing for latency, size, and power restrictions. Fig. 6 is an example of how the layers of the MobileNets architecture affect an input image.

3.4 Transfer learning

Deep neural networks can have millions of parameters that can take weeks, if not months to train, transfer learning (Donahue et al. 2013) significantly reduces the time required for training without requiring GPU. In transfer learning there is no need to fully train a deep architecture. An existing architecture can be retrained and fine tuned to new classification labels (see e.g. Khosravi et al. 2018). This method has already been successfully applied in astronomy to detect galaxy mergers (Ackermann et al. 2018) and to classify galaxy morphologies (Domínguez Sánchez et al. 2018). For Euclid the expected number of SSO detections is very small at high elliptical latitudes (a few per field of view) but as many as thousands on the ecliptic galactic fields. Our simulated data are generated to be representative of the expected abundances of galaxies, stars, and cosmic rays, but we use an asteroid abundance of 0.6 arcmin^{-2}. The simulations rely on Hubble data and are both computationally expensive and volume heavy to generate, therefore in order to have a balanced data set of each class, we are restricted to a small data set ($\mathcal{O} \sim 10^3$). Consequently we implement transfer learning to avoid overfitting and to prevent getting stuck at local minima. The layers of weights are already defined through pre-training on the ImageNet 2012 data set. We only need to retrain a new top layer to include the new classes of images and to preprocess the input images to conform to the input of the architectures (see Table 1). For preprocessing we resize the postage stamps to the input size of the architecture using bilinear interpolation. We further renormalize the image by subtracting all pixel values by 128 and dividing by 128.

CNN layers consist of a three-dimensional volume of neurons with a width, height, and depth. The depth means that we can incorporate the dithers of Euclid, with each dither corresponding to a different depth layer. Since the architectures we use were pre-trained on RGB (3 channel) images, we use only 3 of the 4 dithers at any one image.

[5] http://image-net.org/

For retraining we append a new top layer that consists of a softmax activation function,

$$\phi(x_i) = \frac{\exp(x_i)}{\sum_{n=1}^{N} \exp(x_n)} \tag{5}$$

that gives us a probability of each class (i) over all (N) classes, and a fully connected (dense) layer. Since we use the softmax function, the output of our models are probabilities assigned to each class label. We take the class label with the highest probability as the predicted class. For optimization we use the standard gradient descent (`GradientDescentOptimizer`) and minimize on the mean cross entropy loss (`softmax_cross_entropy_with_logits_v2`),

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_i p_i \log q_i \tag{6}$$

where $\boldsymbol{p} \in \{\boldsymbol{y}, 1 - \boldsymbol{y}\}$ and $\boldsymbol{q} \in \{\hat{\boldsymbol{y}}, 1 - \hat{\boldsymbol{y}}\}$. We initiate the learning rate at 0.0001 and reduce it by 5 per cent every 1000 iterations. This provides good accuracy within an acceptable training time.

3.5 Regularization and augmentation

Apart from transfer learning and increasing the amount of data, there are several other tricks that can be used to prevent overfitting. Regularization by batch norm was discussed in subSection 3.2, another regularization technique is dropout, where connected nodes are randomly disconnected. We implement a 20 per cent drop-out probability. Note however that the dropout does not significantly improve fits since there are few parameters in the final layer and CNNs are applied across several locations of an image. Augmentation prevents overfitting by adjusting the properties of the training data. Resizing, cropping, rotating, transposing, and other image adjustments can also improve the results however this also significantly increases the time required for training. In Section 4.6 we investigate the use of augmentation.

We split our data into training, validation, and test sets with 70 per cent, 20 per cent, and 10 per cent, respectively. The training is run using Monte Carlo cross-validation (Xu & Liang 2001) and stochastic gradient descent with validation batches and training mini-batches of 100 images. The validation is performed every 10th iteration and the test data is only seen after the training is complete and is not used to update the parameters of the architecture.

3.6 Performance metrics

Our testing set is based on simulations so the number of images in each class can easily be defined, however we do not train our model on an astronomically representative training data set. Classification models generally do not perform well when trained on imbalanced data sets. They tend to overfit the more abundant class. This leads us to The Accuracy Paradox – a predictive model with a given accuracy, may have greater predictive power than a model with higher accuracy. Accuracy is defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{7}$$

where TP, FP, TN, FN are true positives, false positives, true negatives, and false negatives, respectively. It is clear that for an imbalanced data set with 1 SSO and 99 999 non-SSOs a high accuracy can be achieved by predicting all images to be non-SSOs, however this model would be useless for our purpose (detection of SSOs). Receiver Operating Characteristic (ROC) curves are a common way to visualize the performance of a binary classification model. It shows the true positive rate (TPR = TP / (TP + FN)) against the false positive rate (FPR = FP/ (FP + TN)) respectively.

Whilst accuracy and area under the (ROC) curve (AUC) are the typically preferred performance summary statistics, it only reflects the underlying class distribution and is not very informative when the training classes are imbalanced. It is therefore also important to consider: precision, a measure of how pure our sample is (i.e. what fraction are SSOs), recall, a measure of how complete our sample is (i.e. what fraction of all SSOs are in the sample), and the F1-score which is a weighted mixture of the two. A more relevant statistic for an unbalanced data set is the Cohen kappa score, a measure of the accuracy normalized by the imbalance of classes. Another common solution is to use a weighted loss function or feed weighted samples to the mini-batch. With transfer learning, we have sufficient data to train on a balanced training set.

In astronomy, to determine how well a classifier method performs we can look at the purity and completeness of the samples. In machine learning, purity and completeness are equivalent to precision and recall. It is clear that purity and completeness is a trade-off, the sample can be very pure if the threshold value of SSOs is very high however this will lead to a low number of classified SSOs, likewise, the sample can be very complete by classifying all images (SSOs and non-SSOs) as SSOs.

However to see how well the CNN performs on real astronomical data we need to rescale our test results to the astronomical abundance of the classes.

$$\text{Purity} \equiv \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8}$$

$$\text{Completeness} \equiv \text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$\text{Astronomical Purity} = \frac{\text{TPR} \times \text{AB}}{\text{TPR} \times \text{AB} + \text{FPR} \times (1 - \text{AB})}. \tag{10}$$

where AB is the astronomical abundance (the number of SSOs / the number of all objects).

4 RESULTS AND DISCUSSION

4.1 Considering two categories

Initially we treat each dither independently and combine cosmic rays, galaxies, and stars collectively into the category non-asteroids. The complete data set contains 3756 images of SSOs and non-SSOs. Of the three models tested; mbnet, nasnet, and inception (see Section 3.3), we surprisingly found that the mbnet architecture outperformed the other two on the test data set. Next we apply the same architectures to the three channel images (col_mbnet, col_nasnet, col_inception) and similarly the top performing architecture is mbnet despite being the quickest architecture to train. We believe this is because our data set consists of low-resolution images and mbnet having been pre-trained on low resolution (224×224 pixel images) is able to better characterize low-resolution features, whereas Inception and NASNet are both higher resolution (299×299 and 331×331, respectively) and deeper models but the added complexity is not adding any extra information and conversely suppressing the ability to generalize classification performance on new images. Fig. 7 shows the ROC curve for these models as applied to the test data set. The dashed line shows the expected ROC curve if the

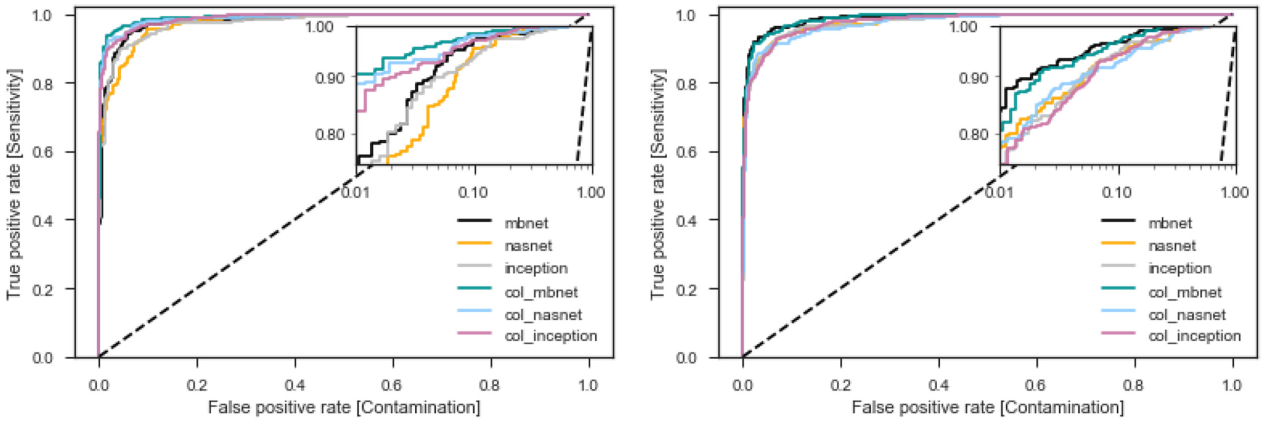

Figure 7. ROC curve for the test data set with the same log scale plot inset. *Left:* two category model. *Right:* four category model. The dashed line represents a false positive rate equal to true positive rate. This would be equivalent to the model prediction occurring purely by chance. The colours indicate different architectures and data used.

predictions from the model are down to chance. ROC curves above the dashed line perform better than a random guess and an ROC curve below the line would be performing worse than randomly guessing and is usually an indication of a bug. As expected, adding the dither information improves the performance of the neural networks.

4.2 Considering four categories

It is not uncommon within the astronomical community to write algorithms specifically to classify a single object, however CNNs, unlike other machine learning methods, do not require feature engineering which means the same network can be easily adapted to multiple class problems. It is therefore a more efficient use of personnel and resources to develop a single network that can identify multiple classes in the Euclid data than to have several networks each classifying a different astronomical object. We retrain the network with four labels; cosmic rays, galaxies, stars, and SSOs, using single and multichannel images (Fig. 7). Each class contains 3756 images. The four-label models perform slightly worse than two-label model, this is not unexpected since the two label case requires at least a threshold probability of 0.51 to be classified as an asteroid, whereas in the four-label case it could be as low as 0.26. Also there are more degeneracies between classes. Once again the MobileNet architecture is the superior model. Table 2 lists the performances of all the variations of architectures and schemes we run. From the confusion matrices, we find that stars are most often misidentified as SSOs in the network trained on single dither images, whereas, when we include the dither information this changes to cosmic rays. False negative SSOs were most often misclassified as galaxies in both scenarios.

4.3 Convergence

The chosen number of iterations used in training needs to be carefully defined to avoid under or overfitting. This is when the model has not had enough time to learn the feature or has had too much time that it is completely fine tuned to perform well on the training set but performs badly on the test set. We monitor the training error as a function of the iteration to determine an ideal stopping point (Fig. 8). It also helps to determine a suitable learning rate. Ideally, the loss as a function of time will gradually decrease. If the decline is too fast, then the learning rate is too high and if the decline is too slow then the learning rate is too low. When the loss reaches a plateau, is a good place to stop training because if the training is run for too long it will overfit. If this is the case, the loss-iteration curve will start to upturn at large times. We also monitor the training and validation accuracies over time. These two curves should closely match each other if the neural net is not under or over fitting.

4.4 Purity and completeness

The expected abundance of asteroids observed with respect to the other classes in Euclid is as low as ∼0.0001 but varies with ecliptic latitude. For this abundance we can hope to achieve 100 per cent purity for at most a 60 per cent complete sample of asteroids (Fig. 9). None the less, it is clear that this significantly improves if the abundance of asteroids fed to the model is 0.5. The neural network approach would make a good follow-up method for the confirmation of potential asteroids detected from existing methods to improve the abundance ratio. Otherwise, another way purity can be further improved is to check the 4 dithers for counterpart images (see e.g. Bouy et al. 2013; Mahlke et al. 2018). We note however that an abundance of 0.0001 is a pessimistic estimate and an advanced Euclid pipeline would be able to remove the majority of cosmic rays.

4.5 Biases

To infer whether our method is biased in any particular way we look at the distribution of the false negatives as a function of asteroid AB magnitude and speed (Fig. 10). We find that there is no particular trend with magnitude, which means the CNN can pick up even the faintest asteroids rather well however it does not perform very well in picking out the slowest moving asteroids. From the confusion matrices of the four-label models (Table 2), we see that SSOs are most likely to be misidentified as galaxies which is not surprising since they are small, extended, and convolved with the PSF just like the asteroids.

Table 2. Summary statistics for the different runs. The right most columns are the confusion matrix. This shows the number of predictions for each class against the true label when the model was applied to the test data set.

Model	Time (s)	Iterations	Test acc (per cent)	Test AUC	Precision	Recall	F1 score	Cohen κ	Predicted	Truth			
2cat										SSO	non-SSO		
Inception	9048	100k	0.922	0.976	0.938	0.899	0.918	0.843	SSO	320	21		
									non-SSO	36	350		
MobileNet	5485	100k	0.935	0.981	0.948	0.919	0.933	0.871	SSO	327	18		
									non-SSO	29	353		
NASNet	54141	100k	0.911	0.977	0.927	0.888	0.907	0.821	SSO	316	25		
									non-SSO	40	346		
col_2cat													
Inception	91557	100k	0.943	0.989	0.954	0.925	0.939	0.885	SSO	356	17		
									non-SSO	29	399		
MobileNet	5737	100k	0.956	0.992	0.973	0.935	0.954	0.912	SSO	360	10		
									non-SSO	25	406		
NASNet	89981	100k	0.945	0.989	0.952	0.932	0.942	0.89	SSO	359	18		
									non-SSO	26	398		
4cat										SSO	CR	galaxy	star
Inception	18808	100k	0.737	0.918	0.739	0.737	0.736	0.649	SSO	304	11	6	27
									CR	13	283	69	60
									galaxy	23	36	249	45
									star	5	42	35	208
MobileNet	5825	100k	0.830	0.962	0.831	0.83	0.830	0.773	SSO	313	6	6	10
									CR	9	305	43	37
									galaxy	17	20	297	33
									star	6	41	13	260
NASNet	38389	100k	0.756	0.924	0.759	0.756	0.756	0.674	SSO	299	10	5	26
									CR	18	281	72	50
									galaxy	24	26	264	38
									star	4	55	18	226
col_4cat													
Inception	11921	100k	0.726	0.911	0.730	0.726	0.727	0.634	SSO	348	22	5	17
									CR	20	249	80	32
									galaxy	23	58	235	69
									star	13	47	30	269
MobileNet	6152	100k	0.790	0.949	0.793	0.790	0.791	0.719	SSO	359	15	3	3
									CR	17	286	66	29
									galaxy	24	37	245	47
									star	4	38	36	308
NASNet	25761	100k	0.725	0.914	0.727	0.725	0.726	0.633	SSO	352	26	6	17
									CR	18	255	77	38
									galaxy	23	49	220	59
									star	11	56	47	273
Other tests													
random	296628	100k	0.504	0.500	0.000	0.0	0.0	0.0	SSO	390	384		
									non-SSO	0	0		
scratch	1219744	100k	0.851	0.916	0.866	0.828	0.847	0.703	SSO	341	66		
									non-SSO	49	318		
aug	307607	100k	0.957	0.988	0.944	0.971	0.958	0.915	SSO	368	11		
									non-SSO	22	373		
4channel	796537	100k	0.900	0.966	0.880	0.934	0.906	0.800	SSO	305	25		
									non-SSO	48	353		

4.6 Further testing

We implement further testing on a cloud virtual machine (Fig. 11). The following networks were trained on the two-category scenario and the MobileNets architecture with single depth channel as the baseline. Whereas previously we used bottleneck values to reduce the computational time, here we do not. First, to ensure the contribution from transfer learning is indeed significant, we rerun the training without loading the Imagenet pre-trained weights, instead

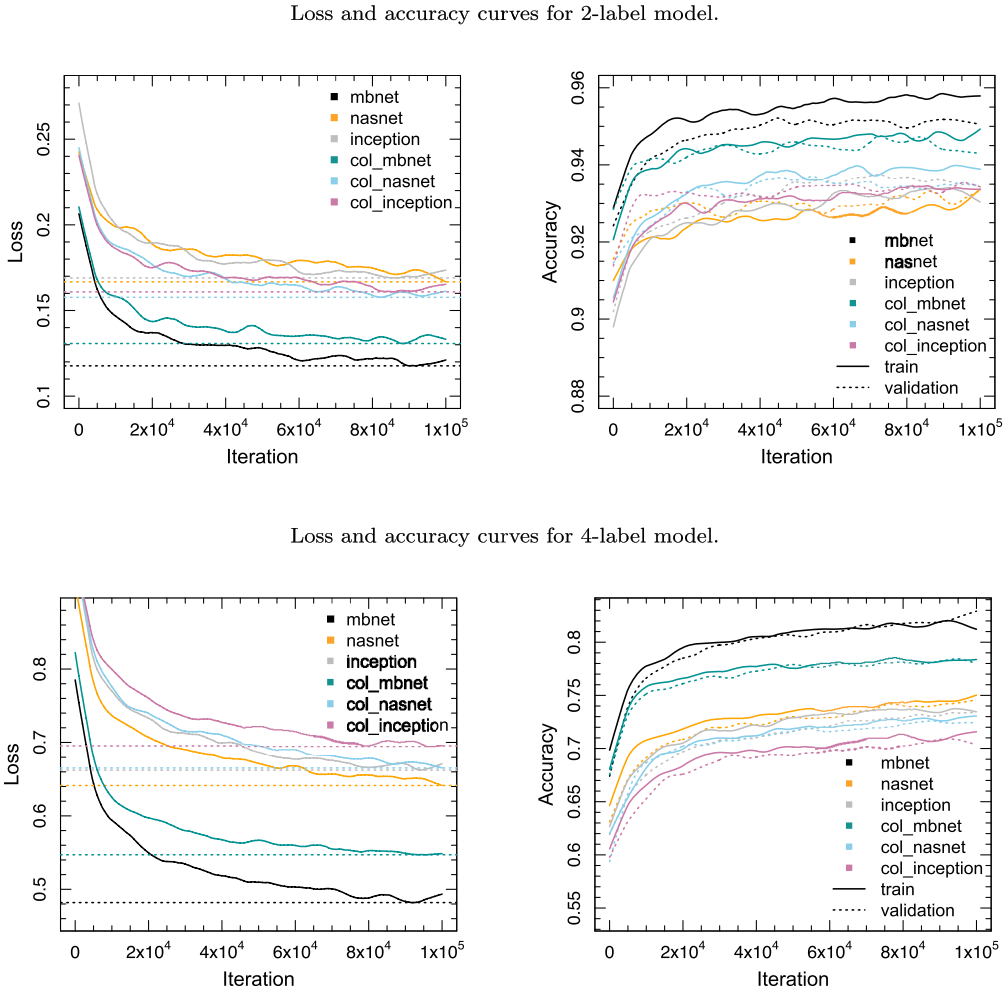

Figure 8. *Top left:* Training loss as a function of iteration for two-label runs. The dotted line shows the smallest loss and is solely for visual purposes. *Top right:* Training and validation accuracy as a function of iteration for the two-label runs. *Bottom:* the same as the top row but for the four-label models.

randomizing the weight values in the MobileNet architecture and fixing them (mbnet_random). As expected we find the classification performance to be consistent with chance. The network classified all the test images as SSOs.

Furthermore we experimented training the MobileNet architecture weight values from scratch (mbnet_scratch) with our data. On the training data and validation data this model seemed to perform the best, however on the test data it achieved 11 per cent lower than the baseline model. On further examination of the loss and accuracy plots, we see that the model begins to overfit after 10 000 iterations. We believe this is due to the small data size that is insufficient to constrain the large number of parameter.

Augmentation of the training data is known to improve the robustness and performance of the model by effectively increasing the training data volume. We train a model with random flip, random image crop of up to 10 per cent, random image scaling of up to ± 10 per cent, and random brightness of up to ± 10 per cent. The augmentation improves the accuracy by 2.2 per cent and the AUC by 0.7 per cent, but increases the training time by $\sim \times 50$. Including augmentation meant that it was not possible to use bottleneck tensors of the pre-trained network output which significantly reduces the training time. None the less, after training the neural network runs instantaneously, therefore if the computational power is not a concern then including augmentation is highly recommended.

Lastly, while it is not possible to use all four dithers as channels in the pre-trained architecture without training it from scratch, it is possible to use 4 dithers by modifying the architecture. To do this we must append additional layers before the pre-trained network that reduce the dimensionality of the input data into the correct input dimensions for the pre-trained network. We use a 2D convolutional layer with a 3×3 kernel and three filters. This gave an accuracy and AUC of 4 per cent lower and 1 per cent lower than that of the baseline model, and 6 per cent and 2 per cent lower, respectively, compared to the 3 channel MobileNet model.

4.7 Velocity predictions

CNNs can also be used to predict continuous quantities such as the velocities of asteroids. This is important for follow-up observations of SSOs. We repurpose the MobileNet architecture to predict the velocity bin of asteroids in the single dither images. The result trained in 10 518 s with a 77 per cent accuracy score on the testing data. Asteroids in the in 10–20 arcsec h^{-1} bin had the highest fraction of correct predictions (90.2 per cent) and

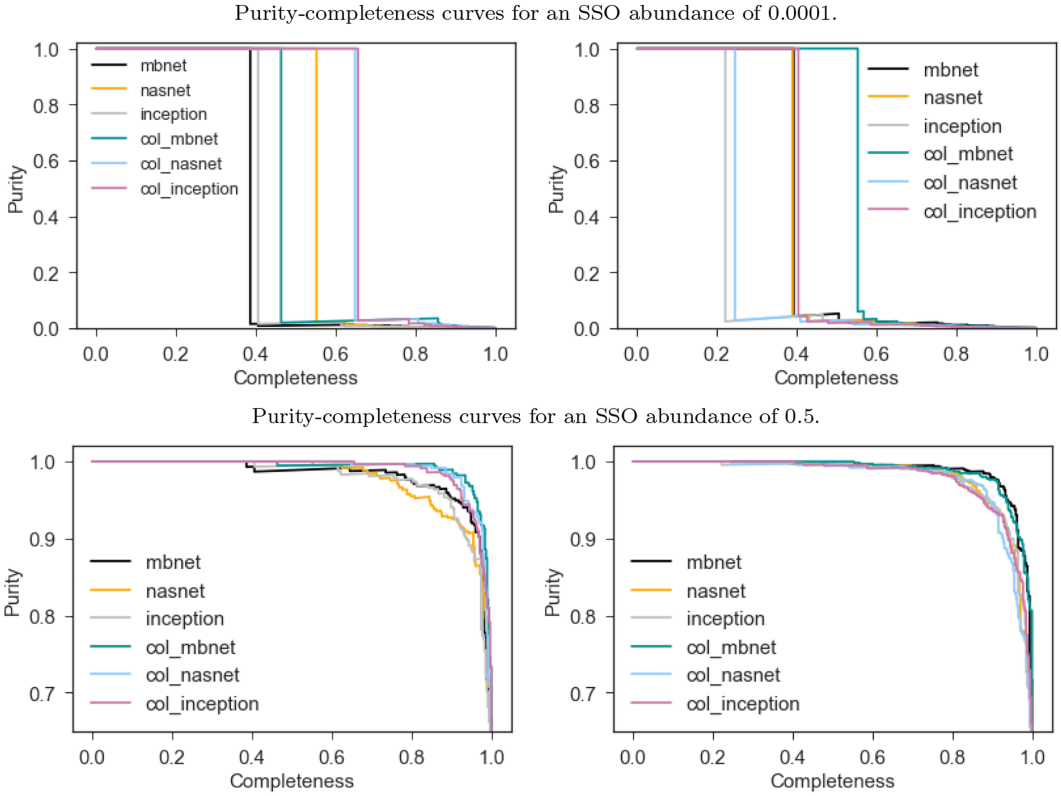

Figure 9. Purity–completeness curves. The plots on the left are the two-label models and the plots on the right are the four-label models. On the top row we use an SSO abundance of 0.0001 and on the bottom row an SSO abundance of 0.5.

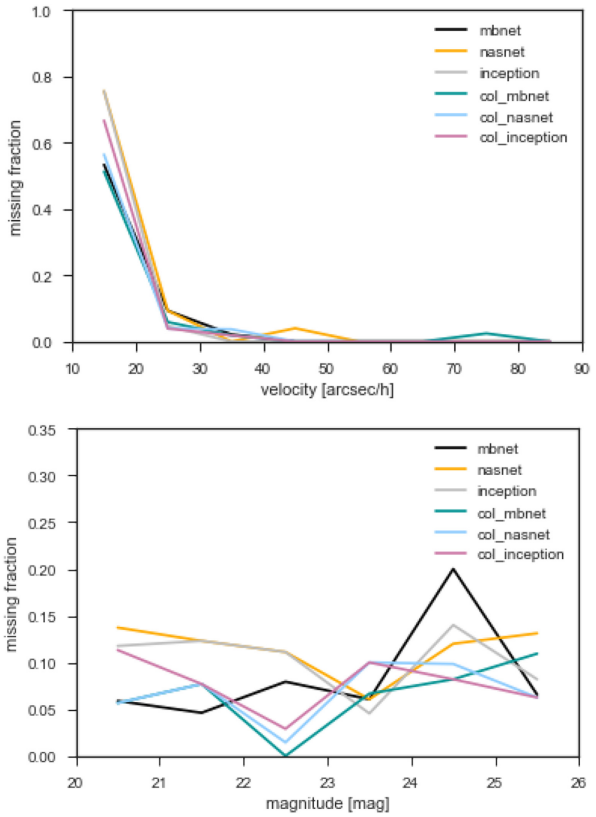

Figure 10. The fraction of SSOs incorrectly classified as a function of velocity and AB magnitude. Here we show results for the two-label models however a similar consensus is drawn from the four-label models.

whereas those in the 40–50 arcsec h^{-1} scored the worst (only 51.9 per cent were correct). It is unlikely that the lowest velocity asteroids also had the largest missing fraction in classification due to bias, and more likely to be due to low number statistics of the test data (∼50 SSOs per velocity bin). We found no trend for the accuracy of velocity prediction as a function of true velocity however, upon further investigation we found that the 40–50 arcsec h^{-1} asteroids that were incorrectly classified tended to be classified in the adjacent velocity bins (Fig. 12). Velocity prediction would be useful to further constrain the identification of asteroids as it would enable the prediction of an asteroid's location in consequent dithers, however this is beyond the scope of this paper.

5 CONCLUSIONS

We use three of the best deep CNNs currently available and apply transfer learning to retrain them for the classification asteroids. The neural networks are retrained on Euclid-like images, to identify asteroids and non-asteroids. The MobileNet model is found to be the best performing architecture on this data set and is also the quickest CNN to train. Our model reaches top accuracies of 94 per cent and marginally increases to 96 per cent when we include an additional 2 of Euclid's 4 dither images. The model is shown to further improve on the addition of augmentation (top accuracy 96 per cent on the single dither images), on the other hand we find that using all 4 dithers or training scratch does not achieve high performance. We suspect that this is due to the limitations of the training data set making it more difficult to constrain the model parameters.

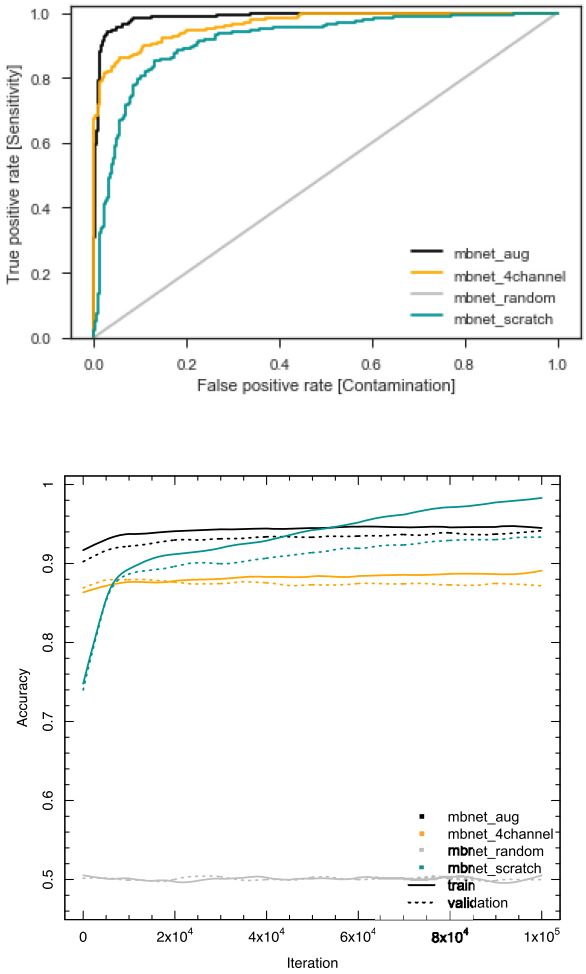

Figure 11. Further architecture tests. The top plot shows the ROC curve of the test data for the networks using augmentation, 4 dithers, random weights, and training from scratch. The bottom plot shows the accuracy of the training and validation as a function of iteration for the same networks.

Our model is robust to the expansion of more labels. When the non-SSO class is replaced with classes; galaxies, stars, and cosmic rays, the AUC (area under the receiver operating characteristic curve) score on asteroids decreases slightly by 2 per cent. This means that method has the potential to perform well on the classification of all objects in Euclid, provided that a more complex training data set of simulations were to be produced. This includes both astronomical objects and instrumental artefacts such as supernova, satellite trails, and ghosts. Our research suggests that the models with and without dithering information are complementary to each other. We find that without using dithering information, our asteroid sample was more contaminated by stars, however including the dithering information the predicted asteroids were more contaminated by cosmic rays.

Our CNNs perform well on even the faintest asteroids but are more susceptible to missing the slowest moving asteroids. The true abundance of asteroids will be low, therefore to achieve a high purity and completeness sample we could preprocess the data with a method such as SEXTRACTOR (Bertin & Arnouts 1996) for an initial removal of stars, galaxies, and cosmic rays. None the less, the Euclid pipeline will likely remove the majority of cosmic rays.

Finally, we show that the same technique can be applied to predict the velocities of the asteroids to 77 per cent accuracy and with no obvious signs of bias. This opens up a large number of possibilities in the future for analysing astronomical data from Euclid and other big data volume missions such as LSST.

ACKNOWLEDGEMENTS

The authors would like to thank Justin Alsing, David Hogg, Will Farr, Stephen Feeny, Michelle Lochner and Matej Kosiba for useful discussions. ML acknowledges a European Space Agency Research Fellowship at the European Space Astronomy Centre (ESAC) in Madrid, Spain. We also thank the anonymous referee for the constructive comments.

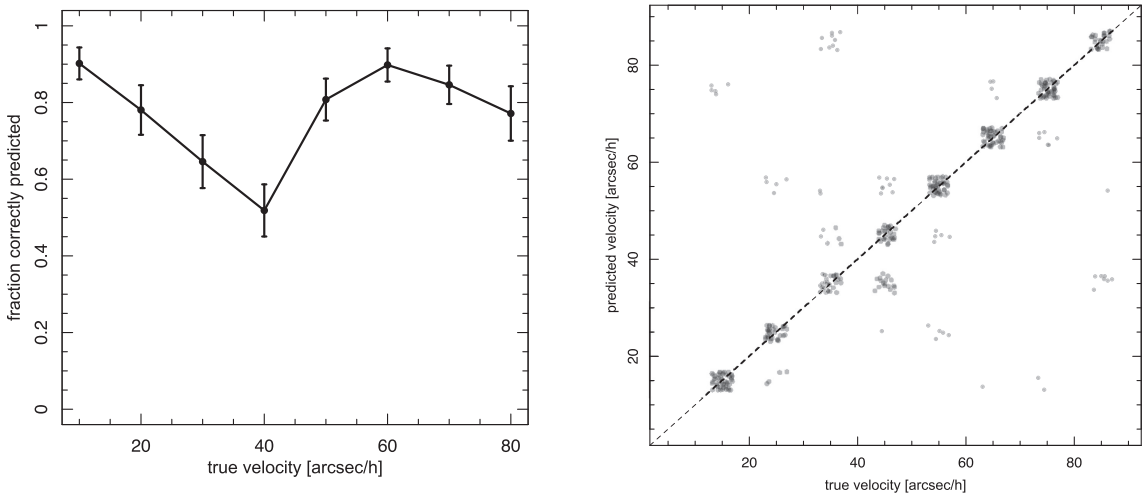

Figure 12. *Left:* Fraction of correctly predicted asteroid velocities against true velocity. The error bars show the standard error. *Right:* Predicted velocity bin versus true velocity bin of asteroids. The dashed line represents equality, and the points are jittered in both x and y for clarity.

REFERENCES

Ackermann S., Schawinski K., Zhang C., Weigel A. K., Turp M. D., 2018, MNRAS, 479, 415

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

Bottke W. F. Jr, Cellino A., Paolicchi P., Binzel R. P., 2002, Asteroids III. University of Arizona Press, Tucson

Bouy H., Bertin E., Moraux E., Cuillandre J.-C., Bouvier J., Barrado D., Solano E., Bayo A., 2013, A&A, 554, A101

Carry B., 2018, A&A, 609, A113

Chollet F., 2016, preprint (arXiv:1610.02357)

Cropper M. et al., 2016, in MacEwen H. A., Fazio G. G., Lystrup M., Batalha N., Siegler N., Tong E. C., eds, Proc. SPIE Conf. Ser. Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 99040Q

Delbo' M., Gayon-Markt J., Busso G., Brown A., Galluccio L., Ordenovic C., Bendjoya P., Tanga P., 2012, Planet. Space Sci., 73, 86

DeMeo F. E., Carry B., 2013, Icarus, 226, 723

DeMeo F. E., Carry B., 2014, Nature, 505, 629

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

Domínguez Sánchez H. et al., 2018, MNRAS, 484, 93

Donahue J., Jia Y., Vinyals O., Hoffman J., Zhang N., Tzeng E., Darrell T., 2013, preprint (arXiv:1310.1531)

Gaia Collaboration, 2018, A&A, 616, A1

Gulati R. K., Gupta R., Gothoskar P., Khobragade S., 1994, ApJ, 426, 340

Hildebrandt H. et al., 2017, MNRAS, 465, 1454

Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., 2017, preprint (arXiv:1704.04861)

Katz J. I., 2018, MNRAS, 478, L95

Khosravi P., Kazemi E., Imielinski M., Elemento O., Hajirasouliha I., 2018, EBioMedicine, 27, 317

Laureijs R. et al., 2011, preprint (arXiv:1110.3193)

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

LeCun Y. A., Bottou L., Orr G. B., Müller K.-R., 2012, Efficient BackProp. Springer, Berlin, Heidelberg, p. 9

Lin M., Chen Q., Yan S., 2013, preprint (arXiv:1312.4400)

LSST Science Collaboration, 2009, preprint (arXiv:0912.0201)

Maciaszek T. et al., 2016, in MacEwen H. A., Fazio G. G., Lystrup M., Batalha N., Siegler N.1, Tong E. C., eds, Proc. SPIE Conf. Ser. Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 99040T

Mahlke M. et al., 2018, A&A, 610, A21

Mcculloch W., Pitts W., 1943, Bull. Math. Biophys., 5, 127

Meech K. J. et al., 2017, Nature, 552, 378

Michel P., DeMeo F. E., Bottke W. F., 2015, Asteroids IV. The University of Arizona Press, Tucson

Misra A., Bus S. J., 2008, AAS/Division for Planetary Sciences Meeting Abstracts #40. p. 508

Morbidelli A., Levison H. F., Tsiganis K., Gomes R., 2005, Nature, 435, 462

Murugan P., 2017, preprint (arXiv:1712.07233)

Ntampaka M. et al., 2018, preprint (arXiv1810.07703)

Odewahn S. C., 1995, PASP, 107, 770

Pasquet J., Bertin E., Treyer M., Arnouts S., Fouchez D., 2019, A&A, 621, A26

Racca G. D. et al., 2016, in MacEwen H. A., Fazio G. G., Lystrup M., Batalha N., Siegler N., Tong E. C., eds, Proc. SPIE Conf. Ser. Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 99040O

Raymond S. N., Izidoro A., 2017, Icarus, 297, 134

Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nature, 323, 533

Schaefer C., Geiger M., Kuntzer T., Kneib J. P., 2018, A&A, 611, A2

Simonyan K., Zisserman A., 2014, preprint (arXiv:1409.1556)

Spoto F., Milani A., Knežević Z., 2015, Icarus, 257, 275

Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z., 2015, preprint (arXiv:1512.00567)

Szegedy C., Ioffe S., Vanhoucke V., Alemi A., 2016, preprint (arXiv:1602.07261)

Xu Q., Liang Y.-Z., 2001, Chemometr. Intell. Lab. Syst., 56, 1

Zoph B., Le Q. V., 2016, preprint (arXiv:1611.01578)

Zoph B., Vasudevan V., Shlens J., Le Q. V., 2017, preprint (arXiv:1707.07012)

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