

ENSEMBLE MACHINE LEARNING MODEL FOR AUTOMATED ASTEROID DETECTION

RAUL URECHIATU¹, MARC FRINCUI², OVIDIU VADUVESCU³, COSTIN BOLDEA⁴

¹*Department of Computer Science, Faculty of Mathematics and Computer Science, West University of Timisoara, Timisoara, Romania Email: raul.urechiatu97@e-uvt.ro*

²*Department of Computer Science, School of Science and Technology, Nottingham Trent University, Nottingham, United Kingdom Email: marc.frincu@ntu.ac.uk*

³*Isaac Newton Group (ING), Apt. de correos 321, Santa Cruz de La Palma, Canary Islands, Spain Email: ovidiu.vaduvescu@gmail.com*

⁴*Department of Computer Science, Faculty of Sciences, University of Craiova, Craiova, Romania Email: cboldea@inf.ucv.ro Email: ovidiu.vaduvescu@gmail.com*

Abstract. The potential threat of Near Earth Objects (NEO) requires a constant survey of the night sky to discover potentially dangerous objects and assess their future impact odds. Several ongoing surveys relying on human operators or automated techniques exist. One such example is the EURONEAR blink mini-survey project which over time developed from a pure manual approach to detecting asteroids to semi-automatic methods (NEARBY) using image processing and service-oriented approaches. In this paper, we propose an extension of NEARBY based on an ensemble model comprising three state-of-art machine learning models, some used in similar approaches. The proposed model is designed for a binary classification problem where candidate images may contain an asteroid in their center. Validation on a real-life dataset comprising 11,000 images shows that our ensemble model is capable of recovering about 55% of the asteroids missed by the previous NEARBY automated process while at the same time having a 0.88 recall on the asteroids already detected by NEARBY. Used together with NEARBY our model increased the detection rate from 89% to 95%.

Key words: Asteroids – Machine Learning – Ensemble Models – Automation.

1. INTRODUCTION

The potential threat of Near Earth Objects (NEO) requires a constant survey of the night sky to prevent future impacts. Existing NEO projects such as EURONEAR (Vaduvescu and Curelaru, 2006) work towards identifying new NEOs or refining the orbits of existing potentially hazardous asteroids. Recent research concluded that there are about 1,000 NEOs larger than 1 km and up to about 70,000 NEOs larger than 100 m all with a trajectory taking them closer than 50 million km from Earth (Tricarico, 2017; Granvik *et al.*, 2016; Harris and D'Abramo, 2015). Out of these, above 30,000 are currently being tracked (NASA-JPL, 2022).

Over the last decade, several pipelines for automated detection and processing of moving targets have been implemented. Recent examples include NEARBY (Ste-

fanut *et al.*, 2018; NEARBY, 2021) and Umbrella (Stănescu and Văduvescu, 2021). The latter study provides a useful comparison between various approaches (Astrometrica manual and automatic, NEARBY, and Umbrella) on 15 different Wide Field Camera fields obtained from Isaac Newton Group of Telescopes with results showing that the detection accuracy depends on the dataset (and implicitly seeing conditions) with no clear best method.

1.1. RELATED WORK

These automated methods rely on image processing techniques without learning from past examples as human operators do. This leads to an increase in the number of false positives (FP) (*i.e.*, detected objects wrongly identified as asteroids) and false negatives (FN) (*i.e.*, missed NEOs). As a result, a Machine Learning (ML) algorithm capable of learning from previous examples could help reduce the number of false detections and provide results closer to that of a human operator. Unfortunately, not much work has been done in terms of ML for asteroid detection. Waszczak *et al.* (2017) introduces a Random Forest (RF) algorithm with 15 features for filtering out false positive streak detections. The system is executed on multi-core systems for faster processing and the authors show the impact of object magnitude and image size on the false positive rate. Several Convolutional Neural Networks (CNNs) are tested against Euclid data by Lieu *et al.* (2019) and results show a precision $\in [0.739, 0.973]$ (cf. Eq. 1) and recall $\in [0.737, 0.935]$ (cf. Eq. 2) depending on the CNN model and the number of classes (two or four). Another RF algorithm with 47 features extracted from images is compared against an Isolation Forest unsupervised algorithm in Lin *et al.* (2017). The RF approach has a recall or true positive rate (TPR) of about 99% and a false positive rate (FPR) of about 5% on multiple classes of objects.

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = TPR = \frac{TP}{TP + FN} \quad (2)$$

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

1.2. PROPOSED SOLUTION AND INTEGRATION IN NEARBY

NEARBY (see Fig. 1) is an automated asteroid detection software comprised of a processing pipeline based on several astronomy image reduction platforms (Stefanut *et al.*, 2018) which include IRAF (Tody, 1986), SExtractor (Bertin and Arnouts,

1996), SCAMP (Bertin, 2006), and SWARP (Bertin, 2010). The NEARBY Asteroids Detection Algorithm identifies asteroid trajectories from the elements detected by SExtractor in the processed images. The output is a list of potential asteroids that are to be validated for FPs by a human operator.

Previous tests on a suite of methods have shown that the NEARBY platform was capable of detecting between $\approx 81\%$ (Stefanut *et al.*, 2018) and $\approx 89\%$ (Stănescu and Văduvescu, 2021) of the total number of asteroids detected by human operators. Combined with the fact that other image processing-based automated detection methods, such as Umbrella, have been shown to detect asteroids missed by NEARBY (Stănescu and Văduvescu, 2021) it is clear that there is a need to improve its performance by reducing the number of FNs specifically.

Our proposed extension is an ensemble method comprised of three state-of-the-art neural networks (see Sect. 3): Inception (also used by Lieu *et al.* (2019)), ResNet-50, and Xception (successor of Inception). Before being sent to the ensemble model, the FITS images are preprocessed by applying various filters to increase the detection rate by reducing model overfit. Since we are concerned only with classifying an image as containing or not an asteroid, we deal with a binary classification problem.

The screenshot shows the NEARBY web interface. At the top, there is a navigation bar with the NEARBY logo and a user profile icon. Below it is a breadcrumb trail: Experiments / INT_May18 / Nights / n1-20180502 / Fields / E101 / Projects / PRJ02 / Validate Field E101. A sidebar on the left contains a calendar with dates 27, 18, 9, 0, and 0. The main content area is titled 'MPC Report' and includes a 'Download...' button, 'View distortion maps', and 'View medians'. The report is for object 'EM00001' and shows statistics: UNDEFINED: 0, VALID: 4, INVALID: 0. Below this, there is a table of MPC info and a table of parameters (E, FWHM, THETA). A thumbnail image of the asteroid is shown with a red crosshair and validation controls: 'Validate Object', 'Invalidate Object', 'Prev', 'Next', and 'Show lines details'.

MPC Info	E	FWHM	THETA	
EM00001 C2018 05 02 88455 10 00 43.94 +12 10 40 0 21.8 R 950	0.39	2.33	54.10	
EM00001 C2018 05 02 89031 10 00 44.05 +12 10 39 6 21.4 R 950	0.26	1.51	53.73	
EM00001 C2018 05 02 89606 10 00 44.14 +12 10 39 0 21.5 R 950	0.32	3.02	20.84	
EM00001 C2018 05 02 90112 10 00 44.24 +12 10 38 4 21.9 R 950	0.19	1.55	20.77	
Trajectory: MIU = 0.20, PA = 250.1	Averages:	0.29	2.10	37.36

Fig. 1 – Validation and reporting module of the NEARBY web interface.

2. PROCESSING PIPELINE

The flowchart of our processing pipeline can be seen in Fig. 2. The entire process consists of two steps: an offline step involving building the model from information extracted from the FITS file and an online process consisting of real-time image classification based on the trained model.

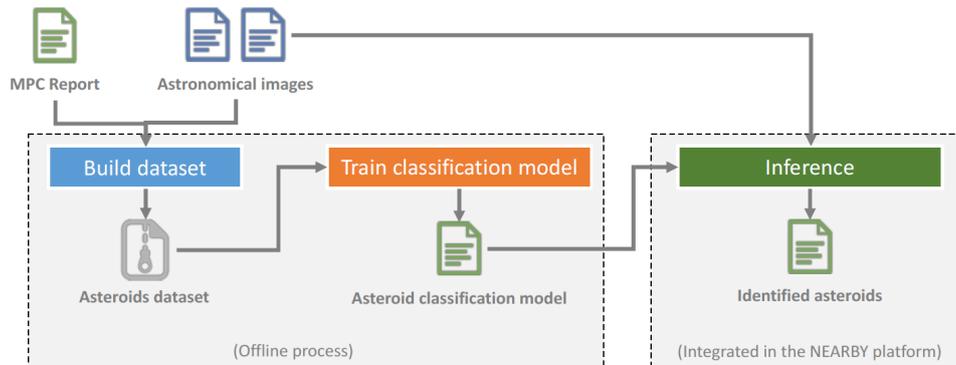


Fig. 2 – Overview of the proposed solution.

Building the trained model Once the astronomical images are obtained and the ground truth is extracted based on an existing method (*e.g.* manual blink using Astrometrica, NEARBY) the initial image dataset is extended with preprocessed images based on various filters as shown next. This enhanced dataset is sent to the ensemble model for training and validation. The outcome of this step represents our classification model.

Real-time asteroid detection The trained model is then integrated into the NEARBY platform pipeline and used for detecting asteroids by feeding it astronomical images in FITS format. At this stage, the automated detection is used similarly to a filter easing a human operator’s work by returning only the detections (TPs and FPs) which represent a fraction of the total number of detections. It is obvious that a high rate of FNs (hidden in the disregarded images) leads to a considerable number of missed asteroids as it impacts the number of TPs and FPs.

2.1. DATA PREPROCESSING

Next we describe the algorithms and techniques used for preprocessing the FITS images to minimize the training overfit while increasing the recall value. A low number of training images can cause overfitting. This means that the neural network will learn to predict with high accuracy similar cases while underperforming on a completely new dataset.

In our images, potential asteroids will appear centered, while the stars apparently jump in sequential images of the same field. The images are centered on the object by NEARBY and this is due to an agreement to display the image using the stacking technique as seen in Fig. 3. We stack 3–4 images to obtain one stacked image that will be used in training and validating the model. Having the potential target

always in the center leads to a trivial problem of searching and validating stacked images that have apparently stationary objects in their center. However, as we have found out in searching the test dataset there are cases where these detections are not asteroids but noise and other artifacts. Section 4 will present some interesting cases.

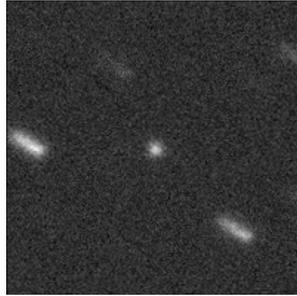


Fig. 3 – Stacked image with an asteroid in the center and stars as trails.

2.2. DATASET

We have used a dataset comprising 11,106 images taken on several runs between October 2019 and October 2020 with the Wide Field Camera on the Isaac Newton Telescope in La Palma, Canary Islands. These were first analyzed with the NEARBY tool identifying 2,663 (TP) and missing 544 (FN) asteroids, giving a recall of 0.83. Consequently, we split the data into training (80%) and testing data (20%). Once training was complete, we also tested the network on a different dataset comprised of another 2,385 different images.

Figure 4 shows some raw image examples displayed with various options.

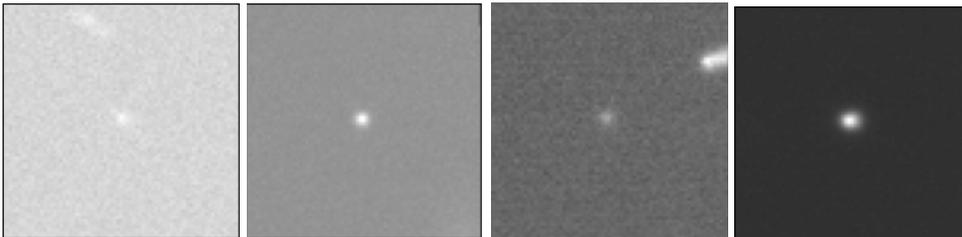


Fig. 4 – Examples of different images with varying background intensity and noise.

2.3. WEAK MODELS PARAMETERS

The ensemble model was implemented using Python 3.8 and TensorFlow 2.5 with CUDA GPU support.

Table 1

Parameters used for training the weak models.

batch size	16
loss function	binary cross-entropy
optimizer	Adagrad
epochs	30

Table 1 shows the parameters used for training the weak models that are later used in the ensemble (see Sec. 3).

The *batch size* represents the number of samples that are propagated through the neural network before updating its weights. These are values modified during the training process to approximate a function that solves the optimization problem, *i.e.*, enables the neural network to learn. The *loss function* describes the error between the neural network’s prediction and the actual classification label. The *optimizer* is an algorithm that adjusts the amount by which individual weights are changed.

Except for the optimizer, the parameters remain the same for the ensemble model. The optimizer used for it was *Adam*.

The number of *epochs* was set to 30 and early stopping was used to halt the training process in case no improvement occurred for more than 10 epochs.

The configuration was chosen empirically, the optimizer parameter having the smallest effect on the performance of the models.

2.4. NORMALIZATION

The pixel values can vary a lot, depending on the pixel depth as indicated by the BITPIX value from the FITS header with integer values in the $[0, 2^{BITPIX} - 1]$ range. In our case, the BITPIX value is 32. This high distribution of values makes learning difficult for neural networks which work better with normalized values. To normalize values we have used the following formula:

$$f(x_i) = \frac{x_i - min}{max} \quad (4)$$

where x_i is the value of the current pixel, *min* is the minimum pixel value from the image and *max* is the maximum pixel value in the image.

After normalizing the values we observed an increase in recall by $\approx 10\%$ while reducing at the same time the overfit.

Figure 5 depicts the pixel values before and after normalization. This example is a cropped 20×20 pixels image with an asteroid in the center (red area). Based on the normalized values, the neural network will learn to better distinguish areas containing potential asteroids.

0.31324	0.32992	0.32131	0.33531	0.35199	0.34607	0.33262	0.34015	0.31485	0.31808	0.33315
0.32939	0.34015	0.35414	0.37836	0.39505	0.40258	0.36168	0.34984	0.31432	0.32939	0.33692
0.32131	0.35953	0.40958	0.50215	0.55328	0.51292	0.42573	0.39290	0.35522	0.32670	0.30840
0.34446	0.37352	0.47147	0.65447	0.83208	0.74112	0.55059	0.44618	0.37513	0.33477	0.32185
0.35576	0.38698	0.51453	0.73197	1.00000	0.96233	0.69160	0.48385	0.36168	0.33262	0.32078
0.35038	0.38321	0.49623	0.71959	0.97255	0.96555	0.74273	0.51561	0.39882	0.34553	0.31163
0.34284	0.36222	0.43111	0.57481	0.76050	0.78364	0.64532	0.48547	0.38698	0.32400	0.31270
0.34123	0.35199	0.38375	0.42734	0.53337	0.57104	0.51561	0.42896	0.35899	0.34446	0.32508
0.34015	0.33746	0.35684	0.37191	0.38805	0.39882	0.39612	0.38213	0.33800	0.30786	0.33423
0.31485	0.30732	0.32400	0.33746	0.34338	0.35576	0.33746	0.33746	0.31970	0.31001	0.32185
0.32131	0.31862	0.31593	0.33315	0.34769	0.34338	0.32347	0.32777	0.33208	0.30947	0.31647
582.00000	613.00000	597.00000	623.00000	654.00000	643.00000	618.00000	632.00000	585.00000	591.00000	619.00000
612.00000	632.00000	658.00000	703.00000	734.00000	748.00000	672.00000	650.00000	584.00000	612.00000	626.00000
597.00000	668.00000	761.00000	933.00000	1028.00000	953.00000	791.00000	730.00000	660.00000	607.00000	573.00000
640.00000	694.00000	876.00000	1216.00000	1546.00000	1377.00000	1023.00000	829.00000	697.00000	622.00000	598.00000
661.00000	719.00000	956.00000	1360.00000	1858.00000	1788.00000	1285.00000	899.00000	672.00000	618.00000	596.00000
651.00000	712.00000	922.00000	1337.00000	1807.00000	1794.00000	1380.00000	958.00000	741.00000	642.00000	579.00000
637.00000	673.00000	801.00000	1068.00000	1413.00000	1456.00000	1199.00000	902.00000	719.00000	602.00000	581.00000
634.00000	654.00000	713.00000	794.00000	991.00000	1061.00000	958.00000	797.00000	667.00000	640.00000	604.00000
632.00000	627.00000	663.00000	691.00000	721.00000	741.00000	736.00000	710.00000	628.00000	572.00000	621.00000
585.00000	571.00000	602.00000	627.00000	638.00000	661.00000	627.00000	627.00000	594.00000	576.00000	598.00000
597.00000	592.00000	587.00000	619.00000	646.00000	638.00000	601.00000	609.00000	617.00000	575.00000	588.00000

Fig. 5 – Example of a pixel map before (bottom) and after (top) normalization.

2.5. AUGMENTATION

The next step in image preprocessing is to augment the dataset with additional images. Augmentation is a technique used in machine learning when the neural network has to work with a limited dataset and helps **obtaining more training data** by providing different views of the same data.

We augmented the dataset by generating 6 times as many images through image filters and transformations:

- **Rotation:** by rotating the image 90° three times either clockwise or counter clockwise;
- **Flipping:** horizontally or vertically;
- **Median filtering:** helps for smoothing the image;
- **Gaussian or salt and pepper filtering:** can be used for adding noise.

These methods are aimed at reducing overfit and providing a richer training set for the neural networks. However, if the original data is not diverse enough these methods can contribute to error propagation. For instance, if all the 11,000 images contain the same type of noise as the top left image in Fig. 4, by rotating them 3 times we end up with 44,000 images used for training a network unable to make accurate classifications on noise-free images.

2.6. CROPPING

The last preprocessing step is to crop our initial 200×200 pixels images. It helps reduce their size and increase the training speed of the neural network. The crop size also impacts the information in the image contributing to a lower accuracy of the network. However, when the right size is used, cropping helps neural networks ignore a large amount of useless information. In our case cropping was the most effective and customizable step. Through a trial and error approach, we trained crops of different sizes. The results show an increase in recall from $\approx 60 - 70\%$ to $\approx 80 - 90\%$.

The crop size we settled upon was 75×75 pixels. To obtain this value we started from the original size and went as low as a 20×20 pixels crop. A crop of 40×40 pixels was also found to provide good results but was dropped due to a limitation in the Inception network.

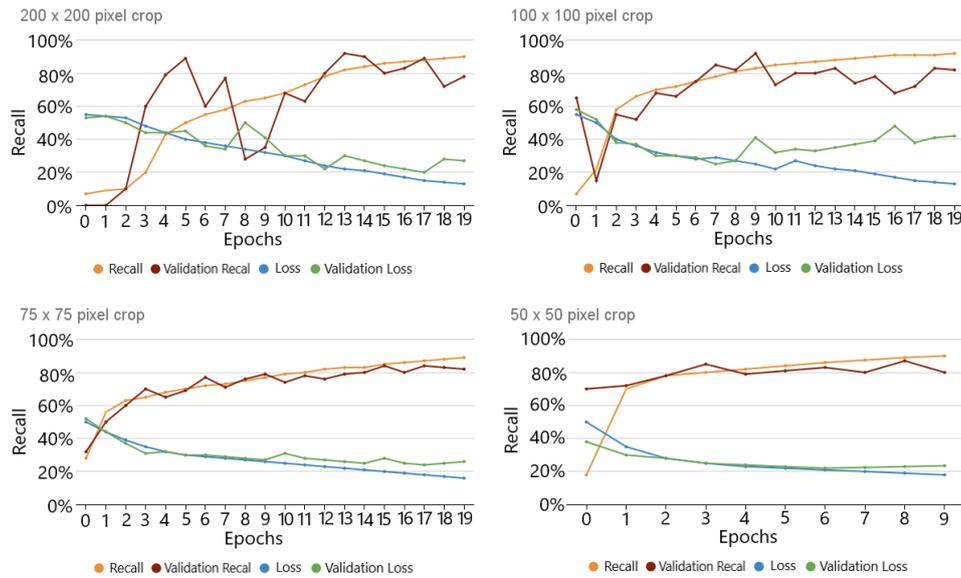


Fig. 6 – Training process and results on 200×200 (top left), 100×100 (top right), 75×75 (bottom left) and 50×50 (bottom right) pixels crops.

Figure 6 depicts the training results for various crop sizes on the Xception network. We are interested in the brown and orange bars representing the validation recall (*val_recall* label) respectively the training recall (*recall* label). From Eq. 2 we observe that the recall and the training recall values should be as close to one as possible. The plots show us how the training recall improves with every new training image (X-axis) with a validation limit of around 0.9.

However, in the top plots of Fig. 6 we can easily notice the high overfit value represented by the high and spiky values of the brown bar with respect to the orange one. We observe that as the crop size decreases (bottom plots), the overfit is less visible.

3. THE PROPOSED NEURAL NETWORK MODEL

In this section, we introduce the proposed ensemble neural network and discuss the CNNs which we chose as *weak* models.

3.1. ENSEMBLE MODEL

Given the challenging nature of our problem, *i.e.*, to identify potential asteroids based solely on the available pixel information, we chose an ensemble model. This technique is commonly used as a state-of-art approach for prediction and classification models as it enables the combination of several pre-trained “weak” models into a single one enhancing the performance of the final model. The individual models are not weak in the literal sense but we use this naming in order to maintain consistency with the domain-specific literature.

Another reason for choosing an ensemble model was to solve the overfitting problem. Combining different models can offer some advantages. The way they are combined and how they use the learned information from the weak models is the key to achieving high recall rates and accurate results.

Finally, using an ensemble model, makes the results much more reliable than using a single model. We observed varying recall rates between models. Values ranged from 82% on the Xception network to 88% on the ResNet. Concatenating them together with the Inception network not only reduced the overfitting value but raised the recall value on the whole validation dataset to around 91%.

3.2. WEAK MODELS

When it comes to achieving the best outcome for the proposed ensemble model we need to have concatenated the best weak models for our problem. This can be rather difficult as there are many choices based on the accuracy of various neural networks. We decided on using three widely used state-of-art models: ResNet-50, Xception, and Inception. They have either won a popular machine learning contest or are commonly used in binary classification problems such as our own.

- **ResNet-50** is a convolution neural network that has 50 layers. The name states the type of it as being a residual network. Its specific feature is that it uses skip

connections (shortcuts) to jump over some layers. The fact that it has so many layers is somehow the reason an approach that jumps over layers works so well in this context. Being such an innovative concept of a network, it won the 2015 ImageNet Large Scale Visual Recognition Challenge, which is one of the most renowned competitions in computer vision and classification contests.

- Google developed the **Inception** network and it has some interesting features to be tried against our problem. This is one of the best choices when it comes to classification problems. One of its issues is that it restricted us from using cropped images of 75×75 pixels. This architecture was also the runner-up at the competition where ResNet-50 was the winner.
- The **Xception** architecture, also developed by Google, is the successor of the Inception. While some argue that it is better than its predecessor, others remain in favor of the first model. Consequently, we considered an approach that used both of the models. Even though they are similar, they provide different results to be included in our current version of the ensemble model.

4. RESULTS

4.1. METRICS

In this paper we focus on the **recall** (cf. Eq. 2) to measure the performance of the proposed ensemble model for two reasons: (1) recall is a metric sensitive to the number of FNs which represent the missed asteroids; and (2) it is similar to the astronomical purity defined by Lieu *et al.* (2019) in the scenario where only solar system objects are considered.

4.2. SUMMARY OF RESULTS

The average recall across runs is 0.88 (with a range between 0.79 and 0.92) if we exclude from our experiments the asteroids missed by NEARBY and the recall decrease to 0.83 if we include them (with a range between 0.68 and 0.9). See Fig. 7 for the plotted results. In addition, there is a strong correlation ($R^2 = 0.98$) between the FNs our model produces across runs when excluding the missed NEARBY asteroids and when including them. Our classification model missed 312 asteroids from the 2,663 identified by NEARBY and 243 from the 544 missed by it.

4.3. ANALYSIS OF RESULTS

Our ensemble model was capable of identifying 301 asteroids from the 544 missed by NEARBY (55.3% success rate). Figure 8 shows the results per night of

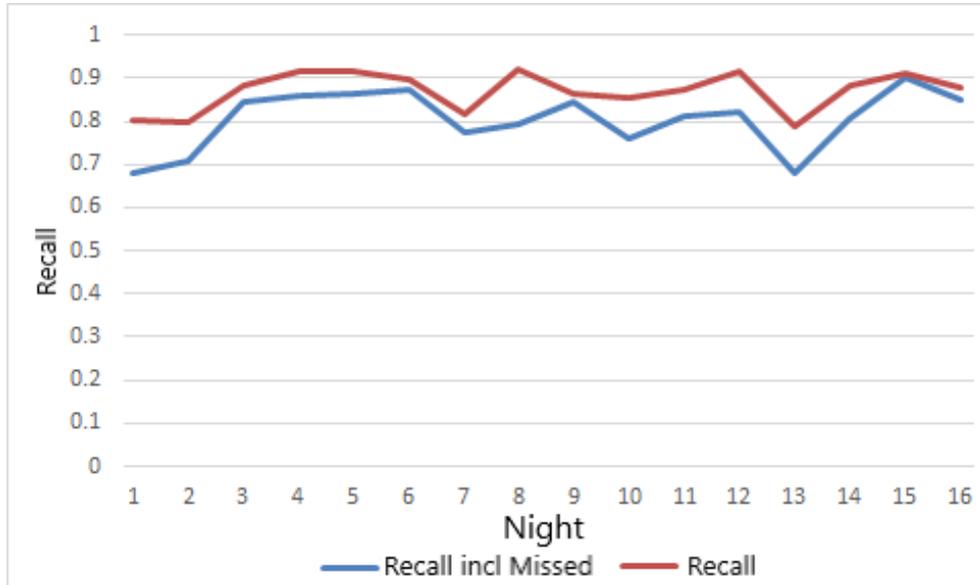


Fig. 7 – Recall of our classification results across runs when asteroids missed by NEARBY are included (blue) and excluded (red).

observation. The recovery of missed NEARBY asteroids had a success rate varying between 0 and 88%.

Furthermore, the model proved it can correctly identify images not containing asteroids with an FPR of 0.049 (cf. Eq. 3) producing 351 FPs (images without asteroids identified as containing one – which should in fact be TNs) and 6,816 TNs on the dataset excluding the missed asteroids by NEARBY.

Our predictions are binary, leading to either 0 (no asteroid in the image center) or 1 (asteroid present in the image center). Figures 9 and 10 show some examples of images containing correctly identified positives (asteroid in image) and negatives.

A careful analysis of the dataset on FP and FN images led to some interesting cases demonstrating the challenge of relying solely on image information and image labeling done by human or automated processes. Figure 11 shows some images which were labeled by NEARBY as false but contain an object in the center leading our model to classify them as having an asteroid. A number of challenging images containing either too much noise or multiple objects have been classified by our model correctly as seen in Fig. 12. These difficult cases also contain examples where no center object is visible despite the image being labeled as true (see Fig. 13). In this case, our model classified the image as false. Finally, in exceptional cases where an object did exist in the center it was wrongly classified by our model as false (see Fig. 13).

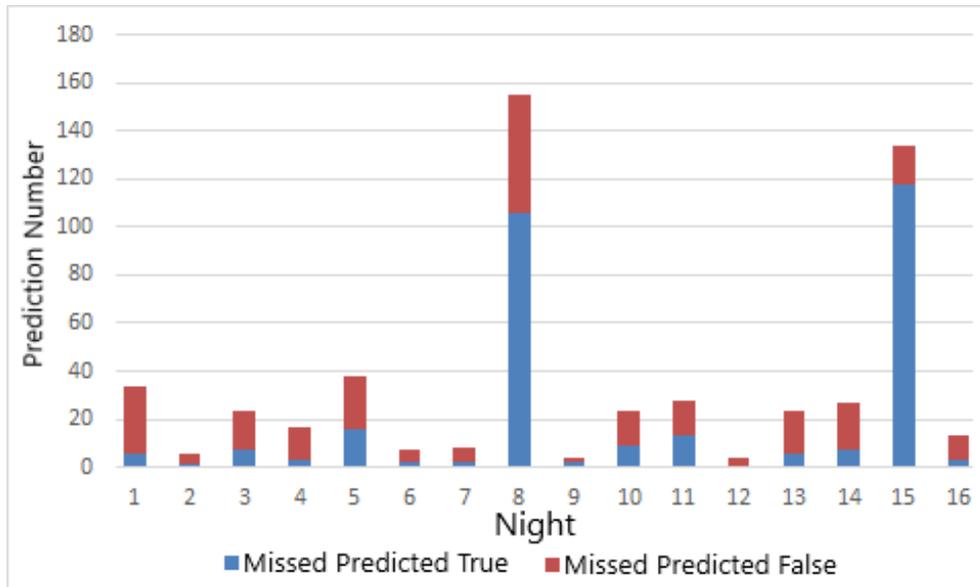


Fig. 8 – Classification outcome for our model across runs in the case of asteroids missed by NEARBY.

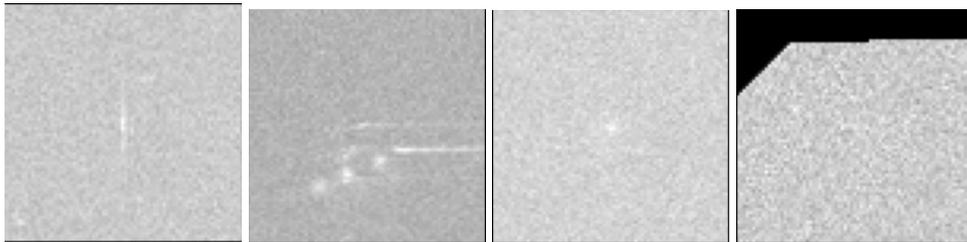


Fig. 9 – Correctly classified true negative examples.

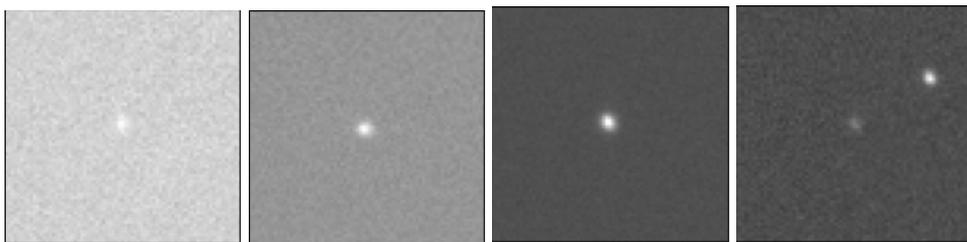


Fig. 10 – Correctly classified true positive examples.

Our ensemble model was however capable of identifying asteroids missed by NEARBY. Some examples can be seen in Fig. 14.

From these examples we can see the complexity of cases involving labels and

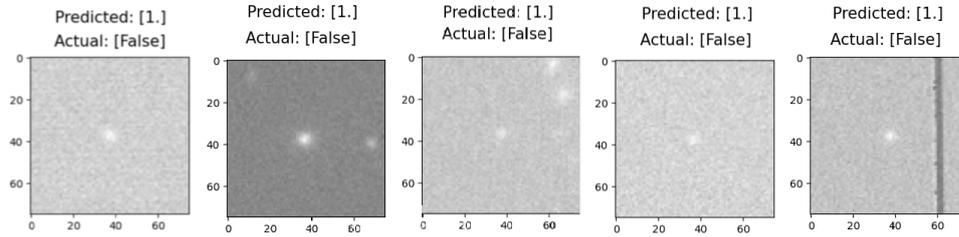


Fig. 11 – A set of images labeled as false but with an object in the middle of the image, leading our model to classify them as containing asteroids.

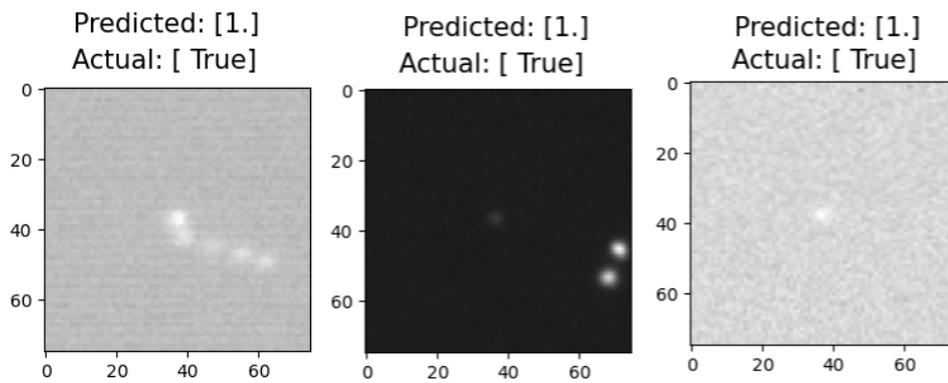


Fig. 12 – Exceptional images that were correctly classified as containing asteroids.

data, making it difficult for any ensemble model to learn all the particular cases. The reality is that we may have to work with either wrong identification or particular images for which there is no train data as their occurrence is rare or too specific.

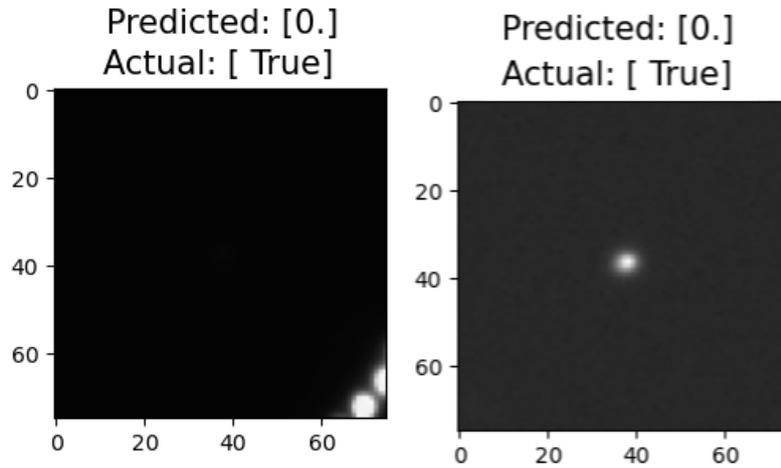


Fig. 13 – Two images incorrectly classified by our network. The first is labeled true but without an object visible in the middle leading our model to classify it as not containing an asteroid. The second one was labeled as true and had an object in the middle but our model classified it as not containing an asteroid.

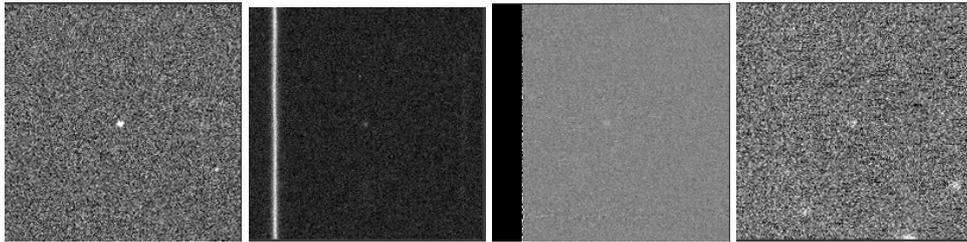


Fig. 14 – Examples of different images missed by NEARBY and identified by our ensemble model.

5. CONCLUSION

We have shown that our ensemble model made of state-of-the-art neural networks can provide a high recall rate and correctly classify difficult cases which contain asteroids. Its efficiency is impacted by particular use cases not known by the model or by *strangely* labeled data by the NEARBY platform and validated by human operators. As an improvement of the current NEARBY human validation method, in terms of time efficiency, our model has a recall of 0.88, missing 312 out of the 2,663 asteroids identified by NEARBY. However, the proposed model proves it can recover 55.3% of the asteroids missed by NEARBY. If we consider that NEARBY itself detects only 89% of the asteroids identified through the manual blink method in Astrometrica (Stănescu and Văduvescu, 2021) this increases the total detection rate of NEARBY to about 95%. Therefore the model can be used as a crosscheck for

the current NEARBY method by revisiting the images output as negatives (implicitly containing FNs) by NEARBY.

The current model relies solely on pixel information that the machine learning model can extract without performing additional operations such as computing the FWHM value of the potential target image. While it has proven efficient in detecting and recovering asteroids missed by the automated process of NEARBY, there is still room for improvement by introducing additional information (*i.e.*, FWHM) planned for future stages of the project.

Acknowledgements. This work was supported by a grant of the Romanian Ministry of Education and Research, CCCDI-UEFISCDI, project number PN-III-P2-2.1-PED-2019-0796, within PNCDI III.

REFERENCES

- Bertin, E.: 2006, Automatic Astrometric and Photometric Calibration with SCAMP. In: Gabriel, C., Arviset, C., Ponz, D., Enrique, S. (eds.) *Astronomical Data Analysis Software and Systems XV, Astronomical Society of the Pacific Conference Series* **351**, 112.
- Bertin, E., Arnouts, S.: 1996, SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement* **117**, 393–404. doi:10.1051/aas:1996164.
- Bertin, E.: 2010, *SWarp: Resampling and Co-adding FITS Images Together*.
- Granvik, M., Morbidelli, A., Jedicke, R., Bolin, B., Bottke, W.F., Beshore, E., Vokrouhlický, D., Delbò, M., Michel, P.: 2016, Super-catastrophic disruption of asteroids at small perihelion distances. *Nature* **530**(7590), 303–306. doi:10.1038/nature16934.
- Harris, A.W., D’Abramo, G.: 2015, The population of near-earth asteroids. *Icarus* **257**, 302–312. doi:https://doi.org/10.1016/j.icarus.2015.05.004.
- Lieu, M., Conversi, L., Altieri, B., Carry, B.: 2019, Detecting Solar system objects with convolutional neural networks. *Monthly Notices of the Royal Astronomical Society* **485**(4), 5831–5842.
- Lin, H.W., Chen, Y.T., Wang, J.H., Wang, S.Y., Yoshida, F., Ip, W.H., Miyazaki, S., Terai, T.: 2017, Machine-learning-based real-bogus system for the HSC-SSP moving object detection pipeline. *Publications of the Astronomical Society of Japan* **70**(SP1).
- NASA-JPL: 2022, *Discovery statistics*. (accessed May 27, 2021). <https://cneos.jpl.nasa.gov/stats/totals.html>.
- NEARBY: 2021, *Visual analysis of multidimensional astrophysics data for moving objects detection*. (accessed May 27, 2021). <http://cgis.utcluj.ro/nearby/>.
- Stefanut, T., Bacu, V., Nandra, C., Balasz, D., Gorgan, D., Vaduvescu, O.: 2018, Nearby platform: Algorithm for automated asteroids detection in astronomical images. In: *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 365–369. doi:10.1109/ICCP.2018.8516594.
- Stănescu, M., Văduvescu, O.: 2021, The umbrella software suite for automated asteroid detection. *Astronomy and Computing* **35**, 100453. doi:https://doi.org/10.1016/j.ascom.2021.100453.
- Tody, D.: 1986, The IRAF Data Reduction and Analysis System. In: Crawford, D.L. (ed.) *Instrumentation in astronomy VI, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **627**, 733. doi:10.1117/12.968154.
- Tricarico, P.: 2017, The near-earth asteroid population from two decades of observations. *Icarus* **284**,

416–423. doi:<https://doi.org/10.1016/j.icarus.2016.12.008>.

Vaduvescu, O., Curelaru, L.: 2006, *Euronear*. (accessed May 27, 2021). <http://www.euronear.org/>.

Waszczak, A., Prince, T.A., Laher, R., Masci, F., Bue, B., Rebbapragada, U., Barlow, T., Surace, J., Helou, G., Kulkarni, S.: 2017, Small near-earth asteroids in the palomar transient factory survey: A real-time streak-detection system. *Publications of the Astronomical Society of the Pacific* **129**(973), 034402. doi:[10.1088/1538-3873/129/973/034402](https://doi.org/10.1088/1538-3873/129/973/034402).

Received on 23 April 2021