Publications of the Astronomical Society of the Pacific, 136:124507 (20pp), 2024 December https://doi.org/10.1088/1538-3873/ad8c83

Leveraging Movement Representation from Contrastive Learning for Asteroid Detection

Noppachanin Kongsathitporn[1], Akara Supratak[1], Kanthanakorn Noysena[2], Supachai Awiphan[2], Danny Steeghs[3], Don Pollacco[3], Krzysztof Ulaczyk[3], Joseph Lyman[3], Kendall Ackley[3], David O'Neill[3], Amit Kumar[3], Duncan K. Galloway[4], Felipe Jiménez-Ibarra[4], Vik. S. Dhillon[5,6], Martin J. Dyer[5], Paul O'Brien[7], Gavin Ramsay[8], Enric Pallé[6], Rubin Kotak[9], Thomas L. Killestein[3,9], Laura K. Nuttall[10], and Rene P. Breton[11]

[1] Mahidol University 999 Phutthamonthon Sai 4 Rd, Tambon Salaya, Phutthamonthon, Nakhon Pathom, 73170, Thailand; akara.sup@mahidol.edu
[2] National Astronomical Research Institute of Thailand, 260 Moo 4, T. Donkaew, A. Maerim, Chiangmai, 50180, Thailand
[3] Department of Physics, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, UK
[4] School of Physics & Astronomy, Monash University, Clayton VIC 3800, Australia
[5] Astrophysics Research Cluster, School of Mathematical and Physical Sciences, University of Sheffield, Sheffield S3 7RH, UK
[6] Instituto de Astrofísica de Canarias, E-38205 La Laguna, Tenerife, Spain
[7] School of Physics Astronomy, University of Leicester, University Road, Leicester LE1 7RH, UK
[8] Armagh Observatory Planetarium, College Hill, Armagh, BT61 9DG, UK
[9] Department of Physics Astronomy, University of Turku, Vesilinnantie 10 , Turku, FI-20014, Finland
[10] School of Mathematics & Physics, University of Portsmouth, University House, Winston Churchill Ave, Portsmouth PO1 2UP, UK
[11] Department of Physics and Astronomy, The University of Manchester, Oxford Rd, Manchester M13 9PL, UK
Received 2024 July 31; accepted 2024 October 29; published 2024 December 27

Abstract

To support asteroid-related studies, current motion detectors are utilized to select moving object candidates based on their visualizations and movements in sequences of sky exposures. However, the existing detectors encounter the manual parameter settings which require experts to assign proper parameters. Moreover, although the deep learning approach could automate the detection process, these approaches still require synthetic images and hand-engineered features to improve their performance. In this work, we propose an end-to-end deep learning model consisting of two branches. The first branch is trained with contrastive learning to extract a contrastive feature from sequences of sky exposures. This learning method encourages the model to capture a lower-dimensional representation, ensuring that sequences with moving sources (i.e., potential asteroids) are distinct from those without moving sources. The second branch is designed to learn additional features from the sky exposure sequences, which are then concatenated into the movement features before being processed by subsequent layers for the detection of asteroid candidates. We evaluate our model on sufficiently long-duration sequences and perform a comparative study with detection software. Additionally, we demonstrate the use of our model to suggest potential asteroids using photometry filtering. The proposed model outperforms the baseline model for asteroid streak detection by +7.70% of f1-score. Moreover, our study shows promising performance for long-duration sequences and improvement after adding the contrastive feature. Additionally, we demonstrate the uses of our model with the filtering to detect potential asteroids in wide-field detection using the long-duration sequences. Our model could complement the software as it suggests additional asteroids to its detection result.

Unified Astronomy Thesaurus concepts: Asteroids (72); Computational astronomy (293); Convolutional neural networks (1938); Astronomy data analysis (1858)

1. Introduction

To facilitate planetary defense and research in astronomy, asteroid detection is an important part as it provides information on known targets and predicts their trajectories (Parfeni et al. 2020; Cowan et al. 2022; Wang et al. 2022). In comparison to other static sky objects, an asteroid is a moving object that can be detected by stacking exposures over a period of time to observe its positional shifting. The majority of asteroids in our solar system are located in a region between the orbits of Mars and Jupiter.[12] Although these asteroids have less chance of impact with the Earth compared to the former, the detection of known asteroids or potential objects supports the studies of early-stage solar system and future impact

[12] https://science.nasa.gov/solar-system/asteroids/

estimation. To obtain data for this research, there are sky surveys such as the Near-Earth Asteroid Tracking (Pravdo et al. 1999), and the Gravitational-wave Optical Transient Observer (GOTO; Dyer et al. 2022, 2024a; Steeghs et al. 2022) that provide rich data for astronomical object detections. Furthermore, the upcoming surveys such as Vera C. Rubin Observatory Legacy Survey of Space and Time (Tyson 2002) will conduct a growing catalog of astronomical objects in the future. To indicate the presence of astronomical objects, the brightness profile and locations of the objects are observed over a sequence. However, the subsequent processes, including background subtraction and movement detection, to identify a small number of moving candidates are based their performance on manual parameter settings. Although the modern approaches using deep learning (Zhai et al. 2014; Wang et al. 2022) could replace the manual settings, they still require the additional images generated from real images, synthetic images, to balance the object classes and improve the data variation for model training. As a result, even the existing approaches could perform the automatic detection using artificial image synthesis and image difference techniques, these approaches not only delay the overall detection process but also require manual parameter settings to generate high-quality data.

Traditional asteroid detection begins with identifying light sources in each specific exposure to distinguish candidate objects from noise and background sources (Copandean et al. 2018; Stănescu & Văduvescu 2020). The brightness and position of the objects are extracted using various tools, both manually and automatically, in a step called source extraction. Static objects are then removed using star and planet catalogs such as those by Gaia Collaboration et al. (2018) and Rudenko (2016). After this screening, the remaining candidates are observed by stacking consecutive exposures. Movement can be represented by positional shifting and visual inspection, with several approaches to detect movement such as proper velocity and difference imaging.

The difference image is a popular approach that dominates in visual inspection for motion representation because static objects can be removed by subtracting the first exposure (template image) from the following exposures (science images). Alternatively, a moving object can be detected as a trail in a stack of exposures taken over a sufficient period of time. This stacking technique (Wang et al. 2022) leaves static objects as point sources. Although these representations help distinguish moving objects, they still require manual parameter settings and extended exposures to perform moving object detections. Moreover, as data preparation is repetitive and labor-intensive, it creates a bottleneck in the overall detection process.

To reduce the bottleneck, machine learning models are utilized to learn the motion representation. Convolutional neural networks (CNNs; LeCun & Bengio 1998) have emerged to effectively extract representations of input features and refine their parameters through the learning process. These networks not only demonstrate promising performance in image classification but also show generality in learning and adapting to various fields. CNNs have been implemented to detect transient sources, supernovae, and asteroids, showcasing their adaptability in handling and learning to extract representations from brightness features (Gieseke et al. 2017; Reyes et al. 2018; Moller & de Boissière 2020; Carrasco-Davis et al. 2021).

Positional shifting of a brightness blob is the key difference of asteroids. CNNs have been utilized to detect asteroids using a streak in image stacking (Cowan et al. 2022; Wang et al. 2022) and brightness value after image subtraction (Carrasco-Davis et al. 2021) as an input feature. However, these approaches encounter the numbers of exposures taken for generating an asteroid streak and manual environmental parameter extraction for image subtraction.

To automate the motion extraction which is robust to the brightness variation, we introduce contrastive learning to extract the motion representation for a classification model. Contrastive learning is a method that generates contrasting representations between inputs, considered as representation learning. This method includes a pairing process that learn the contrast of brightness by pairing a single input to many samples with variation of brightness. This method has demonstrated its capability to improve visual-based tasks in both unsupervised and supervised contexts (Koch 2015; Chen et al. 2020; Khosla et al. 2020; Lin et al. 2022). The output of this method is a representation that reinforces the dissimilarity between inputs. Analogous to the difference image in traditional approaches, this contrastive representation can enhance movement representation from other CNNs for movement detection.

In this work, we propose a movement detection model framed as an image classification task to determine whether a pair of exposures contains a moving object. The model is an end-to-end architecture forsuggesting candidates. We trained and evaluated the model on long-duration exposure pairs, 15–60 minutes, generated from short-duration exposures, around a minute. It includes two branches to generate a contrast representation and learn high-level representation. First, the contrastive branch is pretrained on similarity classification. Second, the classification branch is trained to generate a high-level representation and combine it with the contrast representation to produce the movement representation for an asteroid classifier. During the latter training phase, the parameters of the contrastive branch are fixed. We use the ResNet backbone for the contrastive branch and the backbone proposed by the optical transient detection model (Wardęga et al. 2021) for the classification branch. Additionally, we adopt an equal sampling method on the classification branch to address the imbalance without using synthetic data. Additionally, we perform the photometry post processing to suggest potential asteroids from candidates detected by our model.

Finally, we integrate our proposed model with source selection methods, including sliding windows and the `SExtractor` (Koch 2015), to demonstrate its detection capabilities compared to existing asteroid detection software. We uses the data from GOTO telescope (Dyer et al. 2024b) to demonstrate the proposed model's performance and conduct the comparative study. Our contributions are listed below.

1. We propose the end-to-end classification model trained solely on real exposures under mild degree of imbalance.
2. We introduce a contrastive feature extractor for movement representation atop the end-to-end classification, emphasizing the contrast between two exposures in a sequence.
3. Our proposed model complements traditional approaches by suggesting additional asteroid candidates.

The source code are released at https://github.com/Noppachanin/asteroid-det-contrastml.

2. Literature Review

2.1. Astronomical Object Detection

Traditional astronomical object detection employs algorithmic methods based on statistical analysis of single images or sets of images. Astronomical objects exhibit unique brightness profiles, and image processing techniques such as filtering and statistical estimation are used to extract them from noisy backgrounds. The general detection pipeline includes input preparation and detection steps. Noise removal or background subtraction typically begins with bad pixel masking to reduce their impact. Individual subtraction may involve mean filtering, applied either globally (Stănescu & Văduvescu 2020) or locally (Bertin & Arnouts 1996; Savanevych et al. 2022). For sets of images, a median filter serves as a cross-image estimator to estimate background levels and mask static objects. However, existing estimation algorithms (Bijaoui 1980; Almoznino et al. 1993; Bertin & Arnouts 1996) need careful consideration regarding bias, robustness to noise, and computational resources. Noise-reduced images are then processed for object detection using photometric statistics and coordinate matching. `SExtractor` (`SExtractor`) (Bertin & Arnouts 1996) employs peak-finding and thresholding techniques to extract light sources. For faint targets, the extractor utilizes the Point Spread Function (PSF) to localize the amplitude of point sources.

2.2. Asteroid Detection Tools

As asteroid detection requires multiple observations, individual detection results are combined into a sequence detection. Existing software tools such as `Umbrella` (Stănescu & Văduvescu 2020), `CoLiTec` (Khlamov et al. 2016), and `Astrometrica` (Raaba 2002) facilitate this process by providing object pairing and identity to merge detection results across images in a sequence and identify. Before feeding data into the detection process, fixed stars are typically removed using star-planet catalogs (Tedesco 1994; Gaia Collaboration et al. 2018). Asteroids observed in a sequence can appear as blob-like objects or long trails, depending on exposure time (Stănescu & Văduvescu 2020). Therefore, various pairing algorithms are used to detect movement, such as tree-based motion estimation (Stănescu & Văduvescu 2020), complex decision rules (Khlamov et al. 2016), and shifting of the point-spread function (Raaba 2002). Studies on near-Earth asteroid detection utilize source extraction modules with different matching designs, including the classic blink algorithm (Copandean et al. 2018), proper velocity estimation (Copandean et al. 2017), and shift-and-stack methods (Golovich et al. 2021).

Comparative analyses between detection software, Khlamov et al. (2016) have shown that `CoLiTec` provides 253 more measurements than `Astrometrica`, while `Astrometrica` exhibits a 30%–50% greater positional standard deviation in low signal-to-noise ratio conditions. These tools are designed not only based on observable sources but also require predefined parameters for each step in image preprocessing. Parameters must be carefully defined, as comparative studies indicate their impact on the efficient detection range of signal-to-noise ratio for asteroids. To address these challenges, there are learnable models such as neural networks that can learn patterns from input sequences and extract moving objects based on their brightness profiles.

2.3. Convolutional Neural Networks (CNNs)

LeCun & Bengio (1998) have demonstrated their potential for handling inputs such as images, speech, and time-series data. CNNs utilize convolutional filters, known as kernels, to recognize patterns in inputs and represent them as high-level feature maps containing important information. These filters are connected to learnable weights that are adjusted during training. Image classification involves categorizing an input image based on its visual representation. CNN models such as ResNet (He et al. 2016), VGG (Simonyan & A. Zisserman 2014), MobileNet (Sandler et al. 2018), and Image Transformer (Dosovitskiy et al. 2020) have achieved high accuracy in image recognition benchmarks (Krizhevsky 2009; Russakovsky et al. 2014). Given the success of CNNs, there have been implementations of CNN models in astronomical object detection and classification.

2.4. CNN for Asteroid Detection

Recently, CNNs have been utilized for various astronomical object detection tasks, including real-bogus classifications, supernovae detection, and source detection as multiclass classifications. Real-bogus classifications (Wright et al. 2015; Cabrera-Vives et al. 2017; Gieseke et al. 2017; Wright et al. 2017; Ackley et al. 2019; Mahabal et al. 2019; Mong et al. 2020;

Wardęga et al. 2021; Acero-Cuellar et al. 2022; Takahashi et al. 2022; Dyer et al. 2024b) exemplify the potential of CNNs to categorize images and reduce labor-intensive processes. This task involves classifying an object as either a real source or a defect. The model design typically takes a pair of images and their difference; the latter is a subtraction image of the input images to represent brightness variation over time, emphasizing the continuous brightness of real sources. The subtraction of images is also implemented in supernovae (Reyes et al. 2018; Moller & de Boissière 2020) classification as well because the astronomical object has its unique brightness profile. For multiclass classifications, metadata or statistics from an image are combined with the high-level features from CNN models to increase the variety of predicted classes (Morice-Atkinson et al. 2018; Carrasco-Davis et al. 2021; Aldahoul et al. 2023). Additionally, the interpretation of learning models indicates that these additional features can be effectively used for multiclass source classification (Morice-Atkinson et al. 2018). These implementations demonstrate the feasibility of using CNN models for asteroid detection, as asteroids are among the objects detected by source detection models and can be represented by positional shifting through image subtraction.

2.5. Feature Engineering for Asteroid Detection

Asteroid detection can be approached as either an object detection or image classification task. For object detection, the model takes a wide field of view that includes many light sources and uses image stacking to identify moving objects (Zhai et al. 2014; Cowan et al. 2022). This process requires a large number of images to reveal the trajectories of moving objects. Therefore, the image classification approach overcomes the requirement by using a pair of images and the feature map extracted from the pair (Copandean et al. 2018; Parfeni et al. 2020; Carruba et al. 2022; Rabeendran & Denneau 2021; Cowan et al. 2022; Wang et al. 2022). Difference image is one of the solutions to reduce the need of stacking images as the difference highlights positional shifting through brightness subtraction (Copandean et al. 2018; Rabeendran & Denneau 2021; Wang et al. 2022). There are other approaches using dynamic mapping (Carruba et al. 2022) or a trail/streak (Rabeendran & Denneau 2021; Wang et al. 2022) which show the movement of light sources within an exposure time. However, The classification approach encounters an imbalance problem as the number of real asteroids in a field of view is infinitesimal compared to static sources. The previous classification works to handle this problem by simulating additional images using a statistic of real asteroids (Zhai et al. 2014). Though the simulated images provide reliable results, simulation and also image difference algorithms need the predefined statistic from a fixed set of images. To handle the imbalance and extract the difference between an image pair, contrastive learning is introduced to learn contrast between a pair of input.

2.6. Contrastive Learning

Contrastive learning is designed to compare the similarity between a pair of inputs and classify whether the pair belongs to the same class. This learning method is used in semi-supervised learning by implementing data augmentations on a labeled data set and using the augmented versions to learn a contrastive mapping (Koch 2015; Chen et al. 2020; Tian et al. 2020; Lin et al. 2022). SimCLR outperforms previous self-supervised and semi-supervised learning methods on ImageNet in terms of prediction accuracy and the need for a labeled data set, as it uses augmented images (Chen et al. 2020). The potential of contrastive learning enhances image classification by reducing the necessity for labeled data and producing contrastive features that represent the differences between pairs of inputs. Therefore, we introduce this approach to learn contrastive features and match a single asteroid sample to many non-asteroid samples. This method not only overcomes the imbalance through the matching process but also produces a mapping that semantically represents the differences between images.

3. Data Source

To begin with, we utilize sky images taken by the GOTO during the first seven months (January–July) of 2020. Since GOTO is designed to perform all-sky surveys and detect gravitational-wave events, we start by exploring and understanding the characteristics and statistics of the sky images from GOTO before using this data for further experiments.

The GOTO is a wide-field optical transient telescope designed for rapid observation in search of gravitational-wave events (Dyer et al. 2024b). GOTO uses a CCD sensor. The pixel scale is converted through optical assemblies at a scale of $1\farcs25$ per pixel. The prototype phase of GOTO included four unit telescopes (UT), which later developed into eight unit telescopes. During our work, we use the eight UT system (GOTO-8). We separate the sky images from the prototype telescopes and the latest four telescopes as their mappings are not identical. GOTO provides both triggered and sky-survey modes to serve distinct purposes.

In triggered mode, GOTO responds to gravitational-wave signals and covers thousands of square degrees to capture short-lived sources. In sky-survey mode, sky images are taken continuously to serve as recent references for upcoming gravitational-wave detections. This process provides images for detecting other astronomical objects, such as transients and variable stars. The survey grid consists of tiles with $3\fdg7$ right ascension and $4\fdg9$ declination. The combination of eight unit telescopes covers around 40 square degrees, with each unit producing images of 8176×6132 pixels. This setup generates tiles that cover the entire sky, contributing to the exploration of

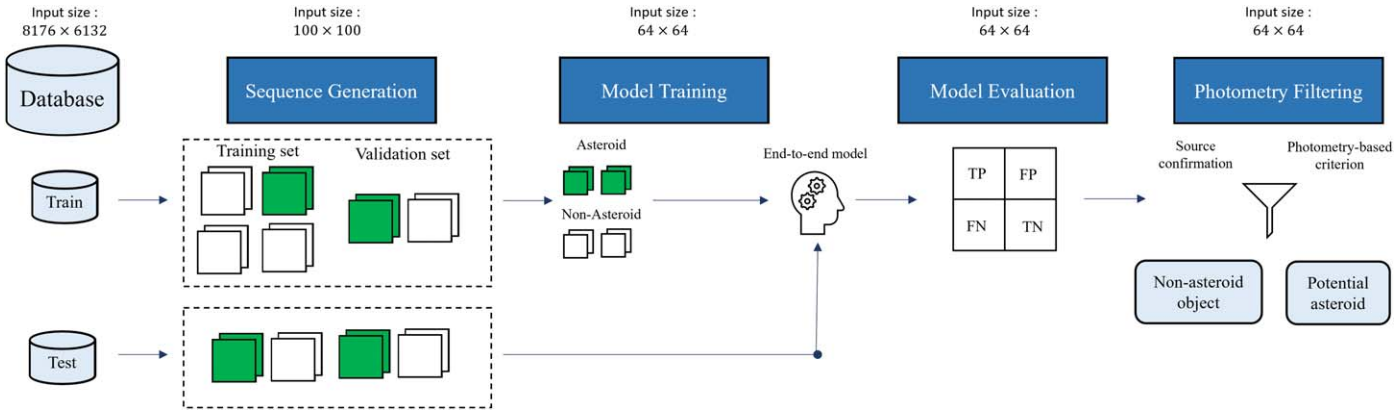


Figure 1. The exposures from NARIT's database are primarily divided into training set and test set. Then, the training set are subdivided into two parts with the ratio of 80:20 for model training and validation, respectively. The exposure sequences are generated for both data sets using time duration criteria and positional matching. The exposures are classified into asteroid and non-asteroid sequences by `SkyBoT`. The end-to-end model is trained by training, and the validation set is used for model selection. The best end-to-end model is evaluated using the sequences from the test set to demonstrate the model performance for asteroid detection.

astronomical objects. Additionally, GOTO provides image data for detecting other astronomical objects, including variable stars, transients, and asteroids. The images are stored as exposures of 2–3 sub-images, each covering ∼40 square degrees and lasting 30–90 s. These sky image exposures are stored in Flexible Image Transport System (FITS) files. The "*.fits*" format includes multiple extensions, delivering arrays of sub-images and their metadata (e.g., camera orientation, timestamp) in a single file. This format facilitates our data preparation process by providing both images and positional conversion information.

Prior to object detection, new data files undergo standard CCD bias, dark, and flat-field corrections using calibration frames taken over multiple nights. The corrected images are then subjected to preliminary instrumental photometry to produce an initial astrometric solution for image quality assessment. A package is used to clean image distortion and transform the image coordinates.[13] After data cleaning, the final refitting process uses the reference catalog, ATLAS-REFCAT2 to compute whether the astrometric solutions exceed expected values, ensuring image quality. Therefore, the pipeline produces a set of 3–4 exposures, each 30–90 s long, with a typical calibration uncertainty of 0.03 mag measured by Kron Apertures (Kron 1980). These procedures are accomplished by the GOTO pipeline, and the outputs are stored in NARIT's database.

The sky image has a size of 8176×6132 square pixels. The exposure features represent the brightness in each pixel, where the feature value and actual brightness are linearly related. The feature value is a 16 bit integer ranging from 0 to 60,000 ADU pixel^{-1}. The acceptable range of the values is demonstrated by existing devices such as the FLI camera showing linearity within 70 to 62,000 ADU pixel^{-1} (Duriscoe et al. 2007).

[13] https://github.com/GOTO-OBS/goto-astromtools

4. End-to-end Model for Asteroid Detection

We propose an end-to-end movement classification model incorporating a contrastive embedding inspired by the Siamese network (Koch 2015). The model consists of "contrastive" and "inductive" branches. The former is pretrained to represent the contrast between exposures, contrastive feature, and the latter branch is induced by the classification goal to generate motion-related feature. The outputs from these branches are combined to represent movement and converted into probabilities using a fully-connected layer to classify a pair of exposures. Formally, suppose is an input sequence $x^{(i)}$ with its label $y^{(i)}$ at epoch i where $x \in \mathbb{R}^{2\times 64\times 64}$ and $y \in \{0, 1\}$. An input sequence $x^{(i)}$ is sampled from s_{ij}, shown in Equation (5). The overview of our pipeline is shown in Figure 1.

The contrastive embedding is responsible for generating the difference representation of exposures in a sequence. This branch is pretrained on similarity classification to learn the discrimination between asteroid and non-asteroid sequences. Since sequences containing an asteroid exhibit significant movement compared to non-asteroid sequences, the embedding vectors are expected to map both classes to different locations in the embedding space. We follow the model training approach in Kongsathitporn et al. (2023), but replace the backbone with a modified ResNet-18, Figure 4(b), where we reduce the number of layers and input dimensions. Our contrastive extractor is pretrained, and the model weights are fixed to preserve its role in generating the contrastive embedding.

$$h_c^{(i)} = \mathrm{Cont}_{\theta_c}(x^{(i)}), \tag{1}$$

where Cont represents a function generating a contrastive feature parameterized by θ_c. The weights θ_c is pretrained

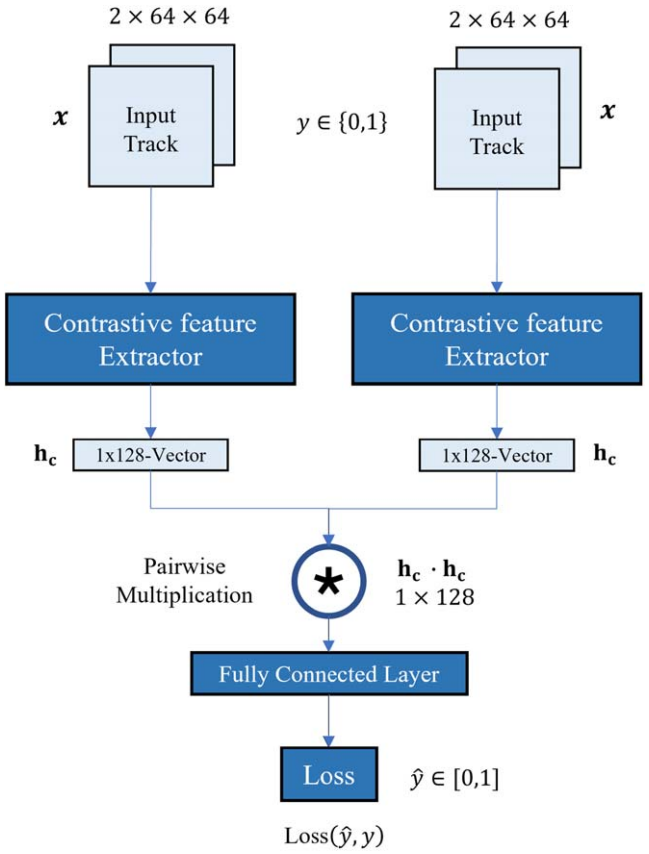


Figure 2. The contrastive training takes two input sequences where the label y is 1 if the sequences are from the same classes, and 0 for different classes. In this figure, we show the output size of ResNet backbone. The output sizes of each extractor is 1×128. Then, the output vectors are fed to pairwise multiplication to represent the similarity of the sequences. The output of the model is the probability to be asteroid represented as $\hat{y} \in [0, 1]$. For VGG backbone, the output size of the extractor is 1×2048.

beforehand, and fixed in end-to-end training for a classification. The embedding vector $h_c^{(i)}$ represent the contrast between exposures $\{e_i. e_j\}$ where $h_c^{(i)}$ has a dimension of 1×128. The contrastive training is shown in Figure 2.

The end-to-end model takes an input sequence to predict whether the sequence includes a moving object. We utilize class-balanced batching to handle the imbalance during the training process. The output of the inductive branch is concatenated with the contrastive branch to represent the movement representation. Figure 4 shows the ResNet backbone for inductive branch. Our end-to-end learning, Figure 3, is performed as follows:

$$h_n^{(i)} = \mathrm{Ind}_{\theta_n}(x^{(i)}), \tag{2}$$

$$h_m^{(i)} = h_n^{(i)} || h_c^{(i)}, \tag{3}$$

$$a_m^{(i)} = \mathrm{FC}(h_m^{(i)}), \tag{4}$$

where Ind represents a function generating a high-level mapping from inductive branch parameterized by θ_n; $h_n^{(i)}$ is high-level features which has similar dimension to $h_c^{(i)}$; $h_m^{(i)}$ is a concatenated vector representing a movement $h_n^{(i)} || h_c^{(i)}$ where $h_m^{(i)}$ has a dimension of 2×128. FC represents a function of fully-connected layer mapping a vector $h_m^{(i)}$ into a 1D-vector $a_m^{(i)}$. The function FC convert the 2×128 vector into a probability $\hat{y} \in [0, 1]$.

5. Experimental Setup

The overview of experimental setup is shown in Figure 4. This setup is applied to our experiments in Section 6.2. For Section 6.5, we modified some parts in the original setup to conduct the experiments on full-sized exposures.

5.1. Sequence Generation

The 8176×6132-sized exposures taken by the GOTO telescope from 2020 January to July were used for model training and evaluation. Initially, we generated a new set of sequences from 2020 January exposures for training and performance validation. These sequences were split in an 80:20 ratio for training and validation, respectively. Subsequently, sequences from February to 2020 July were designated as the test set to evaluate the model's performance. We define sequences containing an asteroid as positive samples, while those containing only static sources were considered negative samples. The asteroid positions in a sequence are retrieved by `SkyBoT`. To ensure that asteroids were visible, we restricted the brightness magnitude to 19 or lower, where lower magnitudes indicate brighter objects. Furthermore, we enhanced the visibility of positive samples using `SExtractor` during sequence generation (Bertin & Arnouts 1996).

We initially generate sequences consisting of two exposures: a template exposure and a science exposure. To ensure sufficient time difference between exposures for detection purposes, we limit the duration of exposures to between 15 and 60 minutes. We use the `SkyBoT` Python package to collect all asteroid names in each exposure, identifying astronomical objects within the observation area (Berthier et al. 2006). For each sequence, exposures are arranged chronologically, and patches are cropped with asteroid positions as the center. We utilize asteroid positions from the template exposure to extract patches from both exposures. However, due to incomplete overlap between exposure pairs, we select only asteroid positions located within the overlapped area. To enhance the diversity of positive samples, we store the training data set in a 100×100 format, allowing for random cropping and rotation to vary the asteroid's position and movement direction within an input sequence. After applying these random augmentations, an input sequence to our model is resized to 64×64. The sequence generation process is illustrated in Figure 5.

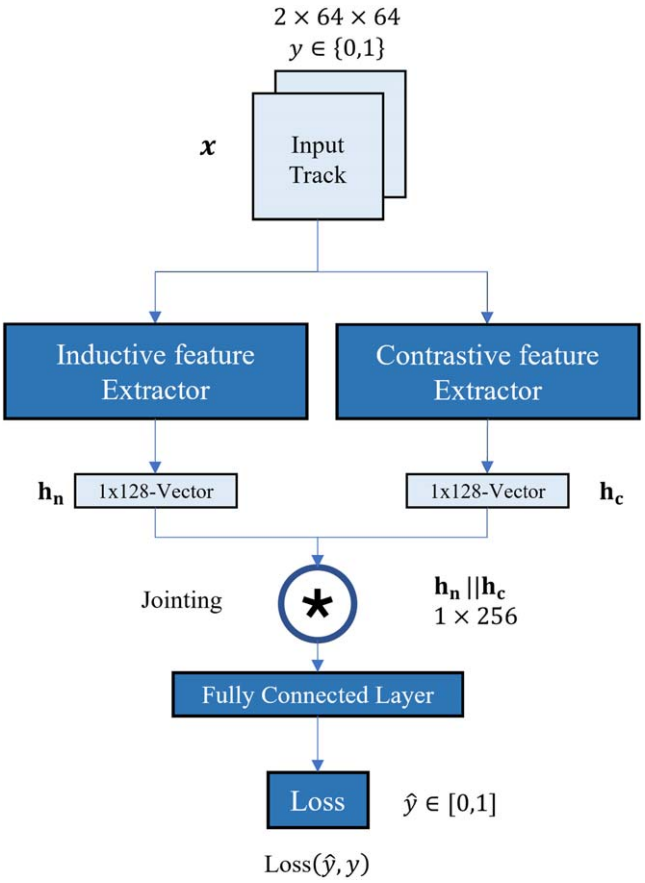


Figure 3. The end-to-end training takes a sequence of two exposures where the label y is 1 ,if the sequence contains an asteroid confirmed by *SkyBot*, and 0 for a non-asteroid target. In this figure, we show the output size of ResNet backbone. The output sizes of each extractor is 1×128. For inductive branch, the output vector size is 1×128. Therefore, the output vectors are joined to represent the 1×256 motion representation. The output of the model is the probability to be asteroid represented as $\hat{y} \in [0, 1]$. For VGG backbone, the output size of the extractor is 1×2048, so the size of the joined vector is 1×2176.

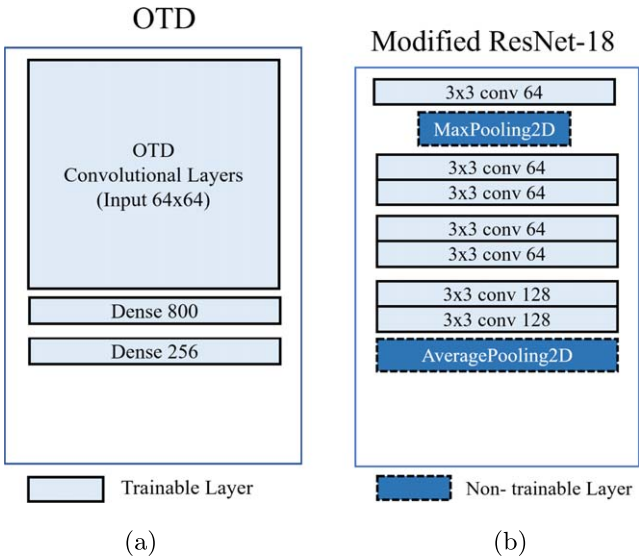


Figure 4. Backbone Architectures; (Left) Optical transient detection (OTD) backbone (OTD) (Wardęga et al. 2021) architecture for a 64×64 exposure; (Right) Modified Resnet-18 (a) Contrastive Embedding (b) Inductive Extractor.

Formally, suppose there are N_e exposures $e_1, ..., e_{N_e}$ and t_i denotes time that an exposure i was taken. The center of 8176×6132-sized exposures at time t_i is denoted by (x_i, y_i, t_i) where the position (x_i, y_i) is mapped from `RA` and `DEC` using `WCS` (World Coordinate System). Our exposure matching process is performed as follows:

$$s_{ij} = \{e_i, e_j\}, \tag{5}$$

where s_{ij} is a sequence of exposures; the exposure e_i is taken before e_j; we limits the time difference between t_i and t_j in unit of minutes to be within 15–60 minutes.

Then, template exposure e_i is projected on the science exposure e_j, Figure 5, to obtain the overlapped area a_{ij} for a sequence s_{ij} where the centers are denoted by (x_k, y_k, t_i) and (x_k, y_k, t_j).

Let V_a and V_{na} denotes asteroid and non-asteroid patches. Each patch is represented by its center (x, y). We generate mildly imbalance data set by cropping approximately 20 negative patches after each positive patches is obtained. Therefore, for each sequence, we collect N_a asteroid patches and $N_n \approx 20N_a$ non-asteroid patches. Data from the overlapped area a_{ij} of a sequence s_{ij} is obtained as follows:

$$V_a, V_{na} = \mathrm{Ext}(a_{ij}), \tag{6}$$

$$V_a = \{ (x_a, y_a) | a \in \mathbb{Z}, 1 \leqslant a \leqslant N_a \}, \tag{7}$$

$$V_n a = \{ (x_n, y_n) | n \in \mathbb{Z}, 1 \leqslant n \leqslant N_n \}, \tag{8}$$

where $\mathbb{Z}$ denotes a set of integer; $\mathrm{Ext}(a_ij)$ is a patch extraction function; N_a and N_n denotes the numbers of asteroid and non-asteroid patches, respectively.

The extraction function starts from the asteroid retrieval obtained from `SkyBoT`. queries define the centers of positive patches. The asteroid patches (x_n, y_n) are cropped using the retrieved positions with size of 100×100. Positions located near the boundary of a_{ij} is padded by root-mean-squared background noise. we filters out asteroid patches which `SExtractor` found no source to ensure visibility of an asteroid at 3σ from their background level. Negative patches (x_n, y_n) are randomly cropped in a_{ij} where there is no asteroid in the patches. As a result, the numbers of positive and negative patches are shown in Table 1.

5.2. Model Training

We performed experiments by varying the input type and model architecture. There are two exposure types: raw exposures from the database and difference images. For the

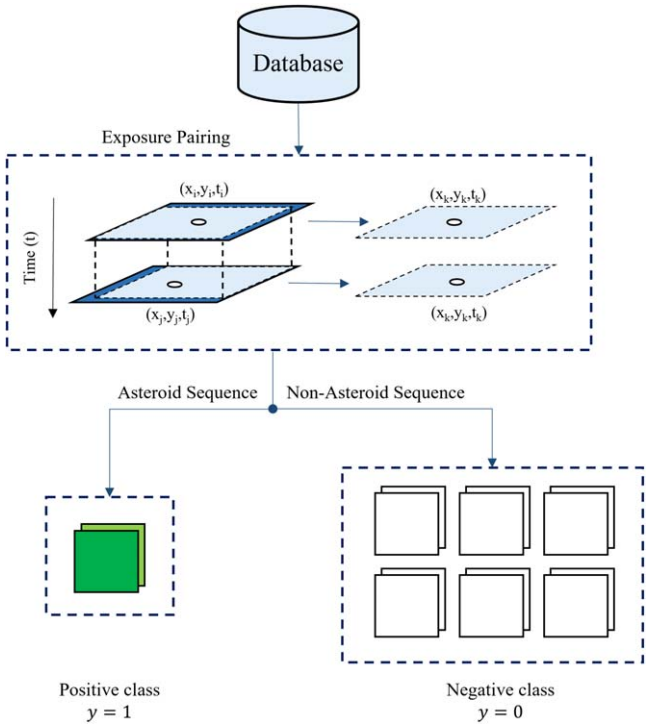


Figure 5. The exposure sequences are generated from the exposure pairs which are taken at nearly similar position, and their time difference is within 15–60 minutes. The exposure pairs are matched to determine asteroid searching area. The asteroid sequences, positive samples, are first generated from the asteroid positions queried by `SkyBoT` in the searching area. The non-asteroid sequences, negative samples, are generated after each asteroid sequence to maintain the level of imbalance as shown in Table 1.

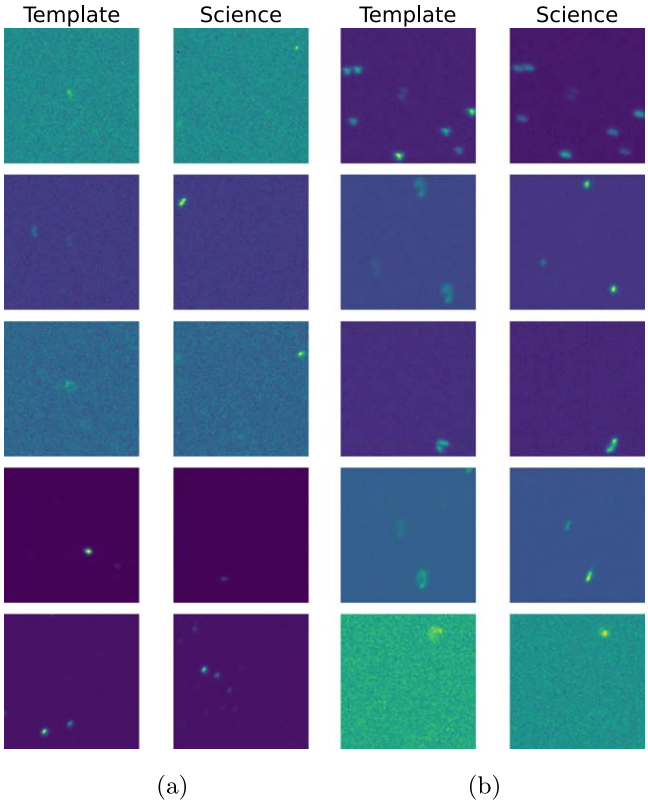


Figure 6. Visualizations show examples of asteroid and non-asteroid sequences in our data set. The asteroid sequences are collected from the overlapped area shown in Figure 5, and the asteroid positions from `SkyBoT` are the center of cropped patches. The non-asteroid sequences are collected by random cropping within the overlapped area where the asteroids are not nearly located (a) Asteroid images (b) Non-asteroid images.

Table 1
Numbers of Positive and Negative Samples in Our Data Set

Dataset	Asteroid Sequences	Non-asteroid Sequences
Training	1703	35912
Validation	341	7183
Test	1808	37819

latter, we used a Python package named OIS[14] to perform image subtraction using bramich method (Bramich 2008). We defined three types of sequences for model training: difference images only, raw exposures only, and raw exposures with difference images. The dimensions of these inputs are $1 \times 64 \times 64$, $2 \times 64 \times 64$, and $3 \times 64 \times 64$, respectively. The examples of asteriod and non-asteroid sequences are shown in Figure 6.

We pretrained the parameters of the contrastive branch $Cont_{\theta_c}$ on similarity learning using VGG and ResNet as the backbone architectures. The training guidelines are provided in the previous paper (Kongsathitporn et al. 2023). Next, we performed experiments on movement representation. We compared the classification performance using only *inductive* extractors, the feature extractor induced by the the goal task, and the combined features between inductive and contrastive features. We selected the optical transient detection (OTD) backbone (Wardęga et al. 2021), VGG, and ResNet for the inductive extractor Ind_{θ_e}. Moreover, the parameters of the contrastive branch θ_c were frozen during the end-to-end training. Therefore, there are five models: end-to-end model with raw features (RawN2N), end-to-end model with contrastive and raw features (CT-N2N), end-to-end model with different and raw features (ADN2N), and end-to-end model with only different features (ODN2N). Lastly, we utilized the architecture for asteroid's streak detection (SDCNN) proposed by Wang et al. (2022) as the baseline to demonstrate performance comparison. We used raw exposures as the input for this baseline model and followed their hyperparameter settings for model training. All model setups are shown in Table 2.

[14] https://optimal-image-subtraction.readthedocs.io/en/stable/

Table 2
Performance on Test Set

Model Name	Feature		Dim	Cont	N2N	Threshold	Precision	Recall	F1-Score	Change
	Raw	Diff	$n \times 64 \times 64$							%
SDCNN (Wang et al. 2022)	✓		2	⋯	EfficientNet (Tan & Le 2019)	0.87	90.02	74.89	81.76	0.00
ODN2N-ResNet		✓	1	⋯	ResNet	0.67	5.10	98.29	9.70	−88.14
ODN2N-VGG		✓	1	⋯	VGG	0.83	5.02	99.61	9.55	−88.31
ADN2N-ResNet	✓	✓	3	ResNet	⋯	0.56	56.51	75.61	64.68	−20.89
ADN2N-VGG	✓	✓	3	VGG	⋯	0.72	48.96	71.85	58.24	−28.77
RawN2N-OTD	✓		2	⋯	OTD	0.52	84.91	81.25	83.04	+1.57
RawN2N-ResNet	✓		2	⋯	ResNet	0.65	80.71	90.27	85.22	+4.23
RawN2N-VGG	✓		2	⋯	VGG	0.72	77.80	85.29	81.37	−0.47
CT-ResNet-N2N-OTD	✓		2	ResNet	OTD	0.54	86.30	89.88	88.05	+7.70
CT-ResNet-N2N-ResNet	✓		2	ResNet	ResNet	0.73	83.39	87.72	85.50	+4.57
CT-ResNet-N2N-VGG	✓		2	ResNet	VGG	0.71	80.04	89.82	84.65	+3.53
CT-VGG-N2N-OTD	✓		2	VGG	OTD	0.50	84.40	59.57	69.84	−14.57
CT-VGG-N2N-ResNet	✓		2	VGG	ResNet	0.51	69.62	87.06	77.37	−5.37
CT-VGG-N2N-VGG	✓		2	VGG	VGG	0.62	77.88	87.44	82.39	+0.77

5.2.1. Training Hyperparameters

For the contrastive branch, we mainly followed the training methods and hyperparameters used in the previous work (Kongsathitporn et al. 2023). The input of this branch is a pair of sequences. The pairing process is performed randomly in every epochs to compare a single asteroid sample to many non-asteroid samples. The label for each epoch is determined after pairing with the binary labeling, similar class (1) and dissimilar class(0). Then, each representation extractor obtain two 64×64 exposures and generate a motion representation for an input sequence. The contrastive branch is trained to compare these motion representation from a pair of sequences and determine whether they are similar classes. In other words, the representation extractor is trained to discriminate the sequences with a moving object from other static objects, and grouping up the sequences having a similar class. The 64×64 exposures are generated from the center cropping of their 100×100 exposures. We do not apply other augmentations rather than the cropping. The batch size for representation learning is reduced to 32 samples. Each sample is randomly fed to the model without replacement. Therefore, the training steps continue until the model has processed all pairs of sequences before starting a new epoch. After the training process, the model parameters in this branch are fixed during the training for asteroid classification.

The inductive branch is trained for asteroid classification directly, where the motion representation is the additional feature concatenated at a high-level feature extraction. We use a training batch size of 1000 samples, including 500 positive and 500 negative samples. We sample the set of images using a random sampler with additional augmentations. The batch size for the test set is fixed at 256 samples, using a set of images from sliding windows. We also applied a similar batch size for the test set with `SExtractor` selection. We use binary cross-entropy[15] with equal weights as we feed equal numbers of samples. For baseline models, we use a similar batch size for both model training and testing. However, we follow the architecture from streak detection by removing our equal sampling process and using cross-entropy loss to visualize the learning process. The loss function shows the convergence of predictions to the expected results, and other metrics indicate model accuracy under the imbalance. With class-balanced batching.

For other hyperparameters, we train the models with additional early stops based on losses. The maximum number of epochs is set to 5000. However, we expect the minimum decrease in the loss value to be 0.0001, so the training process will be terminated if the loss encounters lower continuous changes for 50 epochs. In each epoch, we calculate the F1-score to save the best model and save the last model if the training process is terminated by the prior conditions. Finally, we develop the models using an NVIDIA GeForce RTX 2080Ti, and the CUDA version is 11.4.

We evaluated our experiments on the validation set to select the prediction thresholds for each model based on the validation's F1-score. Subsequently, we assessed the models on the test set to demonstrate their prediction performance. Our proposed model was compared with the baseline model for asteroid detection. The model achieving the highest F1-score on the test set was selected for performance comparison with existing software.

5.2.2. Data Augmentation

As the model architectures require two 64×64 images, we augment the asteroid training sets using random cropping,

[15] https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html

vertical flipping, horizontal flipping, and rotation with `PyTorch` (Paszke et al. 2019). Random cropping from 100×100 to 64×64 helps vary the asteroid's position, while the other augmentations alter the asteroid's movement direction. Additionally, we implement a random sampler with replacement for the asteroid sets to generate class-balanced batches. Consequently, since we have 35,912 negative samples in the training set and a batch size of 1000 samples, the classification models take 72 steps per epoch to encompass all distinct negative samples, while positive samples are randomly sampled with replacements.

5.3. Model Evaluation

We evaluate the classification tasks using precision, recall, and F1-score metrics. The model performance is highlighted by the ROC-AUC and Precision-Recall curve using the scikit-learn package.[16] The predicted class is determined based on the prediction threshold ϵ. Suppose $\hat{y}_c \in 0, 1$; the predicted class is defined as follows:

$$\hat{y}_c = \begin{cases} 1, & \text{if } \hat{y} > \epsilon. \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The equations for precision, recall, and F1-score are shown.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{11}$$

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{12}$$

where TP is true positive samples, FP is false positive samples, TN is true negative samples, and FN is false negative samples. A receiver operating characteristic curve (ROC curve) uses the true positive rate (TPR) and false positive rate (FPR) with respect to a prediction threshold and represents the model performance by the area under the curve (AUC).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{13}$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}. \tag{14}$$

5.4. Photometry Filtering

5.4.1. Determining Potential Asteroids

We perform the post processing on asteroid candidates, positive predictions from our model, to verify the potential asteroids as the prediction results do not confirm that moving objects in an exposure pair are identical. We define the potential asteroid to be an object that shows positional shifting within an area of an exposure pair and their photometry parameters should be similar. We compute the flux and flux radius of an object using ellipse aperture photometry. The source candidates are detected by `SExtractor` on template exposure to ensure that the area contains observable objects. The intention of this criterion is to classify the potential moving object into blinking/vanishing objects, distorted objects, and potential asteroids. The order of criterion and object classes are shown in Figure 7.

To identify similarity of the objects, first, the flux of objects based on the ellipse aperture is computed to consider the twisted shape of the objects due to their movement or background noise. Thus, if there is no detected object or object with similar flux, we assume that the candidates completely vanish or blink within the observation time. Let flux_{ref} and flux_{sci} denotes the flux of an object obtained by ellipse aperture photometry in reference and science exposures, respectively. Δflux denotes error of flux calculation. Therefore, if any object in science exposure has flux value within one of $\text{flux}_{\text{ref}} \pm \Delta\text{flux}_{\text{ref}}$, those objects are nearly identical objects.

Next, we compute the flux radius to ensure that the objects with nearly identical flux also have nearly similar size. The radius is defined by the size of flux half maximum. We evaluate the threshold radius using 0.1, 0.5, 1, 2, and 5 pixels. We denotes the radius of object in reference and science exposures as r_{ref} and r_{sci}, respectively. As a result, the candidates are determined to be potential asteroid if the radius difference is lower than the threshold, $|r_{\text{ref}} - r_{\text{sci}}| <$ radius threshold.

5.4.2. Hyperparameters for Potential Asteroid Filtering

To extract the photometry parameters, we utilize `SExtractor` to extract objects and compute the parameters. The objects are extracted from background-subtracted exposure where the background noise of each 64×64 pair is computed separately. We set the detection threshold to be 3σ for primary source detection. For flux calculation, we use the data summation in elliptical apertures using source position, semimajor axis, semiminor axis, and the angle of semimajor axis. We follows the hyperparameter setting in SEP documentation.[17] We set the multiplication of semimajor and semiminor axes to be 6. We uses this factor as we follows the documentation and we visually confirm the correctness of the factor. We set the gain parameter to be 1.0 to include the summation of flux to background noise to estimate flux error. For radius estimation, we compute the radius of the circle containing half of the total flux. To consider the elongation of objects, we add the semimajor axis for flux radius.

[16] https://scikit-learn.org/stable/

[17] https://sep.readthedocs.io/en/stable/apertures.html

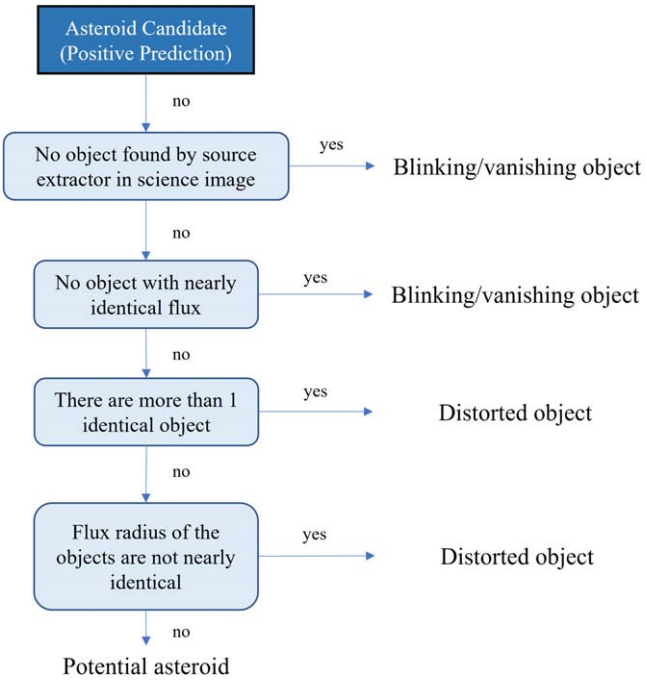


Figure 7. The diagram shows four criteria to classify the positive prediction from our best model into three types. The blinking/vanishing objects are classified based on existence of identical objects, so the science exposure that contains no object or no object with similar flux implies the disappearance of an asteroid candidate. For the last two criteria, we classify a candidate into a distorted object. If there are at least 2 objects with similar flux confirmed by previous criteria, we assume the distortion of brightness in one of the object. The radius criteria ensure that the similar flux estimated in the first two criteria is not biased by size of an object.

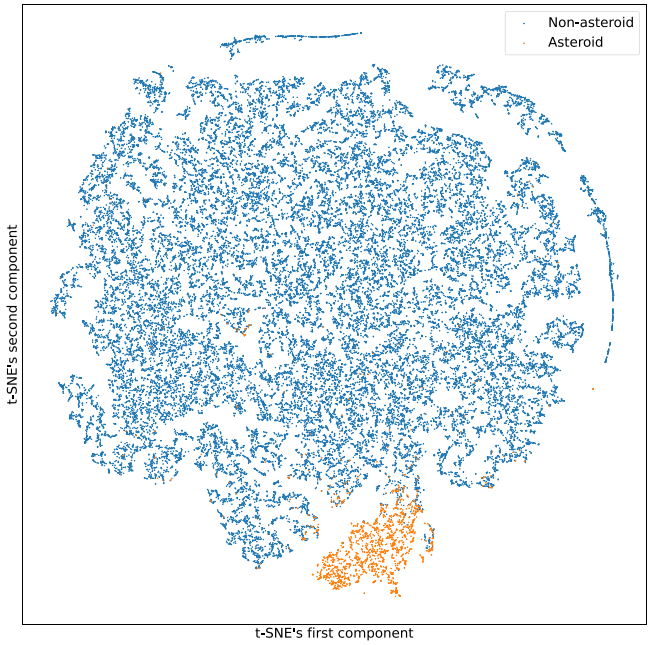


Figure 8. The outputs of ResNet-backbone contrastive branch are visualized by t-SNE. The contrastive features of asteroid and non-asteroid sequences are shown the blue and orange dots, respectively. The plot shows that the output represents contrast of the sequences by separating in high-dimensional space.

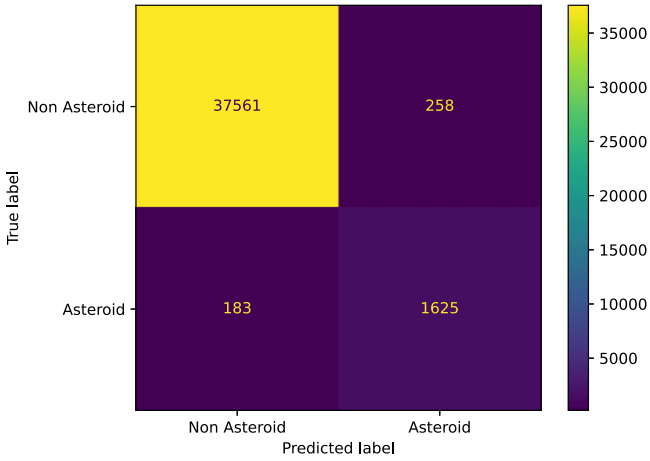


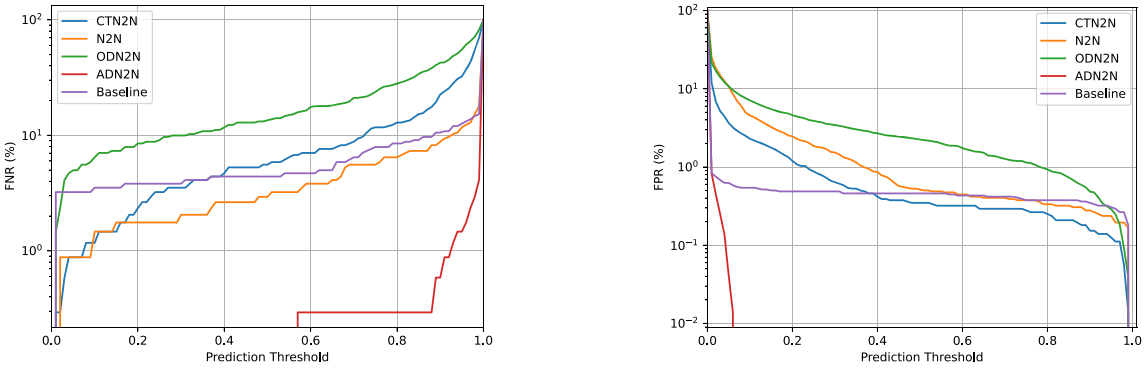
Figure 9. Confusion Matrix of CT-ResNet-N2N-OTD; The model demonstrate the best performance in test set.

6. Result and Discussion

6.1. Asteroid Classification

The t-SNE plot of the contrastive features (Figure 8) illustrates the dissimilarity between asteroid and non-asteroid sequences in the test set. According to the classification results in Table 2, the best-performing model is the end-to-end model with contrastive features using OTD and ResNet as the backbone architectures, achieving an F1-score of 88.05%. This performance surpasses the baseline model by +7.70%. While the baseline model exhibits higher precision (+4.31%), it shows lower recall (−16.68%) compared to CT-ResNet-N2N-OTD. The confusion matrix of is shown in Figure 9.

Compared to the baseline, RawN2N-OTD and RawN2N-ResNet outperform the baseline model by +1.57% and +4.23% in F1-score, respectively. The end-to-end model with VGG achieves a slightly lower F1-score compared to the baseline. These results highlight the potential of using backbones inspired by source detection models for detecting moving objects. Moreover, these backbones are less complex than the EfficientNet used in the baseline. Therefore, our experiments suggest that the backbones employed from source detection models provide comparable performance to the baseline in asteroid detection. To emphasize the impact of contrastive features, we observed that classification models incorporating contrastive features outperform end-to-end models trained solely on raw features by 0.3% to 6.0% in F1-score. The ResNet contrastive backbone enhances all RawN2N



(a) (b)

Figure 10. Each line is the representatives of each architecture achieving highest F1-score in validation set. The prediction threshold is optimized for each model to minimize numbers of false predictions. However, ADN2N shows the signal of overfitting in both plots as the model perfectly reduces false predictions at a specific threshold (a) The percentage of false negative rates with respect to all negative prediction for each prediction threshold in validation set. The plot is shown in log-scale. (b) The percentage of false positive rates with respect to all positive prediction for each prediction threshold in validation set. The plot is shown in log-scale.

performances, while the VGG contrastive backbone generally diminishes performance. Furthermore, the ResNet backbone shows less sensitivity to variations in the architecture of inductive feature extractors compared to the VGG backbone. In conclusion, integrating contrastive features enhances the representation of movement using automated contrastive extractors. Notably, our best-performing model achieves this promising performance without relying on synthetic images.

6.2. Impact of Contrastive and Difference Features

Compared to using difference images alone, models trained on difference images underperform as indicated in Table 2. We observed that combining raw sequences with difference images yields superior results compared to using difference images alone. This enhancement is particularly evident in the increased precision of ADN2N over ODN2N for both backbone architectures. However, we also identified that ADN2N faces overfitting issues. In the FPR and FNR plots of the validation set (Figures 10(a) and (b)), both fall near zero, indicating ADN2N achieves negligible false positive and false negative samples at a specific threshold, characteristic of overfitting. Hence, the parameters of ADN2N could perfectly detect asteroid on the validation set, but lose their generalizability to be implemented on the test set which are the sequences from next 6 months. Therefore, the experiment highlights the generalizability of our contrastive features compared to traditional difference image because our proposed architecture achieve significantly higher F1-score. As a result, while ADN2N parameters can perfectly detect asteroids on the validation set, they lack generalizability to the test set, which comprises sequences from the subsequent 6 months. This experiment underscores the superior generalizability of our contrastive features compared to traditional difference images, as our proposed architecture achieves a significantly higher F1-score.

The performance on the test set demonstrates that our contrastive features enhance movement representation while maintaining generalizability to detect moving objects in sequences from the subsequent 6 months (February–July). The contrastive extractor automatically generates features and is trained to handle data set imbalances effectively. Importantly, our training methods achieve promising performance without the need for synthetic images

6.3. Analysis of Photometry Filtering

After the model inference, we achieve 1883 positive prediction in total. We apply the criteria, Figure 7, on this result to determine the potential asteroids. The first criteria removes 1379 sequences equal to 76.27% of all positive predictions. The second criteria removes 467 sequences equal to 24.80% of all positive predictions. The third criteria removes 5 sequences equal to 0.27% of all positive predictions showing that there is small chance for multiple possible candidates in science exposure. Based on the reduction by these criteria, most of the positive sequences are blinking/vanishing objects where our model determines this change as the contrast of two exposures. These objects are not the potential asteroids as objects could be omitted by the noise or move outside the field of view. As shown in Table 3, 1379 sequences removed by the first criteria contain 74.22% of true positive and 67.05% of false positive sequences. Thus, there are 1846 sequences of blinking/vanishing objects and 5 distorted objects.

For the radius criteria(rm_4), we vary the radius threshold and measure numbers of potential asteroids. Figure 11 shows

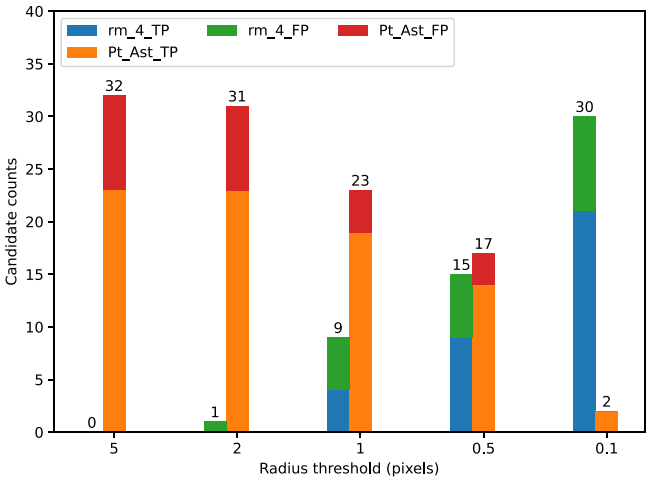


Figure 11. Visualization of potential asteroids at different radius thresholds in the test set. Pt_Ast denotes counts of potential asteroids. rm_4 denotes numbers of candidates removed by the 4th criteria, the flux radius. TP stands for true positives, and FP stands for false negatives. At 0.1 pixels of the threshold, the photometry filtering removes all false positive and provide 2 potential asteroid predicted as true positive by our best model.

Table 3
Numbers of Sequences Removed by the First Three Criteria for the Test Set

Criteria	True Positive	False Positive
No object in a science exposure (rm_1)	1206	173
No object with nearly identical flux (rm_2)	391	76
More than one possible candidate in science exposure (rm_3)	5	0

Table 4
Numbers of Sequences Removed by the Radius Similarity for the Test Set

Radius Threshold (pixels)	True Positive	False Positive
5	0	0
2	0	1
1	4	5
0.5	9	6
0.1	21	9

numbers of sequence removed from all positive prediction and numbers of potential asteroids. The numbers of the sequences removed by the flux radius threshold are shown in Table 4. We found obvious reduction at 0.5 and 0.1 pixels. Moreover, at the threshold of 0.5 pixel, the cumulative reduction of false positive sequences is 98.83%, and, at the threshold of 0.1 pixel, we achieve 2 potential asteroids, Figure 12, labeled as true positive prediction. The sequences satisfying all criteria are determined to be potential asteroids shown in Table 5.

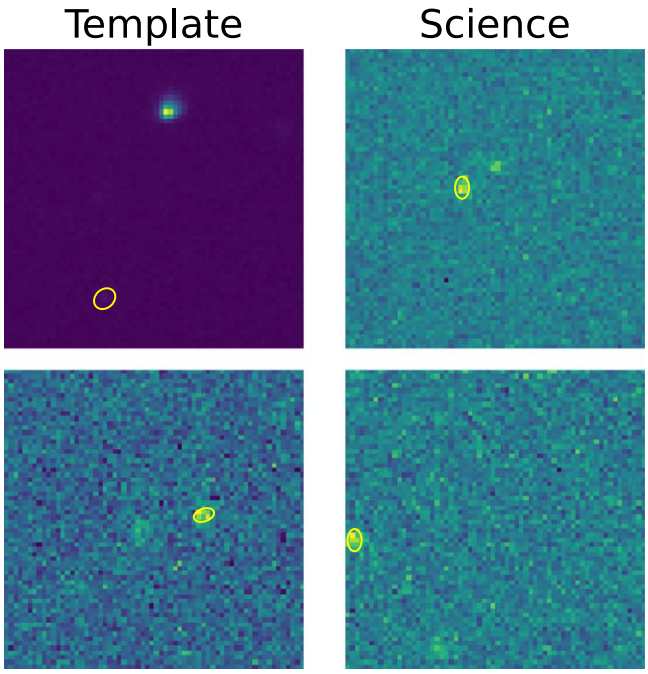


Figure 12. Visualizations of potential asteroids at the radius threshold of 0.1 pixels; The yellow ellipses shows the potential asteroids suggesting by the photometry filtering.

Table 5
Numbers of Sequences Determined to be Potential Asteroids for the Test Set

Radius Threshold(pixels)	True Positive	False Positive
5	23	9
2	23	8
1	19	4
0.5	14	3
0.1	2	0

The results show that our model could leverage the contrast between two exposures in a sequence. However, without the photometry analysis, our model could not ensure the identity of the moving/blinking objects. The utilization of photometry criteria not only filters out blinking/vanishing objects, but also selects potential asteroids based on their flux parameters. As a result, we suggests to implement our model to detect primary moving objects, and perform the photometry analysis to identify potential asteroids.

6.4. Wide-field Detection of Long-duration Data Set

To demonstrate the utilization of our model in wide-field detection, we generates additional test set from overlapped area of long-duration full-size exposures mentioned in Section 5.1. We utilizes the `SExtractor` to extract the source from the whole overlapped area before implement the post processing.

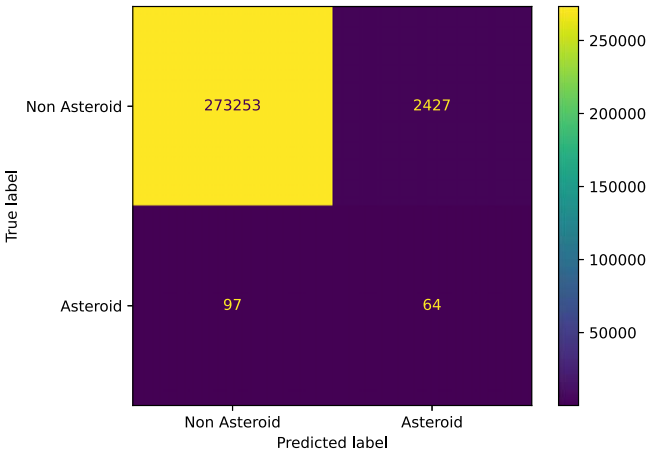


Figure 13. Confusion Matrix of CT-ResNet-N2N-OTD; The model demonstrate the best performance in wide-field test set.

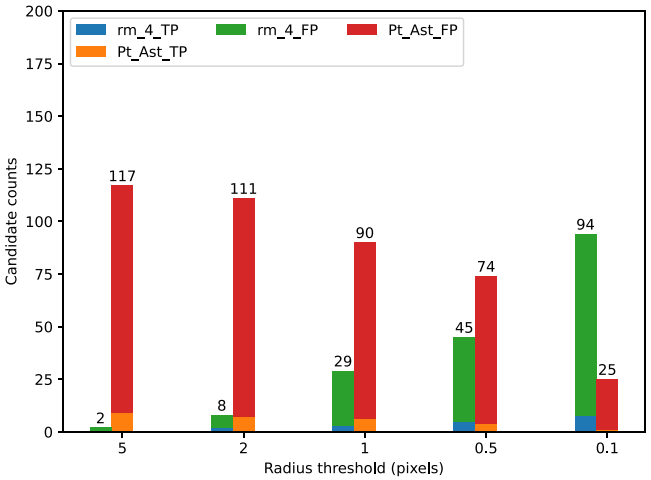


Figure 14. Visualization of potential asteroids at different radius thresholds in the wide-field detection. Pt_Ast denotes counts of potential asteroids. rm_4 denotes numbers of candidates removed by the 4th criteria, the flux radius. TP stands for true positives, and FP stands for false negatives. In this scenario, our model could not reduce all false positives at 0.1 pixels of the threshold. However, the photometry filtering help select 25 sequences from 2491 positive sequences.

We label the asteroid exposures by position collected from `SkyBoT` similar to the previous experiments, Section 6.1. As a result, we achieves 161 asteroid exposure pairs and 275,680 non-asteroid exposure pairs.

To evaluate our model and the criteria in wide-field detection, the prediction result by our best model is presented in Figure 13. We achieve 2491 positive predictions. After we apply the first three criteria, there are 1540 blinking/vanishing objects and 832 distorted objects, so 95.22% of positive predictions are removed. As shown in Figure 14, at the threshold of 0.1 pixels, we achieve 25 sequences of potential

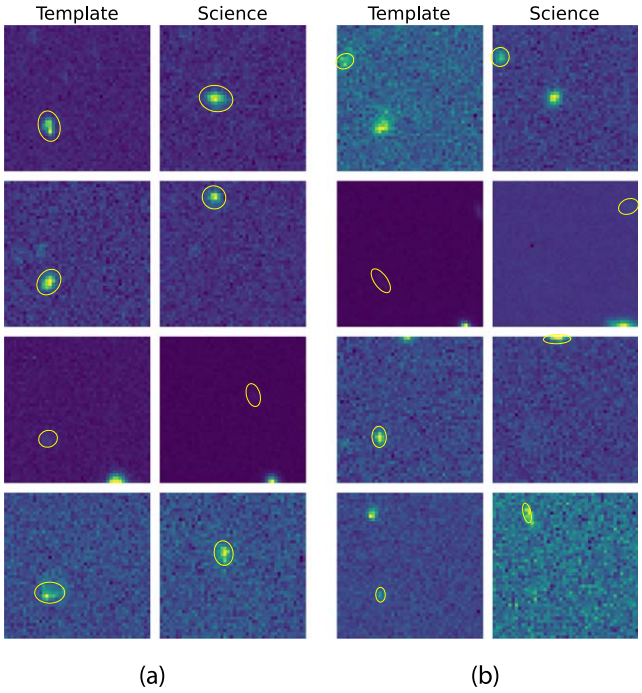


Figure 15. Visualizations of potential asteroids at the radius threshold of 0.5 pixels; The yellow ellipses shows the potential asteroids suggesting by the photometry filtering; (a) true positive sequences show the positional shifting of potential asteroids; (b) false positive sequences show the false object identification when the brightness of a static object is nearly similar to a candidate in a reference exposure. Moreover, we found the sequence that moving object is not detected by the photometry process (a) Potential asteroids from true positives (b) Potential asteroids from false positives

asteroids containing 24 false positive sequences and 1 true positive sequences.

Next, We visually inspect the potential asteroids at the threshold of 0.5 pixels, Figure 15(a), to observe more the potential asteroids from true positive sequences. We found that the positional shifting could be observe, but shapes of the potential asteroids are changed due to their brightness distribution. Therefore, the extreme threshold of 0.1 pixels could removed the latter candidates though their movement is visually observed. For false positive sequences shown in Figure 15(b), we found the exposures showing positional shifting, but they are labeled as the non-asteroid or detected as the different objects in science exposures.

The result shows that our model could detect the contrast between exposures in wide-field detection, and the photometry criteria removes 97.12% of false positive prediction at 0.5 pixels of radius threshold. The numbers of potential asteroids are shown in Table 6. The visualization of potential asteroids are shown in Figure 15 We suggest to uses 0.5 pixels instead of 0.1 pixels of the threshold as the decision is more robust to the shape of brightness distribution. The criteria help

Table 6
Numbers of Sequences Removed by the Radius Similarity for the Wide-field Detection

Radius Threshold (pixels)	True Positive	False Positive
5	0	2
2	2	6
1	3	26
0.5	5	40
0.1	8	86

Table 7
Numbers of Sequences Determined to be Potential Asteroids for the Wide-field Detection

Radius Threshold (pixels)	True Positive	False Positive
5	9	108
2	7	104
1	6	84
0.5	4	70
0.1	1	24

select 74 potential sequences, mentioned in Table 7, from the previous 2427 sequences, 3.05% of all positive prediction, for this additional step. As a result, our approaches shows the promising performance in false positive reduction in wide-field detection. it also reduce labor-intensive inspections by suggesting a small quantity of potential asteroids.

6.5. Comparative Analysis of Model Implementation

The asteroid detection software `Astrometrica` processes sequences of three consecutive exposures, each with a size of 8176×6132 pixels. In contrast, our work utilizes only the 1st and 3rd exposures to extend the sequence duration adequately. 64×64 exposures are generated by Sliding Windows (SW) and `SExtractor` (SE). The patch generation is illustrated in Figure 16. We evaluate three configurations for asteroid detection using this approach: the detection model with Sliding Windows (SW+Model), the model with `SExtractor` (SE +Model), and `Astrometrica`. Moreover, as the source extraction is a part of our experiments in this section. The ground truth is retrieved by the `SkyBoT` without the additional filtering from `SExtractor`. Thus, in this section, we compare only the prediction results from our model to investigate the asteroid candidates without the photometry filtering.

6.5.1. Software Hyperparameters

Four primary hyperparameters include scale, CCD, detection limitations, and image alignment. Scale and orientation are manually defined. Regarding scale and orientation, the focal length is set to $993.0 \pm 1.0\%$, the position angle to $90.0 \pm 10°$,

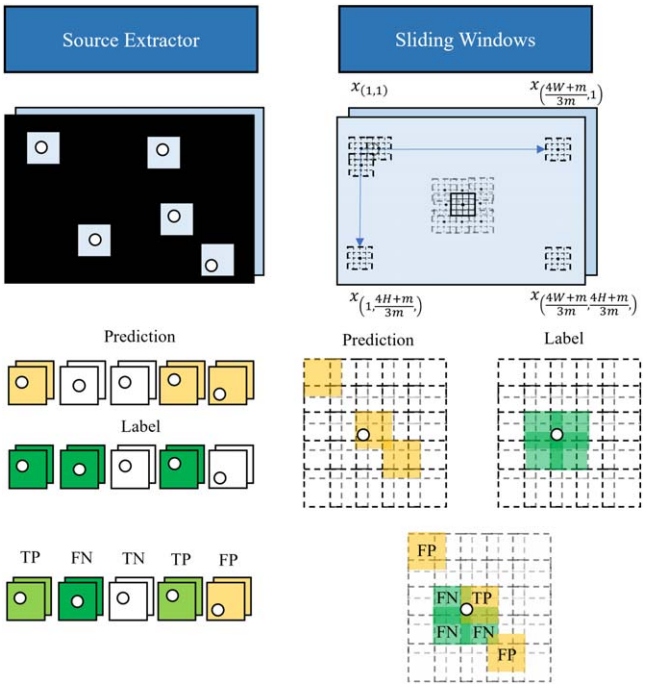


Figure 16. Patch extraction methods; (Right) The sliding window shorten 5% of the length in every edges, and slices the exposure into 64×64 patches with the overlapped area; (Left) The `SExtractor` generate the candidate positions based on its brightness profiles with respect to background noise.

and the pointing accuracy to $\pm 5\farcs0$. The CCD parameter specifies a saturation brightness value of 65000. For object detection parameters, `Astrometrica` reads background noise from the Point-spread Function (PSF) file, with a value of 0.20 for the root mean squared PSF-fit. Aperture radius and search radius are set at 3 and 0.6 pixels, respectively. The detection limit is defined as 3σ, and the minimum size for full-width half maximum is 0.70 pixels. Image alignment uses 50 reference stars. Detection results are presented in both right ascension and declination. A comprehensive list of all parameters is provided in Table 8.

6.5.2. Sequence Generation from Full-sized Exposures

We performs sequence generation in a different approach for this comparison to `Astrometrica`. The full-sized exposures are processed by two patch selectors: sliding windows and `SExtractor`. Sliding windows involves slicing the entire exposure into 64×64 windows directly. However, this method lacks overlap between patches, which can miss brightness shifts near the edges of patches. Additionally, near the edges of the full-scale exposure, there can be mapping errors due to the spherical coordinates of sky positions. To mitigate this, we reduce the field of view by 5% at each edge. The sliding stride

Table 8
Astrometrica's Detection Parameters

Header	Parameters	Value	Unit
Scale and Orientation	Focal Length	993.0 ± 9.3	mm
	Positional Angle	90.0 ± 10.0	deg
	Pointing	±5.0	’
CCD Chip	Pixel Width	6.0	μm
	Pixel Height	6.0	μm
	Saturation	65000	
Object Detection	Aperture Radius	3	pixels
	Detection Limit	3.0	sigma
	Minimum FWHM	0.70	pixels
	PSF-Fit rms	0.20	
	Search Radius	0.60	pixels
	Background from	PSF	
Plate Constants	Constant	Quadratic Fit	
Residuals	Astrometric Limit	1.00	”
	Photometric Limit	1.00	mag
Star Catalog	Catalog	Gaia DR2	
	Upper Limit	14.0	mag
	Lower Limit	19.0	mag
Reference Star Matching	Number of Stars	0	
	Search Radius	2.00	pixels
Image Alignment	Number of Stars	50	

of the 64×64 window is set to 48 pixels to ensure a 25% overlap with nearby patches along all edges. Another patch selector used is SEP,[18] which calculates background noise and evaluates source candidates based on their brightness profiles. We define the source detection threshold to be 3σ which is similar to detection threshold shown in Table 8. The output of SEP is the x–y coordinates of source candidates, so we generate the 64×64 patches by using the positions as the center of the patches.

Formally, suppose $s_i \in \mathbb{R}^{3\times W\times H}$ is a sequence of three full-sized exposures taken consecutively. Removing the second exposure reduces the dimension of the sequence to $s'_i \in \mathbb{R}^{2\times W\times H}$. As the best model evaluated in Section 6, an input sequence is defined by $x_i \in \mathbb{R}^{2\times m\times m}$ where m is the size of a window for asteroid detection. In this part, we extract the $m \times m$ windows using SExtractor (SE) and Sliding Window (SW) on the $W \times H$ exposures. As we shorten the field of view for 5% in every edge, the sliding area is reduced from $W \times H$ to $w \times h$. The ranges of w and h are $[0.5W, 0.95W]$ and $[0.5H, 0.95H]$, respectively. Let N_s be the numbers of source found by SExtractor in the first exposure of a sequence s'_i. For the sliding windows, the stride size is determined by $\frac{3}{4}m$, so the overlapped area after a slide is $\frac{1}{4}m^2$. The patch extraction is performed as follows:

$$\mathrm{SE}(s'_i) = \{p_i | i \in \{1, 2, \ldots, N_s\}\}$$

$$\mathrm{SW}(s'_i) = \{p_{xy} | x \in \{1, \ldots, \frac{4h+m}{3m}\}, y \in \{1, \ldots, \frac{4w+m}{3m}\}\},$$

where $p \in \mathbb{R}^{2\times m\times m}$ denotes a patch extracted by each extractor. In our work, the width (W) and height (H) are 8176 and 6132 pixels, respectively. The patch size (m) is 64 pixels, resulting in a stride size of 48 pixels. The extractions are illustrated in Figure 16. Then, the patches p from each extractor are fed to the model as the input sequence x_i.

For Astrometrica, we input the exposure sets directly into the software and collect the positions of all predicted asteroids to compare the detection results with our models. Suppose p'_a is a candidate position detected by Astrometrica. p'_a is stored in sky coordinates, as shown in Equation (15). We compute the distance threshold ϵ_a in the sky coordinate. In our work, we uses a patch size of 64×64, so the distance threshold is 32 pixels to ensure that there is an asteroid in a patch using p'_a as the center. We define the list of all asteroids (p_a) in a sequence using SkyBoT. The label for each position is determine by y_a. Moreover, all prediction results from Astrometrica (y'_a) are positive class. The formal expressions are define as follows:

$$p_a = \{(\text{R.A., decl.}) | (\text{R.A., decl.}) \in \mathbb{R}^2\}, \quad (15)$$

$$y_a = \mathrm{Dist}(p'_a, p_a, \epsilon_a), \quad (16)$$

where R.A. are decl. are right ascension and declination in units of degree; Dist is a function that calculates the distance between a candidate in $p'a$ to all asteroid in p_a, and labels them as positive sample if the distance is less than ϵ_a; The label and prediction are determined by $y_a, y'_a \in \{0, 1\}$.

Table 9
Performance in Comparison to Astrometrica

Score	SW+Model	SE+Model	Astrometrica
True positive counts (TP)	5192	98	930
False Discovery Rate (FP/P)	99.38	99.64	96.75
Unique asteroid counts (Ua)	4816	87	344
Percentage of Ua/TP	92.76	88.76	36.99
Similar names to Astrometrica	9	0	

6.5.3. Detection Performance

We follows the evaluation method mentioned in Section 5.4.2 where the parameters are determined by y_a and y'_a. In this section, we begin by extracting patches and collecting their center positions. We label every patch bounding each asteroid as positive class, while the rest are labeled as negative class. The sliding window encounters

[18] https://sep.readthedocs.io/en/v1.1.x/

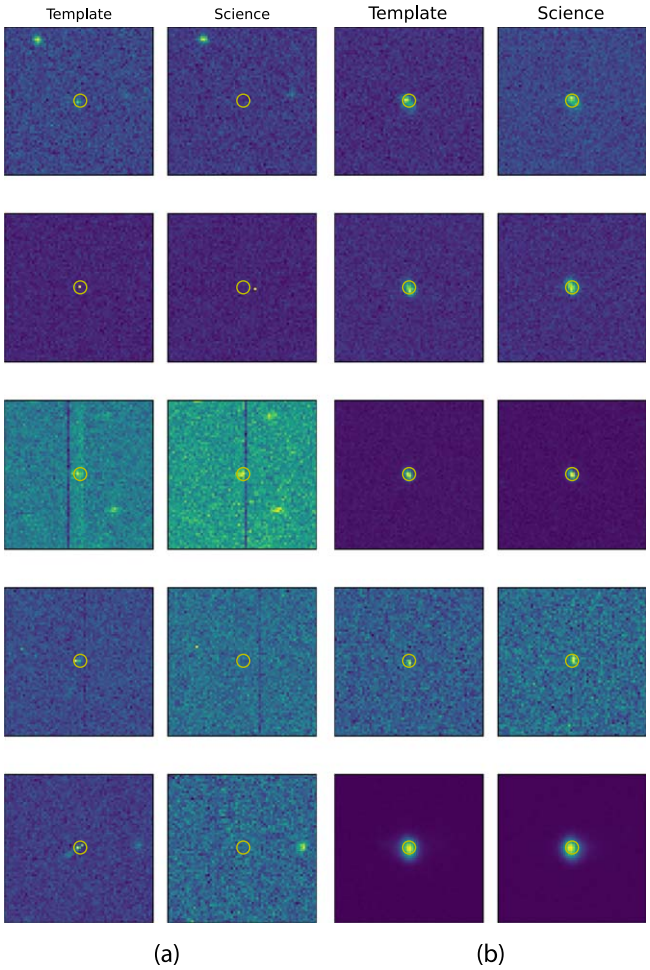


Figure 17. Visualizations of sequences from positive predictions; (a) True positive sequences shows the positional shifting where an asteroid is located near the centers of a template exposure; (b) False negative sequences shows false prediction due to the brightness variation which is not the actual movement (a) True positive (b) False negative.

duplicate counting due to the $\frac{1}{4}$-overlapped area at the boundaries of each patch. The evaluation method is demonstrated in Figure 16.

The result reflects that our proposed model only complements `Astrometrica` because our model found additional targets on top of the software, Table 9. Even our model still generate high false discovery rate, we found that the true positives from sliding window method provides 28 times higher than `Astrometrica`. The model detection with `SExtractor` (SE+Model) found 87 additional asteroids which were not detected by `Astrometrica`. Moreover, our model detections tends to avoid similar target detection as the percentage of unique asteroid names (Ua/TP) are 92.76%, 88.76% for sliding windows (SW+Model) and `SExtractor` (SE+Model), respectively.

6.5.4. Visual Analysis

When the duration of real data is short, the model could not detect obvious movement. Thus, the result shows that our approach achieves unpromising performance. We perform visual inspection to confirm this conclusion. In Figure 17(b), the false negative samples mostly appear static within the sequences, while some true positive samples (Figure 17(a)) do not exhibit obvious positional shifts because the duration between exposures is not sufficiently long. These objects do not show evident movement similar to the data set in the previous section, and the detection models struggle to discriminate between asteroid and non-asteroid samples.

For false positive detections, we can classify the samples into three types based on their visualization including brightness error, disappearance ,and brightness shifting objects. First, the brightness error (Figure 18(a)) is caused by sufficiently bright events such as cosmic ray hits or satellite trails, where the detection model incorrectly detects them as brightness shifts of moving objects. Second, there are objects that `SExtractor` identifies in a template exposure but which disappear from the field in the following exposure (Figure 18(b)). This disappearance is caused by variations in brightness over time. Lastly, in Figure 18(c), there are false positive samples showing positional shifting. This suggests that our model could provide additional evidence for further exploration. We found 5–10 such samples out of 320 random samples; however, we identified these false positives solely through visual inspection.

As a result, our model achieves surpass percentage of unique asteroid per true positives (Ua/TP) compared to `Astrometrica` The asteroid candidates suggested by our model with `SExtractor` are notably different based on their names. Therefore, in this scenario, our model complement `Astrometrica` results as the model could not significantly false positives. We suggest using our detection model alongside `Astrometrica` to identify additional asteroid candidates that have already been discovered or candidates of unknown moving objects. We also demonstrated the use of the model with source extraction methods including sliding windows and `SExtractor`. We also suggests to implement our models with the traditional pipelines (Copandean et al. 2017, 2018; Stănescu & Văduvescu 2020; Golovich et al. 2021) as we demonstrate the uses of our model with the source detection tools. Moreover, our model is trained using the predefined duration sequences, Table 1, so our detection process is recommended for long-duration sequences shown in Section 6.5.

7. Conclusion

We propose an end-to-end convolutional network for moving objects detection under an imbalance scenario. We perform the performance analysis and comparative

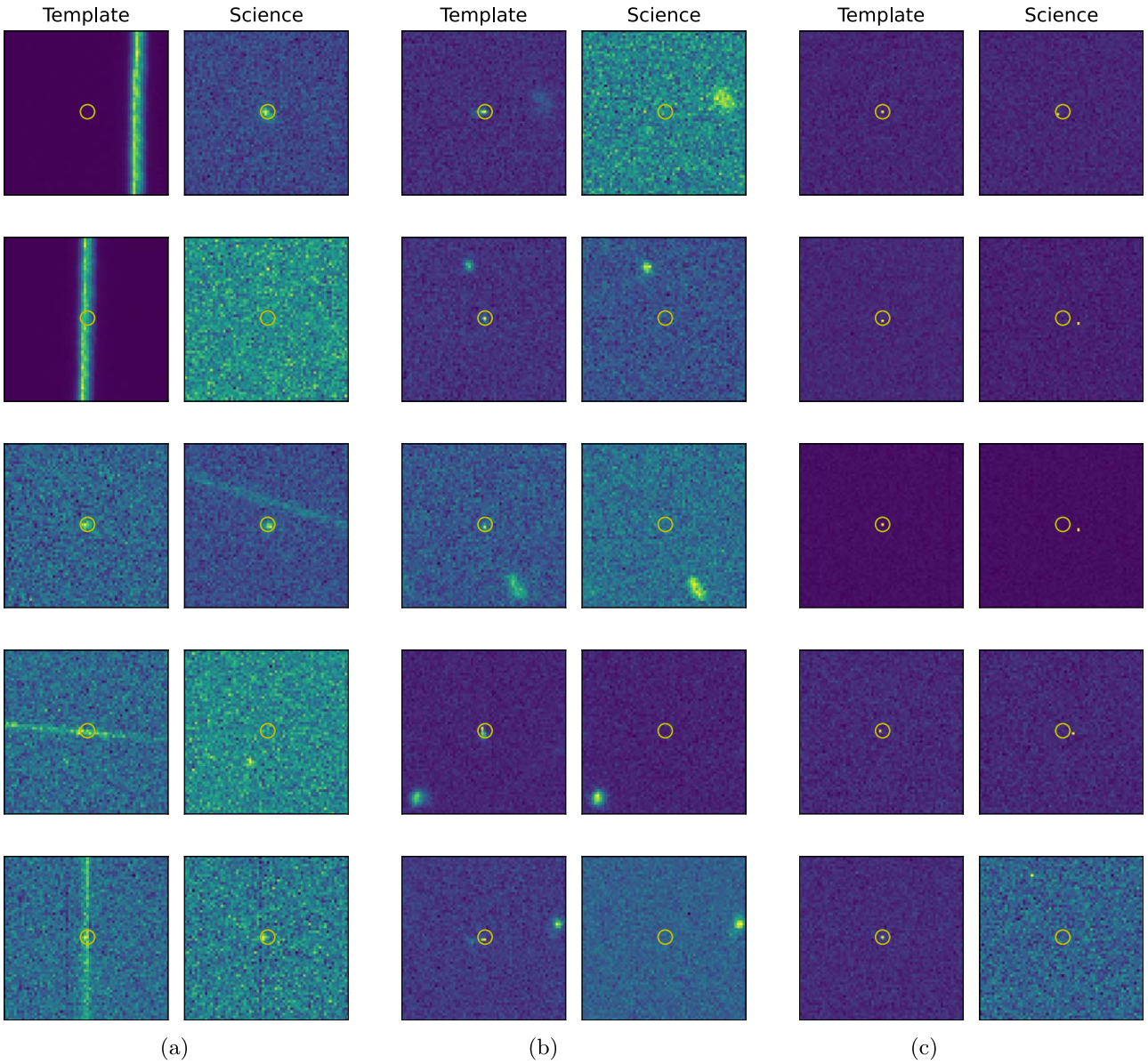


Figure 18. Visualizations of false positive predictions found by SE+Model; (a) there are the brightness error in template or science exposures; (b) the objects disappear in the science exposures due to the their own brightness variation and neighbors' brightness; (c) the approach found moving object candidates that do not exist in the ground truth, so this result demonstrates that our model could also suggest asteroid candidates for new asteroid searching(a) Brightness Error (b) Disappearance objects (c) Moving object candidates.

implementation using the exposures from GOTO. The movement is represented by concatenating contrastive and inductive features. While the model trains to generate inductive features, the pretrained contrastive branch helps emphasize the contrast between static sources and moving objects.

Our best model achieved an F1-score of 88.05 on the test set, surpassing both the end-to-end model without contrastive features and the baseline model (Wang et al. 2022). Remarkably, this performance was achieved without the use of synthetic images. The detection process is automated and does not rely on predefined parameters, contributing to its promising results. The additional photometry criteria suggests the potential asteroids atop the movement detected by our model. For wide-field detection of long-duration sequences, our model and photometry filtering reduce 97.12% of false positive prediction. However, our model did not perform as well on full-sized exposures used by `Astrometrica`. Despite generating a high false discovery rate, the detection results

using sliding windows and SExtractor found additional asteroids that were not detected by Astrometrica. Therefore, we suggest implementing our model as a complement to Astrometrica for achieving additional candidates in the full-sized exposures.

Our work has several limitations. First, we extended the duration between exposures to ensure clear movement, which resulted in our model achieving less promising results on full-sized detection where the sequences duration is shorter than those in our newly generated sequences. Second, the motion representation models are trained on the data set labeled by SkyBoT only which is restricted to the brightness mentioned in Section 5.1. Lastly, we used OIS with Bramich's algorithm (Bramich 2008) for generating difference images, as the mapping between exposures in our sequences is not entirely consistent. Other subtraction tools such as Hotpants (Becker 2015) and Miller's algorithm (Miller et al. 2008) could be explored in the future to compare with our model.

Acknowledgments

This project is funded by National Research Council of Thailand (NRCT). This work was also supported by a National Astronomical Research Institute of Thailand (NARIT) and Thailand Science Research and Innovation (TSRI) research grant. We are also grateful to Assoc. Prof. Dr. Suppawong Tuarob and Asst. Prof. Dr. Thanapon Noraset from the Faculty of Information and Communication Technology, Mahidol University for their suggestions, and moral support. The Gravitational-wave Optical Transient Observer (GOTO) project acknowledges the support of the Monash-Warwick Alliance; University of Warwick; Monash University; University of Sheffield; University of Leicester; Armagh Observatory & Plane- tarium; the National Astronomical Research Institute of Thailand (NARIT); Instituto de Astrof´ısica de Canarias (IAC); University of Portsmouth; University of Turku. We acknowledge financial support from the Agencia Estatal de Investigaci\'on of the Ministerio de Ciencia e Innovaci\'on MCIN/AEI/10.13039/501100011033 and the ERDF "A way of making Europe" through project PID2021-125627OB-C32, and from the Centre of Excellence "Severo Ochoa" award to the Instituto de Astrofisica de Canarias.

Facility: GOTO.

Software: astropy (Astropy Collaboration et al. 2013), OIS (Bramich 2008), SExtractor (Bertin & Arnouts 1996), Astrometrica (Raaba 2002).

ORCID iDs

Noppachanin Kongsathitporn https://orcid.org/0009-0006-1850-986X
Akara Supratak https://orcid.org/0000-0002-6739-7642
Kanthanakorn Noysena https://orcid.org/0000-0001-9109-8311
Supachai Awiphan https://orcid.org/0000-0003-3251-3583
Amit Kumar https://orcid.org/0000-0002-4870-9436
Martin J. Dyer https://orcid.org/0000-0003-3665-5482
Enric Pallé https://orcid.org/0000-0003-0987-1593

References

Acero-Cuellar, T., Bianco, F., Dobler, G., Sako, M., & Qu, H. 2022, AJ, 166, 115
Ackley, K., Eikenberry, S. S., Yildirim, C., Klimenko, S., & Garner, A. 2019, AJ, 158, 172
Aldahoul, N., Karim, H. A., Momo, M. A., Escobara, F. I. F., & Tan, M. J. T. 2023, IEEEA, 11, 5089
Almoznino, E., Loinger, F., & Brosch, N. 1993, MNRAS, 265, 641
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Becker, A., 2015 HOTPANTS: High Order Transform of PSF ANd Template Subtraction, Astrophysics Source Code Library, record, ascl:1504.004
Berthier, J., Vachier, F., Thuillot, W., et al. 2006, in ASP Conf. Ser. 351, Astronomical Data Analysis Software and Systems XV, ed. C. Gabriel et al. (San Francisco, CA: ASP), 367
Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393
Bijaoui, A. 1980, A&A, 84, 81
Bramich, D. M. 2008, MNRAS, 386, L77
Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, ApJ, 836, 97
Carrasco-Davis, R., Reyes, E., Valenzuela, C., et al. 2021, AJ, 162, 231
Carruba, V., Aljbaae, S., Domingos, R. C., Huaman, M., & Barletta, W. A. 2022, CeMDA, 134, 36
Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. E. 2020, in Proc. of the 37th International Conference on Machine Learning, 119, (PMLR), 1597
Copandean, D., Nandra, C. I., Gorgan, D., & Vaduvescu, O. 2018, in 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) (IEEE), 1
Copandean, D., Vaduvescu, O., & Gorgan, D. 2017, in 2017 13th IEEE Int. Conf. on Intelligent Computer Communication and Processing (ICCP) (IEEE), 377
Cowan, P., Bond, I. A., & Reyes, N. H. 2022, A&C, 42, 100693
Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2020, arXiv:2010.11929
Duriscoe, D. M., Luginbuhl, C. B., & Moore, C. A. 2007, PASP, 119, 192
Dyer, M. J., Ackley, K., Jiménez-Ibarra, F., et al. 2024a, Proc. SPIE, 13094, 130941X
Dyer, M. J., Ackley, K., Lyman, J., et al. 2022, Proc. SPIE, 12182, 121821Y
Dyer, M. J., Ackley, K., Lyman, J. D., et al. 2024b, Proc. SPIE, 13094, 130941X
Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2018, A&A, 616, A1
Gieseke, F., Bloemen, S., van den Bogaard, C., et al. 2017, MNRAS, 472, 3101
Golovich, N. R., Lifset, N., Armstrong, R., et al. 2021, arXiv:2104.03411
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Las Vegas, NV, USA), 770
Khlamov, S. V., Savanevych, V. E., Briukhovetskyi, O. B., & Pohorelov, A. V. 2016, Proc. Int. Astron. Union, 12, 349
Khosla, P., Teterwak, P., Wang, C., et al. 2020, arXiv:2004.11362
Koch, G. R. 2015
Kongsathitporn, N., Supratak, A., Awiphan, S., et al. 2023, in 2023 27th Int. Computer Science and Engineering Conf. (ICSEC) (Samui Island, Thailand) (IEEE), 9
Krizhevsky, A. 2009, Master's thesis, University of Tront
Kron, R. G. 1980, ApJS, 43, 305
LeCun, Y., & Bengio, Y. 1998, Convolutional Networks for Images, Speech, and Time Series (Cambridge, MA: MIT Press), 255
Lin, Z., Wang, Y., & Lin, H. 2022, in 2022 IEEE Int. Conf. on Multimedia and Expo (ICME) (Taipei, Taiwan) (IEEE), 1
Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, PASP, 131, 038002
Miller, J. P., Pennypacker, C. R., & White, G. L. 2008, PASP, 120, 449

Moller, A., & de Boissière, T. 2020, MNRAS, 491, 4277
Mong, Y.-L., Ackley, K., Galloway, D. K., et al. 2020, MNRAS, 499, 6009
Morice-Atkinson, X., Hoyle, B., & Bacon, D. 2018, MNRAS, 481, 4194
Parfeni, A. A., Caramete, L. I., Dobre, A. M., & Bach, N. 2020, arXiv:2010.15425
Paszke, A., Gross, S., Massa, F., et al. 2019, arXiv:1912.01703
Pravdo, S. H., Rabinowitz, D. L., Helin, E. F., et al. 1999, AJ, 117, 1616
Raaba, H. 2002, in Proceedings of Meeting on Asteroids and Comets in Europe (MACE) (Visnjam), 1
Rabeendran, A. C., & Denneau, L. 2021, PASP, 133, 1021
Reyes, E., Estévez, P. A., Reyes, I., et al. 2018, in 2018 Int. Joint Conf. on Neural Networks (IJCNN) (IEEE), 1, arXiv.1808.03626
Rudenko, M. 2016, Proceedings of the International Astronomical Union, 318, 265
Russakovsky, O., Deng, J., Su, H., et al. 2014, Int. J. Comput. Vision, 115, 211
Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., & Chen, L.-C. 2018, in 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (Salt Lake City, UT, USA) (IEEE), 4510
Savanevych, V. E., Khlamov, S. V., Akhmetov, V. S., et al. 2022, A&C, 40, 100605
Simonyan, K., & Zisserman, A. 2014, CoRR, arXiv:1409.1556
Steeghs, D., Galloway, D. K., Ackley, K., et al. 2022, MNRAS, 511, 2405
Stănescu, M., & Văduvescu, O. 2020, A&C, 35, 100453
Takahashi, I., Hamasaki, R., Ueda, N., et al. 2022, PASJ, 74, 946
Tan, M., & Le, Q. V. 2019, arXiv:1905.11946
Tedesco, E. F. 1994, in IAU Symposium 160, Asteroids, Comets, Meteors 1993, ed. A. Milani, M. di Martino, & A. Cellino (Dordrecht: Kluwer Academic Publishers), 463
Tian, Y., Sun, C., Poole, B., et al. 2020, arXiv:2005.10243
Tyson, J. A. 2002, Proc. SPIE, 4836, 10
Wang, F., Ge, J., & Willis, K. 2022, MNRAS, 516, 5785
Wardęga, K., Zadrożny, A., Beroiz, M., Camuccio, R., & Díaz, M. C. 2021, MNRAS, 507, 1836
Wright, D. E., Lintott, C. J., Smartt, S. J., et al. 2017, MNRAS, 472, 1315
Wright, D. E., Smartt, S. J., Smith, K. W., et al. 2015, MNRAS, 449, 451
Zhai, C., Shao, M., Nemati, B., et al. 2014, ApJ, 792, 60