



Article A Novel Deep Learning-Based Relabeling Architecture for Space Objects Detection from Partially Annotated Astronomical Images

Florin Dumitrescu ^{*,†}[®], Bogdan Ceachi ^{*,†}[®], Ciprian-Octavian Truică ^{*}[®], Mihai Trăscău [®] and Adina Magda Florea [®]

- Computer Science and Engineering Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, RO-060042 Bucharest, Romania
- * Correspondence: florin.dumitrescu97@upb.ro (F.D.); bogdan.ceachi@upb.ro (B.C);
 - ciprian.truica@upb.ro (C.-O.T.)
- + These authors contributed equally to this work.

Abstract: Space Surveillance and Tracking is a task that requires the development of systems that can accurately discriminate between natural and man-made objects that orbit around Earth. To manage the discrimination between these objects, it is required to analyze a large amount of partially annotated astronomical images collected using a network of on-ground and potentially space-based optical telescopes. Thus, the main objective of this article is to propose a novel architecture that improves the automatic annotation of astronomical images. To achieve this objective, we present a new method for automatic detection and classification of space objects (point-like and streaks) in a supervised manner, given real-world partially annotated images in the FITS (Flexible Image Transport System) format. Results are strongly dependent on the preprocessing techniques applied to the images. Therefore, different techniques were tested including our method for object filtering and bounding box extraction. Based on our relabeling pipeline, we can easily follow how the number of detected objects is gradually increasing after each iteration, achieving a mean average precision of 98%.

Keywords: astrometry; space surveillance; image processing; relabeling; deep learning; object detection; classification

1. Introduction

As space assets are becoming a serious concern in terms of security and safety across the world, Space Surveillance and Tracking is the field that utilizes different optical and radar sensors to collect space data in order to create an inventory of objects that are orbiting around Earth [1]. To monitor and survey the space objects, sensors need to be fast and accurate to predict the trajectory and prevent spacecraft collisions. The collected data are in turn used to provide the best information to governmental protection services to prevent uncontrolled events. Thus, the aim of the Space Surveillance programs is to support the utilization of and the access to space-related information for research or services by providing timely and quality visual and radar sensor data. This information can be used to develop real-time environment-aware knowledge-based services that can mitigate any threats and offer the support for sustainable exploitation of the outer space surrounding our planet.

The Space Surveillance and Tracking (SST) of both natural and man-made objects requires the development of architectures that can identify in real-time the orbiting objects around Earth, estimate their orbital parameters, and determine the evolution of their orbits. These systems must be able to (1) collect sensor data in form of images from a network of on-ground and potentially space-based optical telescopes. (2) process images, individually



Citation: Dumitrescu, F.; Ceachi, B.; Truică, C.-O.; Trăscău, M.; Florea, A.M. A Novel Deep Learning-Based Relabeling Architecture for Space Objects Detection from Partially Annotated Astronomical Images. *Aerospace* 2022, *9*, 520. https:// doi.org/10.3390/aerospace9090520

Academic Editor: Peng Wei

Received: 13 July 2022 Accepted: 14 September 2022 Published: 17 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). for object detection or sequentially for tracking, to extract and then discriminate between man-made and natural objects.

Even though the task of detecting and discriminating between space objects in astronomical images is not new, this task is a very hard one. The current solutions in the literature either use synthetic datasets [2] or apply a classifier on the objects extracted from the image using SExtractor [3]. Collecting a dataset of images acquired by Space Surveillance Optical Telescopes and annotating the objects of interest is no easy feat, as the image quality is impacted by a lot of external factors, such as clouds, light pollution, operational temperature, etc., making template matching an unreliable method for the object detection task. Because of this, each object of interest should be manually annotated as either *point-like* or *streak*, which is an arduous and time-consuming process. Thus, the main objective of this article is two-fold:

- Propose a novel relabeling architecture for satellite detection from partially annotated astronomical images;
- Benchmark our proposed analysis using a real-world dataset.

To address the first objective, the proposed architecture utilizes partially annotated images to extend a real-world dataset with more annotated visible objects in each input image. This pipeline consists of an object detector and classifier ensemble. To implement the object detection model, we use the Detectron 2 [4] framework that employs Faster R-CNN [5] with ResNet-18 [6] as a backbone. The classifier ensemble uses a convolutional encoder with three convolutional layers and a dense block with three fully-connected layers to discriminate between point-like and streak objects in the images. We train our deep learning models in a *K*-fold manner. For the second objective, we perform an in-depth analysis using our architecture with a real-world dataset.

The rest of this paper is structured as follows. In Section 2, we discuss the current state-of-the-art methods related to relabeling astronomical images for space object detection. In Section 3, we present our methodology and the proposed novel architecture. In Section 4, we present the experimental setup and our results on a real-world dataset. In Section 5, we discuss our findings and hint at current challenges that we identify for the task of image relabeling for space object detection. Finally, in Section 5, we present conclusions and outline future directions.

2. Related Work

Algorithms for searching and recognizing celestial objects using neural networks have been used since 1988 [7], but, unfortunately, the hardware capabilities at that time were not so advanced. Thus, algorithms such as subgraph isomorphism and pattern recognition [8] are frequently used to identify and search celestial objects, but with time expense in terms of database search. With the advancement of technology, deep learning methods become frequently used, managing to obtain much better results than the classic algorithms. One big advantage of their use in identifying celestial objects is by removing the time spent searching for them in databases because all the extracted features of an object are stored in the network.

In a recent work, Rijlaarsdam et al. [9] demonstrate how using a small fully connected network is the preferred choice in terms of speed and lightweight design for the star identification algorithm, being able to classify a large number of stars.

Another similar approach is used by Yang et al. [10] where the authors proposed a one-dimensional Convolutional Neural Network (1D CNN) to identify stars in catalogs, being highly robust to position and magnitude noise with an identification accuracy of 98%.

Jia et al. [11] investigates the use of relatively simple convolutional (CNN) and recurrent (RNN) network architectures to classify transient astronomical objects into three categories: point, artifact or streak. The classification achieves an accuracy of over 98%, but the candidate images in their case come directly from the SExtractor utility, the classification not being executed on the primary images. González et al. [12] look for deep learning methods for detecting and identifying galaxy types (e.g., spiral, elliptical, edge-on) from telescope images (e.g., Hubble) or from the Galaxy Zoo archive. The paper investigates the use of an adapted version of the YOLOv3 [13] network (an already classic network for detecting objects in everyday life) for the detection and classification of galaxy types. An important aspect that the authors are looking for is to increase the amount of data, especially by different methods of converting the image space from FITS (Flexible Image Transport System) format to RGB format, in order to exploit the existing YOLO network.

One of the problems of existing object detection and classification networks is that they are less efficient when the objects of interest are small (i.e., they occupy a small area as the number of pixels in the picture). Thus, depending on the performance of the fine-tuning approach of the existing networks, an alternative method can be explored, which modifies the existing networks to become more efficient in the detection of small objects.

Techniques of this kind try to increase the image area to which the convolutional layers look (effective receptive field), as well as the construction of auxiliary objectives for segmenting small objects (e.g., Sun et al. [14]) modifying the types of proposed regions (i.e., rectangular templates, mode common for searched objects) for object detection, or the use of feature pyramid networks.

3. Method and Processing Pipeline

3.1. Data Preprocessing

The dataset used in our experiments was collected using the *gendared* application, which is described in Piso et al. [15] and was developed by GMV. The *gendared* pipeline [15,16] receives as input the raw images in FITS format from optical telescopes and generates as output astrometric and photometric data for the target objects detected in an observation image.

The main goal of our proposed pipeline is to detect the objects of interest in astronomical images, which can appear as *point-like* or *streaks* corresponding to either satellites or stars, depending on the observational context, using neural network approaches. The images that we are using are coming from two telescopes with different fields of view and image sizes (i.e., 2.106° width/ 2.105° height for an image size of 4096×4096 px for one telescope and around 43.9 arcminutes width/29.25 arcminutes height for an image of size 2004×1336 px for the other), as well as different exposure times. In one telescope, stars appear as streaks because it is tracking satellites, while with the other they appear as points because the telescope is compensating for the sky rotation. The resolution is high enough to not influence the appearance of the visible objects (i.e., a point-like object is unlikely to appear as a streak or vice versa). However, there are still objects that appear in star catalogs that are not visible in our images.

Because FITS files contain raw pixel data, represented as 16-bit floating point values, we could not train an object detection model directly on them, as the interval on which the pixel values are defined is not fixed. As such, we decided to convert the images from FITS format into JPEG format using different preprocessing techniques. We had to decide what attributes in our data are the most important, for example, the details of a specific set of values or the full dynamic range. A common tactic is by using two main types of transformations: normalization and stretching of image values. Image values can be set in the range [0, 1] using lower and upper limits or by using a linear or non-linear function. In literature, those approaches are frequently used, one example is the most popular framework Astropy [17], which provides several classes for automatically determining intervals (e.g., linear stretch with a slope and offset, log stretch, a sinh stretch, using image percentiles or interval based on IRAF's z-scale). The first technique was to use the 1st and the 99th percentile of the raw data values in each image and map them into smaller values inside the [0, 255] range. However, the resulting images, were extremely noisy, causing the annotation filtering and bounding box generation algorithm to be very sensitive to the choice of its parameters, and in turn, limiting the ability of the object detection model to

learn; an example can be observed in Figure 1a. Since the brightest sources are often orders of magnitude brighter than the low-level variations in the sky background, we used the logarithmic function to stretch the interval of pixel values, resulting in a higher contrast between the image background and the objects of interest, as can be observed in Figure 1b.





We used as ground truth for the object detection algorithm the centroids of the objects detected using the *gendared* pipeline. However, the detection algorithm requires either bounding boxes around the objects or segmentation masks to learn their features. Thus, we used computer vision methods to extract the bounding box of each object based on the provided centroid. The idea is that if an object is present at the centroid's coordinates, then the contour of the object is also centered and is unique. To filter out the outliers (i.e., image noise annotated as an object of interest) we also computed the object's contour area and, if it is under a set threshold, the object is discarded. Algorithm 1 describes this procedure. A drawback of this approach is the dependency on the image preprocessing method employed in the previous step, as we have to manually set the parameters of the functions. Our solution involves visually inspecting the results of the algorithm at different parameter values and choosing those values that result in the least number of outliers. The resulting annotated dataset has then been validated by the experts.

1	• • • •	- 1	$-\infty$	• •	C•1.	• •	- 1	1	1.	1		
	OOP1TP	nm I	()r	MOCT	±111	toring	and	hound	ding	nov	ovtra	Ction
	goiiu		. Ul	JUUL	111 U		anu	Douin	unig	DUA	UNITIA	Cuon
				,								

Require: (c_x, c_y) , image	
Ensure: $(x_{min}, y_{min}, x_{max}, y_{max})$	
1: crop \leftarrow extractCrop (image, (c_x, c_y))	
2: binaryCrop ← binaryThreshold (crop)	
3: contours	
4: for each cnt \in contours do	
5: cntArea ← computeContourArea (cnt)	
6: if cntArea \leq areaThreshold then	Drop any small contour
7: continue	
8: end if	
9: if not isCentered (cnt) then	▷ Drop any uncentered contour
10: continue	
11: end if	
12: $(x_{min}, y_{min}, x_{max}, y_{max}) \leftarrow fitBoxAroundContour$	(cnt)
13: return $(x_{min}, y_{min}, x_{max}, y_{max})$	
14: end for	
15: return None	

Because the images had different sizes and some were too big to reasonably train the model with, we had to slice them to a smaller, fixed size with a small overlap to ensure that no objects were removed during this process. Any object that becomes partially visible in an image slice is discarded, as it will most likely still be fully visible in an adjacent image

slice. A file containing the relationships between the original image and its slices is also saved to be able to map the detections on the slices to the original image.

To build the dataset that is used in this pipeline, we used the image slices from both telescopes and split them into train and test sets. As we want to re-annotate the entire dataset using this pipeline, we use *K*-fold cross-validation, i.e., the original set is split into *K* equal groups, where we use K - 1 groups for training and the last group for validation.

3.2. Architecture Description

The problem with our dataset is that only a small percentage of the total number of visible objects in images are annotated and a supervised object detection algorithm has trouble learning which objects are of interest or not. However, it will also detect a lot more than the annotated objects, albeit with low confidence or assigning the wrong class.

Filtering out the outliers using Algorithm 1 helps the learning process by reducing the number of noise patches classified as objects of interest and increasing the confidence of the object detector, but does not completely solve our problem. By using a secondary image classification algorithm trained on crops centered around the filtered annotated objects, we can remove any noise patch classified as an object of interest by the detector. Unlike an object detector, which looks after candidate regions of interest in an image (i.e., can detect an unannotated object), a classifier only has access to the image patches (crops) that we are sure to contain the objects of interest (point-like or streaks), as well as noise (background). Because of this, the classifier has less variance in the training data than the object detector. The results are then saved in the same format as the original image annotations and the process is repeated for other train-test splits.

We call the full *K*-fold cross-validation procedure a re-annotation step, with the pipeline replacing the annotations at the end of each step, i.e., when the pipeline finishes relabeling the test set of the i^{th} train-test split, the new annotations are not immediately used in the $(i + 1)^{th}$ train-test split but instead are stored until the next relabeling step. Figure 2 presents an overview of our pipeline for a single train-test split, while Figure 3 shows how a single re-annotation step works. After the last re-annotation step, the objects detected in the image slices are mapped back to the original image.



Figure 2. An overview of our re-annotation model. It consists of 3 stages: the detection stage, in which the object detection model is trained and the objects of interest are detected on the validation set; the cropping stage, in which crops for the classifier are extracted based on the detections from the previous stage; the re-annotations stage in which the classifier is trained on the extracted crops of the objects of interest from the train set, as well as noise, and then used to detect which objects found in the detection stage are valid. In the last stage, the valid detected objects are also saved to be used in the next re-annotation step.



Figure 3. An example of a single re-annotations step. For the sake of this example, we split the dataset into 4 sets, and, for each model, we choose 1 set to be re-annotated while the others are used to train the model. At the end of the re-annotation stage, the entire dataset is re-annotated. The re-annotation step can be repeated multiple times.

We used Detectron 2 [4] framework to implement our object detection algorithm: Faster R-CNN [5] with ResNet-18 [6] as backbone. We chose to use an existing framework, as it allows us to easily implement and test different architectures, such as Faster R-CNN, which is already available. As for the backbone of choice, we found that using a smaller backbone does not negatively impact the performance of the detection algorithm compared to using a deeper one, such as ResNet-50, while also reducing the computational cost of the model.

To create a dataset to train the classification model, we used a modified version of Algorithm 1, which saves the cropped objects as images of a fixed size and also keeps a mapping between the cropped images and their coordinates in the original image. It also extracts random noise patches that do not intersect with any detected object or contain possible objects of interest (using the same contour area thresholding logic as in Algorithm 1). This cropping algorithm is also used to save image crops of the detected objects from the test set.

For the classifier, we used a simple and fast architecture, as our objects of interest appear as *point-like* or *streaks* in the images. It consists of two parts: a convolutional encoder with three convolutional layers and a dense block with three fully-connected layers. Each convolutional block has a convolutional layer with a kernel size of 3×3 and after each layer Batch Normalization is used for network convergence rate, followed by a rectified linear unit (ReLU) activation function and max pooling operation (stride 2) to reduce the dimensionality via spatial downsampling. The final output layer corresponds to the number of possible output classes (line, point and noise). Due to the limited data available and the class imbalance between the number of point-like and streaks, a weighting random sampling technique was used along with data augmentations, such as rotations and flipping. For this type of architecture, we achieved 98% accuracy, and what should be noted is that a low complexity network, such as the one described, can be scaled up to be used with other environments.

In Figure 4, we can see the image heatmaps for the areas that our network is looking at in order to decide which class to assign to the image. We can assume that the network always looked in the center of the image and took the shape of our objects of interest (line, point and noise). In general, this is exactly how any other classical computer vision techniques (template matching and canny) used in this domain of expertise would approach such a task. The resulting procedure is fast and with good results, but it depends on the type of preprocessing techniques applied before the training process.



Figure 4. The gradient heatmaps for our 3 classes (computed using GradCam [18]).

4. Results

In our scenario, we used real-world images coming from two different telescopes and, because of the problems that we mentioned, there is a limited number of experiments we can conduct. Unfortunately, we do not have a gold standard due to the partially annotated dataset and are unable to measure the performance of our pipeline using standard quantitative metrics (i.e., the AP metric, which is commonly used to evaluate object detection algorithms, is negatively impacted by the high number of unannotated objects). Because of this, our scope is not to present how an object detector would fare on real astronomical data, but to propose a general pipeline that can be implemented in similar problems.

Our re-annotation pipeline was trained and evaluated using a collection of 70,243 image slices of size 512 × 512 px. We trained the detector for 15 K iterations for each data split, with a batch size of 16 images and using SGD with an initial learning rate of 1×10^{-3} and a step scheduling policy, the learning rate being multiplied with 0.1 after 10 K and 12.5 K steps, respectively. The classifier was trained for 6 epochs on image crops of size 70 × 70 px (which was determined experimentally to fit all objects), using AdamW [19] with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-2} . The learning rate was reduced according to an exponential learning rate scheduling method with $\gamma = 0.5$.

As shown in Table 1, after each iteration, the number of detected objects is gradually increasing, with the model achieving a score of 0.62 mAP in detecting *point-like* or *streaks* objects after two iterations.

		Total	
	Point	Line	
Initial	14,205	22,893	
Iteration 1	36,817	32,397	
Iteration 2	44,814	36,437	

Table 1. Number of objects of interest in the entire dataset at different steps of the re-annotations pipeline.

A visual representation of the results from the pipeline at each iteration for a sample image is shown in Figure A1. For each step starting from left to right, the procedure will rely heavily on the trained classifier to keep or eliminate proposed objects coming from the detector. We used a confidence threshold of 0.6 for the detector to find as many candidates in the image as possible. We also dropped any objects for which the average confidence score between the detector and classifier scores was lower than 0.6. The results at each steps are represented in Tables 2 and 3.

Comparison		Unchanged		Rem	oved	Added	
		Point	Line	Point	Line	Point	Line
Initial Initial	Iteration 1 Iteration 2	11,184 7280	17,381 17,087	3020 6925	5506 5802	25,626 37,530	15,015 19,350
Iteration 1	Iteration 2	34,797	30,663	2020	1734	10,017	5774

Table 2. Comparison between the number of unchanged, removed and newly added objects in the entire dataset at different steps of the re-annotation pipeline.

Table 3. Quantitative results before and after each re-annotation step. If not specified, AP is considered the mean average precision calculated at different IoU thresholds between 0.5 and 0.95 with a step size of 0.05.

	mAP	mAP@0.5	mAP@0.75	AP-Point	AP-Line
Initial	34.01	63.78	33.08	36.06	31.96
Iteration 1	59.34	95.67	71.9	59.01	59.68
Iteration 2	62.25	98.04	77.68	59.98	64.53

5. Discussion

It is important to mention that in the initial step we do not have a gold standard (due to the partially annotated dataset), and we are not able to evaluate the performance of our initial object detector using any quantitative metric (i.e., the initial detector will always be weak). This is why we use a secondary classifier to validate the candidate objects proposed by the object detector. The secondary object detector is also dependent on the filtered dataset, as it uses the annotated data in the training stage, but has fewer outliers than the object detector. There may be annotated objects that are not visible in the images, regardless of the preprocessing method used, although they may exist in star catalogs such as 2MASS [20], GAIA [21] and Tycho-2 [22].

Our pipeline manages to overcome this obstacle by eliminating false positive annotations and detecting new objects. By comparing the performance of the object detection algorithm before and after each step of the pipeline, we observe an increase in the quality of the detections, both quantitatively due to the lower number of unannotated objects (see Table 2) and qualitatively (see Figure A1), the model being able to detect objects of interest with higher confidence. Figure A1 also illustrates how the classification model filters possible false positives. However, regardless of the models used in the pipeline, the process can be repeated only a limited number of times as, considering the nature of the pipeline, the models can become biased towards the results from the previous iteration. This also emphasizes that our pipeline is heavily reliant on the classifier and the data with which it is trained.

A possible limitation of the approach is that the time required for a single step is directly proportional to the size of the dataset and the number of splits. For example, for a *K* split of our dataset, both the detector and the classifier will need to be trained *K* times.

Another limitation of our pipeline is the object filtering and bounding box extraction algorithm, which is reliant on the hyper-parameters obtained via experimentation that are different depending on the quality of the telescope images. As such, a more generalized method that is less sensitive to the choice of hyper-parameters needs to be explored. A good outlier (bounding box with no object inside, only noise) removal technique is essential to train any supervised classification model. A starting point could be to explore adaptive binarization or segmentation techniques based on classic computer vision methods, the rest of the algorithm remaining the same. We can also eliminate the need for the threshold area of the counter used to determine whether an annotation contains an object or not, through an unsupervised image categorization process. For example, one could use an image feature extraction algorithm (e.g., SIFT [23], SURF [24] etc.) followed by a dimension reduction

step (e.g., PCA [25]) and a non-parametric clustering algorithm (e.g., K-means [26]) to filter out the false positive annotations.

Given that the images in the dataset are captured in bursts of at least length 3, we wish to expand our pipeline by implementing an additional validation by a tracking stage, which should remove some of the bias created by training the pipeline on the results from a previous step of the re-annotation process. We need at least three images in sequence to compute a trajectory for every object detected using the detector-classifier ensemble, as the distance between the position of an object in 2 consecutive frames can be quite large due to the high exposure time needed to capture an image.

After a reliable dataset is built using the aforementioned pipeline, we can use it to train any model, which can then be utilized to detect objects in astrometry images in real-time.

6. Conclusions

This work demonstrates that datasets of astronomic images with incomplete data (i.e., not all visible objects of interest are annotated) can be re-annotated using our proposed approach. To achieve our first objective to relabel satellite detection from partially annotated astronomical images, we developed a novel architecture composed of a detection algorithm, which identifies the celestial bodies in the images, and a classifier, which filters the detected objects and removes any noisy or low confidence results.

To achieve our second objective of benchmarking our approach, we used the dataset provided by *gendared*, which consists of real-world images from telescopes that were preprocessed using the *gendared* application, and performed an in-depth analysis of the proposed novel relabeling architecture on it. The object filtering and bounding box extraction algorithm relies heavily on the chosen hyper-parameters, which makes our approach less flexible when it comes to using it on images of different quality. New techniques need to be explored to achieve better or similar results without relying on manually choosing the parameters. Some possible future approaches have been proposed.

The experimental results using a real-world dataset prove that the pipeline increased the number of annotated objects by at least twofold, which means that a lot of the visible objects of interest were not annotated. The decrease in the number of objects that were not originally annotated improves the accuracy of the detection algorithms with higher confidence.

Author Contributions: Conceptualization, F.D., B.C., C.-O.T. and M.T.; methodology, F.D., B.C., C.-O.T. and M.T.; software, F.D. and B.C.; validation, F.D. and B.C.; formal analysis, F.D., B.C., C.-O.T. and M.T.; investigation, F.D. and B.C.; resources, A.M.F.; data curation, F.D. and B.C.; writing—original draft preparation, F.D., B.C., C.-O.T., M.T. and A.M.F.; writing—review and editing, C.-O.T., M.T. and A.M.F.; visualization, F.D. and B.C.; supervision, C.-O.T., M.T. and A.M.F.; project administration, A.M.F.; funding acquisition, A.M.F.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset will be made available upon request.

Acknowledgments: This work was supported in-part by a grant of the Romanian Ministry of Education and Research, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PTE-2019-0554 "Îmbunătățirea Capacităților Funcționale ale Aplicației GENDARED prin Utilizarea Algoritmilor de Inteligență Artificială" (Artificial intelligence upgrade for the generic data reduction framework for space surveillance), within PNCDI III and in-part by and University Politehnica of Bucharest.

Conflicts of Interest: The authors declare no conflict of interest.

10 of 11

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
R-CNN	Region-based Convolutional Neural Network
YOLO	You Only Look Once
ResNet	Residual Convolutional Neural Network
GradCAM	Gradient-weighted Class Activation Mapping
RGB	Red-Green-Blue
SGD	Stochastic Gradient Descent
AP	Average Precision
mAP	Mean Average Precision
SST	Space Surveillance and Tracking
SExtractor	Source-Extractor
2MASS	Two Micron All Sky Survey
FITS	Flexible Image Transport System
JPEG	Joint Photographic Experts Group
gendared	The Generic Data Reduction Framework for Space Surveillance

Appendix A





Figure A1. Comparison between the annotated objects (left) and the detector results (right) at different steps of the re-annotation pipeline.

References

- 1. Allahdadi, F.A.; Rongier, I.; Wilde, P.D. Safety Design for Space Operations; Butterworth-Heinemann: Oxford, UK, 2013.
- Fletcher, J.; McQuaid, I.; Thomas, P.; Sanders, J.; Martin, G. Feature-Based Satellite Detection Using Convolutional Neural Networks. In Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI, USA, 17–20 September 2019.
- 3. Bertin, E.; Arnouts, S. SExtractor: Software for source extraction. Astron. Astrophys. Suppl. 1996, 117, 393–404. [CrossRef]
- 4. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2, Version 0.5. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 20 August 2021).
- 5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
- 6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Alvelda, P.; Martin, A.M.S. Neural Network Star Pattern Recognition for Spacecraft Attitude Determination and Control. Proc. NIPS 1988, 1, 314–322.
- 8. Rijlaarsdam, D.; Yous, H.; Byrne, J.; Oddenino, D.; Furano, G.; Moloney, D. A Survey of Lost-in-Space Star Identification Algorithms Since 2009. *Sensors* 2020, 20, 2579. [CrossRef] [PubMed]
- 9. Rijlaarsdam, D.; Yous, H.; Byrne, J.; Oddenino, D.; Furano, G.; Moloney, D. Efficient Star Identification Using a Neural Network. Sensors 2020, 20, 3684. [CrossRef] [PubMed]
- 10. Yang, S.; Liu, L.; Zhou, J.; Zhao, Y.; Hua, G.; Sun, H.; Zheng, N. Robust and Efficient Star Identification Algorithm based on 1D Convolutional Neural Network. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, 1. [CrossRef]
- 11. Jia, P.; Zhao, Y.; Xue, G.; Cai, D. Optical Transient Object Classification in Wide-field Small Aperture Telescopes with a Neural Network. *Astron. J.* **2019**, *157*, 250. [CrossRef]
- 12. González, R.; Muñoz, R.; Hernández, C. Galaxy detection and identification using deep learning and data augmentation. *Astron. Comput.* **2018**, 25, 103–109. [CrossRef]
- 13. Redmon, J.; Farhadi, A. YOLOV3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767. [CrossRef]
- 14. Sun, S.; Yin, Y.; Wang, X.; Xu, D.; Zhao, Y.; Shen, H. Multiple receptive fields and small-object-focusing weakly-supervised segmentation network for fast object detection. *arXiv* **2019**, arXiv:1904.12619.
- Piso, A.M.A.; Voicu, O.; Sprimont, P.; Bija, B.; Lasheras, Ó.A. gendared: The Generic Data Reduction Framework for Space Surveillance and Its Applications. In Proceedings of the The 8th European Conference on Space Debris, Darmstadt, Germany, 20–23 April 2021.
- 16. Bija, B.; Lasheras, O.A.; Danescu, R.; Cristea, O.; Turcu, V.; Flohrer, T.; Mancas, A. Generic Data Reduction Framework for Space Surveillance. In Proceedings of the The 7th European Conference on Space Debris, Darmstadt, Germany, 18–21 April 2017.
- Price-Whelan.; A.M.; Sipőcz, B.M.; Günther, H.M.; Lim, P.L.; Crawford, S.M.; Conseil, S.; Shupe, D.L.; Craig, M.W.; Dencheva, N.; Ginsburg, A.; et al. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *Astron. J.* 2018, 156, 123. [CrossRef]
- 18. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2019**, *128*, 336–359. [CrossRef]
- 19. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. arXiv 2017, arXiv:1711.05101.
- 20. Skrutskie, M.F.; Cutri, R.M.; Stiening, R.; Weinberg, M.D.; Schneider, S.; Carpenter, J.M.; Beichman, C.; Capps, R.; Chester, T.; Elias, J.; et al. The Two Micron All Sky Survey (2MASS). *Astron. J.* **2006**, *131*, 1163–1183. [CrossRef]
- Gaia Collaboration; Brown, A.G.A.; Vallenari, A.; Prusti, T.; de Bruijne, J.H.J.; Babusiaux, C.; Bailer-Jones, C.A.L.; Biermann, M.; Evans, D.W.; Eyer, L.; et al. Gaia Data Release 2. Summary of the contents and survey properties. *Astron. Astrophys.* 2018, 616, A1. Available online: http://xxx.lanl.gov/abs/1804.09365 (accessed on 12 December 2021). [CrossRef]
- 22. Høg, E.; Fabricius, C.; Makarov, V.V.; Urban, S.; Corbin, T.; Wycoff, G.; Bastian, U.; Schwekendiek, P.; Wicenec, A. The Tycho-2 catalogue of the 2.5 million brightest stars. *Astron. Astrophys.* **2000**, *355*, L27–L30.
- 23. Lindeberg, T. Scale Invariant Feature Transform. Scholarpedia 2012, 7,10491. [CrossRef]
- 24. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded up robust features. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Volume 3951, pp. 404–417. [CrossRef]
- Karl Pearson, F.R.S. LIII. On lines and planes of closest fit to systems of points in space. Dublin Philos. Mag. J. Sci. 1901, 2, 559–572. [CrossRef]
- 26. Lloyd, S. Least squares quantization in PCM. IEEE Trans. Inf. Theory 1982, 28, 129–137. [CrossRef]