Streak detection in astronomical images based on convolutional neural network

Streak detection in astronomical images based on convolutional neural network

A A Elhakiem[1] **, T S Ghoniemy**[1] **and G I Salama**[1]

[1] Department of Computer Engineering & A.I., Military Technical College, Cairo, Egypt.

E-mail: Ahmed.A.Elhakiem@gmail.com

Abstract. Streak detection is important in space situational awareness and space asset protection. It is desirable to detect moving targets (e.g., satellites, space debris, or meteorites) in images of the sky. This paper presents a comparison between two astronomical frameworks for streak detection based on deep CNNs. The first framework uses the extended feature pyramid network (EFPN) with faster region-based CNNs (Faster R-CNN) and compares it with the second framework that uses the feature pyramid network (FPN) with Faster R-CNN. Because there aren't enough publicly available astronomical data sets, we use the simulated data set to train the neural network. The metrics of mean average precision (mAP), recall, precision, and F1 score were used to measure the performance of the two frameworks. The experimental results confirmed that the EFPN-based framework achieves a significant improvement in streak detection than the Faster R-CNN framework based on the FPN model.

1. Introduction

There is an increased interest in sky monitoring because the number of astronomical objects (active satellites and space debris (SD)) is continuously increasing. SD is all artificial targets created by explosions, including elements and fragments that revolve around the orbit of the earth. The amount of SD is rapidly increasing, and it poses a significant risk to other space stations and satellites in orbit. If SD collides with a satellite, it will cause satellite damage. For this reason, it is necessary to monitor the motion and the position of SD and satellites to make an accurate prediction for preventing accidents [1].

Streak detection is a vital process in monitoring the sky. Optical observations are used for the detection of celestial objects in the sky. Astronomical images that have a long exposure time are taken sequentially. The SD or satellites appear as streaks in these images. A streak is a line segment parameterized by position, length, width, brightness, and angular orientation. Finding streaks in astronomical images has been the main focus of earlier research on the topic of streak detection. Astronomical images usually suffer from severe noise effects that make a vast challenge for the streak detection step. The most common sources of noise in astronomical images may include atmospheric disturbances like airglow, cosmic rays, and hot and cold pixels as well as instrument misalignments like shot noise, read-out noise, and dark noise. Astronomical images also have several types of background variations that could be due to the clouds or the light pollution from bright stellar targets such as the Moon, and other reasons. So, the denoising process is vital to preprocess the astronomical images, to decrease the harmful impact of this noise. Various machine-learning techniques have been proposed to denoise astronomical images like denoising using dictionary learning [2], and Astro U-net framework [3].

The celestial objects appear in astronomical images as small objects around (5×5) pixels for stars with a moderate signal-to-noise ratio (SNR) and extend to tens of pixels for bright stars and streaks [4]. So the detection of small objects like streaks poses a challenge because it is difficult to gather data about these objects. Also, the faint streaks and the overlapping celestial objects with streaks pose a challenge in the detection.

Recently, many different CNNs frameworks have been presented for streak detection in astronomical images like YOLO-v2 [5], DeepStreaks [6], and Faster R-CNN based on FPN [4]. Inspired by their work, we propose a comparison between the EFPN-based framework [7] and the FPN-based framework [4] for streak detection in astronomical data sets. Because there aren't enough publicly available astronomical data sets, we use the

simulation observation data set to estimate the performance of the two frameworks. We discover that the EFPN-based framework [7] is more effective at achieving satisfactory accuracy in detecting streaks in astronomical images than the FPN-based framework [4].

The paper is organized as follows: Section 2 presents the related work associated with this paper. Section 3 discusses the structure of an EFPN-based framework [7]. Section 4 presents the experimental results of the comparison between the EFPN-based framework [7] and the FPN-based framework [4] for streak detection. Section 5 presents the conclusion of the paper.

2. Related work

Recently, the detection of streaks in astronomical images has attracted increasing attention in space situational awareness. This section provides a quick overview of the existing streak detection studies that have been divided into 2 groups: traditional and CNNs frameworks.

2.1. Traditional streak detection

There are many frameworks proposed for streak detection. Ding et al. [8] presented a framework based on the Hough transform (HT) and the prior orbit data. This framework can detect streaks in the low earth orbit (LEO) effectively. Santoni et al. [9] proposed a framework for the automatic detection of SD that appear as streaks in astronomical images based on the HT model and the morphological filtering technique. It detects SD without a-priori data and is based on a single image's analysis. It can detect SD in both the object-tracking and the sidereal operations. Zhou et al. [10] implemented an optical masking technique for streak detection in geostationary-earth orbit (GEO). It presents a technique for eliminating the noise and detecting SD that appear as streaks with (SNR > 3) in astronomical images. It consists of three steps: the first step is to remove the background by using The top hat method, the second step is to remove stars by using a masking method, and the final step is to detect the SD by using a weighted model. Niu et al. [11] proposed a real time streak detection technique using the multicore DSP. It can detect weak and faint celestial objects and SD and it includes two processes the first one is the SD detection technique and the second one is the DSPs' scheme. These proposed frameworks frequently have a respectable performance depending on the prior data, but the computation load and the probability of detection can still be improved.

Sease et al. [12] proposed a framework for streak detection depending on phase congruency transform (PCT) that had four processes. The first process is edge detection used to determine if the target belongs to a streak or a star. The second process is thresholding which PCT is thresholded using a histogram of the pixel intensities to produce a binary image. The third process is cleaning in which the image is cleaned by making any pixel less than four non-zero neighbors in the adjacent eight pixels to zero. The fourth process is background removal which the background is calculated as a polynomial of order quadratic and subtracted to get rid of large-scale gradients. This method is simple, but the detection accuracy can still be enhanced. The Hough Transform (HT) [13] framework uses Gabor filters (GFs) to recognize line segments and then detect streaks. A subtracted image is processed with each GF by convolving it with the filter, creating a response image. These response images are handled by the HT for detecting line segments. Despite the simplicity of this technique, the detection probability can still be increased.

2.2. Deep learning-based streak detection

Varela et al. [5] proposed a CNN streak detection framework (YOLO - v2). This framework is tested on a system with a multi camera wide-field of view (WFoV) which has a (60 - 160) degree. The structure of the YOLO-v2 contains layers such as convolutional, and max-pooling layers, and also one concatenation layer. It makes a good performance improvement (specifically in precision). By using a GPU, it processed about 4.7 images every second to satisfy the real-time requirements for the wide-field of view (WFoV) system.

Duev et al. [6] proposed a CNN streak detection framework (DeepStreaks) that is designed to detect fast and near targets from the earth that appear as streaks in the astronomical images. It achieves a (96 - 98)% true positive rate while preserving a false positive rate of less than one percentage. DeepStreaks framework has decreased human participation in the streak detection procedure. But, the ability to detect fainter and weak celestial targets in images is still to be further improved [6].

Krucinski et al. [14] proposed a CNN framework for single streak detection of moving objects like a satellite in astronomical images captured from an optical camera. It divides the process into 2 steps the first step is the detection of streaks and the second step is the localization. It is faster and more accurate. However, it can not detect several streaks in the same image.

Xi et al. [1] proposed a CNN framework for detecting SD that appears as a streak in astronomical images using FLCR. It divides into more than steps. The first process is to remove the noise from astronomical images by the

one-dimensional mean iteration technique. The second process is the FLCR framework presented to detect SD from the extracted candidate regions. It has good performance and can detect targets with low SNR, but the probability of detection can still be improved.

Jiang et al. [15] proposed a CNN framework depending on a small SD detection that had two processes. In the first process, the astronomical image's spatial contrast map (SCM) is produced using the local-contrast technique. In the second process, the spatiotemporal data is taken by combining it with the SCM. The framework has good robustness and also it is effective for small SD detection. However, lens glare has an impact on the framework that is sensitive to it.

Liu et al. [4] propose an astronomical detection framework based on CNNs to detect and classify celestial objects. It approves the Faster R-CNN by using an adapted Resnet-50 and FPN model to extract features from astronomical images. It improves (25%) of the detection ability than the classical technique when the threshold is (0.6). It is more reliable and has a good performance than the traditional method. However, the accuracy of detection can still be improved.

Due to the astronomical images nature, the above mentioned approaches had restrictions that could be summarized as First: the detection accuracy of fainter streaks is still to be further improved. Second: can not detect several streaks in the same image. Third: sensitive to the impact of noise. Forth: the accuracy of streak detection in astronomical images can still be improved. Accordingly, we propose a modified Faster R-CNN framework that overcomes such limitations by extracting features from astronomical images using EFPN.

3. Streak detection using deep learning

Since astronomical images' exposure time is short and the spatial sampling rate is low, astronomical images have around (5×5) pixels for stars with moderate SNR and extend to tens of pixels for bright stars and streaks. To assign an appropriate feature level for small celestial targets like streaks, we use EFPN as a feature extraction network in the Faster R-CNN [7] and then compare it with the FPN-based framework [4]. The FPN-based framework is discussed in detail in [4]. But now we focus on the EFPN-based framework in this paper [7]. Figure 1 demonstrates the general structure of The EFPN-based framework [7].

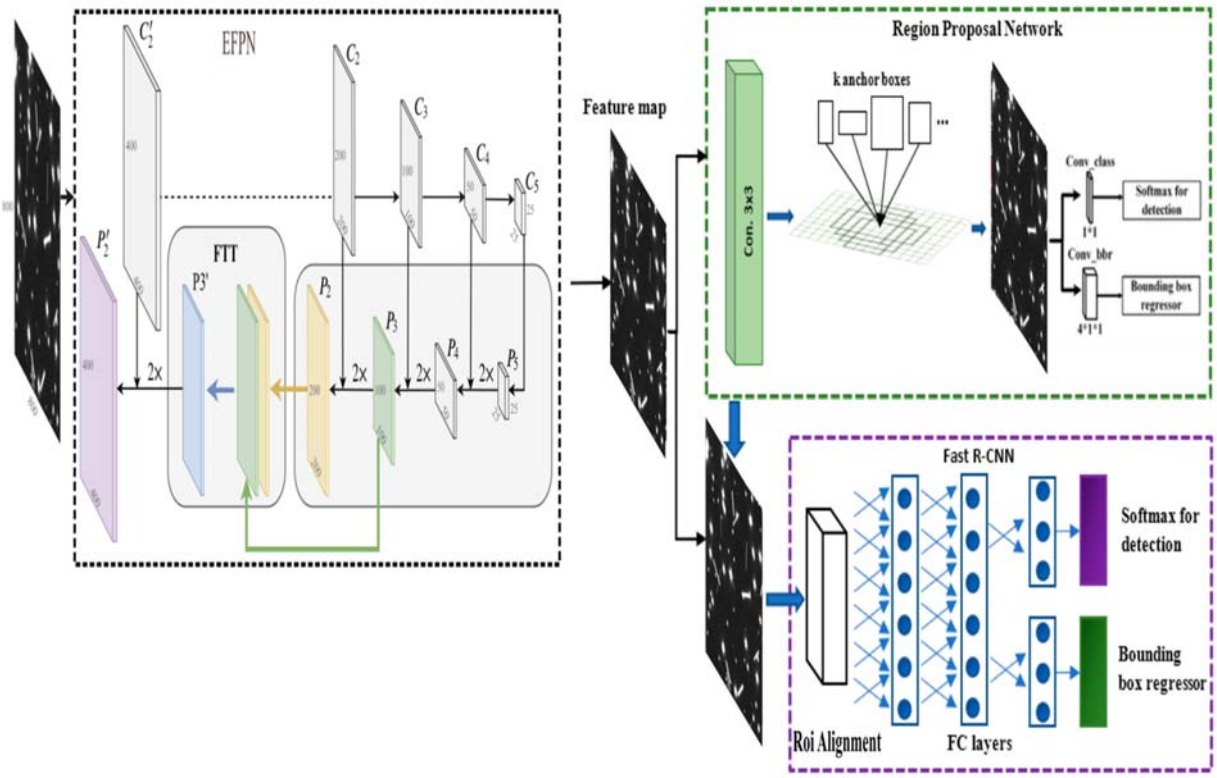

Figure 1. The overall construction of The EFPN-based framework [7].

3.1. Extended Feature Pyramid Network

The EFPN [7] architecture is implemented by the FPN-like network [4] integrated with a feature super-resolution (SR) model. This model produces high-resolution features to support small and faint streaks from low-resolution astronomical images, whereas keeping a low cost of computation. Figure 2 depicts the construction of the EFPN. We represent the feature maps that share an identical semantic level by C_i/P_i but with the high-resolution levels as $C_i^{'}/P_i^{'}$. In the bottom extension in the EFPN that contains a feature texture transfer (FTT) model, the third and fourth pyramid layers of the EFPN are represented by green color and yellow color respectively are involved to produce the intermediate feature $P_3^{'}$ with selected regional data that is represented by the color blue. Then, the intermediate feature $P_3^{'}$ merged with a high-resolution feature map $C_2^{'}$, constructing the final layer $P_2^{'}$. This process can be defined as:

$$P_2^{'}=P_3^{'}\uparrow_{(2\times)}+C_2^{'} \tag{1}$$

where $\uparrow_{(2\times)}$ represents double-upscaling with the nearest-neighbor approach.

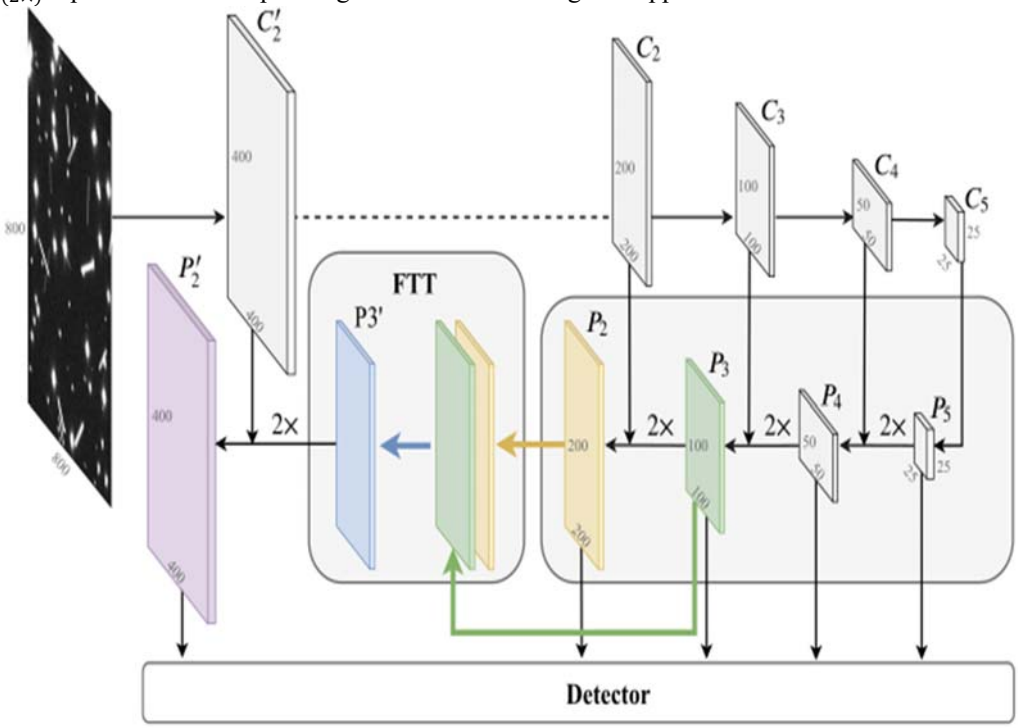

Figure 2. The overview of EFPN [7].

3.2. Feature Texture Transfer

The FTT model [7], which is based on image reference-based SR, is made to synchronously super-resolve features and then extracting regional textures from the reference features. In figure 3, the feature map P_3 from the third layer of the EFPN model serves as the main input for the FTT model, and the feature map P_2 from the fourth layer serves as the reference. The output $P_3^{'}$ can be associated with:

$$P_3^{'}=EX_t\left(P_2\|EX_c(P_3)\uparrow_{(2\times)}\right)+EX_c(P_3)\uparrow_{(2\times)} \tag{2}$$

where $EX_t(.)$ represents a texture-extractor component, $EX_c(.)$ represents a content-extractor component, $\uparrow_{(2\times)}$ represents double-upscaling with the sub-pixel CNN, and $\|$ represents the feature concatenation.

Both the texture-extractor block and the content-extractor block are built from residual blocks. Within the reference layer, The texture-extractor block receives the super-resolved content feature P_3 and the reference feature P_2. Within the main layer, the sub-pixel CNN is applied to upscale the content features' spatial resolution from the main layer P_3 while taking efficiency into account. The output feature map $P_3^{'}$ combines both the semantic and the regional data from the input feature P_3 and the reference feature P_2.

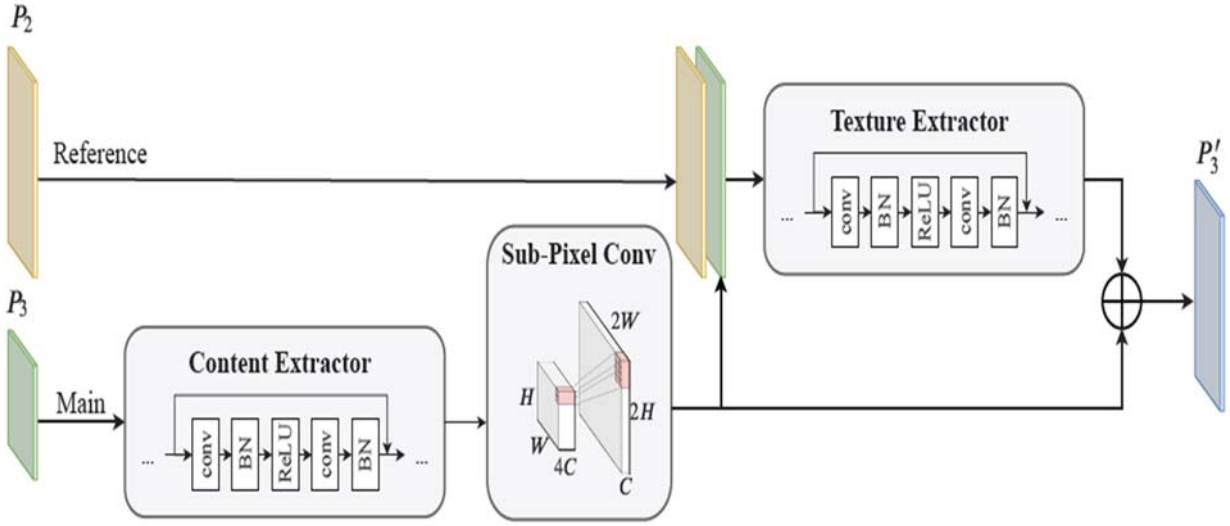

Figure 3. The overview of FTT [7].

3.3. Region Proposal Network

The RPN [4] is a CNN that predicts objectness scores and object bounds at each position. RPN takes an astronomical image and generates several rectangular region proposals (RPs). The network slides over the convolutional feature map and takes as input the spatial window with a size (m × m) to generate RPs. At each sliding-window location, it predicts several RPs, and the maximum number of RPs that can be obtained for each location is represented as k. Each sliding-window produces a low-dimensional vector that is fed into 2 fully connected layers (FC), the first layer is the box regression layer (reg) and the second layer is the box classification layer (cls). The reg layer has ~ 4k outputs, while the cls layer has ~ 2k outputs for each RP.

3.4. Region of Interest Alignment

The ROI alignment [4] converts RPs that have distinct sizes into RPs of the same size through RPs and their matching feature maps. Then feature maps of resized RPs sent to the FC layer for classification. To accurately maintain floating point boundaries, it iterates over each RP and divides each RP into (p × p) parts. A bilinear interpolation technique is used to compute the values of 4 fixed coordinate positions in each part. It estimates the values of these 4 positions based on the distance between recognizable coordinate positions, and the values of theirs. After that, the maximum pooling process is carried out to aggregate the result.

3.5. Classification and Regression

The feature maps of RPs are classified using a classification neural network. Also, RPs are sent into bounding box regression to rescale and shift it again to improve classification and regression accuracy. Finally, the final types of these RPs will be output. Following this, it determines the type and position of the candidate astronomical frames.

4. Experiments and Results

4.1. Datasets for training and testing

Because there aren't enough publicly available astronomical data sets, we use the simulation observation data [4] which is generated by using the Skymaker application [4]. Skymaker application usually is used to create simulation astronomical frames. The performance of the streak detection framework was tested on a simulated data set [4]. The astronomical data set includes a variety of astronomical targets, including streak-like sources and point-like sources. Streak-like sources are caused by moving targets during the capture, such as satellites, SD, asteroids, and meteors, while point-like sources such as stars, galaxies, supernovae, etc.

Figure 4 shows examples of sample images from the data set. It contains 2500 astronomical images with an image size of (800 × 800) pixels that are divided into 2000 frames for training and 500 frames for testing. Before sending the training set into the frameworks, we apply data augmentation by randomly rotating the images to create about 8000 frames to raise the quantity of training set images and the generalization ability of the frameworks. A rectangular box known as the ground truth box is used in the original images to label the locations

and kinds of celestial objects. For each celestial object i, the ground truth box of this object has data kept in a 4-dimensions vector: X_{imax}, X_{imin}, Y_{imax}, and Y_{imin}. X_{imax} and X_{imin} represent the celestial object's maximum and minimum values along the X direction, while Y_{imax} and Y_{imin} represent the celestial object's maximum and minimum values along the Y direction.

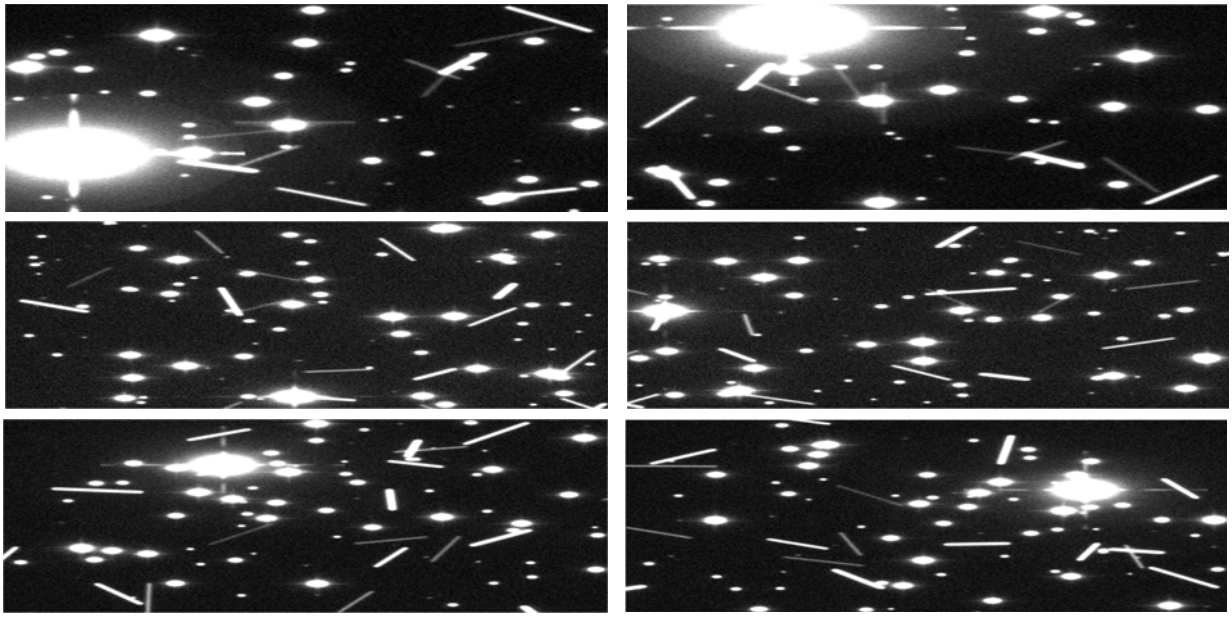

Figure 4. Examples of an astronomical image dataset.

4.2. Details of the experiments

The neural network is initialized with random weights and trained for 30 epochs using the Adam algorithm as the optimization. The starting learning rate is set to 0.00003 with the warming-up technique. For the classification neural network's loss function, the CrossEntropy function defined in Equation (3) is used and the smooth L1 loss function defined in Equation (4) for the bounding box regression.

$$L_H(p_i,y_i)=-\sum_{k=1}^{d}\left[p_{ik}\log_{y_{ik}}+(1-p_{ik})\log_{(1-y_{ik})}\right] \tag{3}$$

where p_{ik} represent the probability for celestial frames of type y_i.

$$L1_{loss}(a,b)=\frac{1}{n}\sum_i C_i \tag{4}$$

where

$$C_i=\begin{cases}0.5(a_i-b_i)^2, & \text{if } \|a_i-b_i\|<1\\ \|a_i-b_i\|-0.5, & \text{otherwise.}\end{cases} \tag{5}$$

4.3. Performance evaluation

Streak detection performance is quantified using mAP, recall, precision, and F1-score which are in turn computed from the counts of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

4.3.1. mAP: The average accuracy of all categories and it can be defined as:

$$mAP=\frac{\sum_{q=1}^{Q}AP(q)_{11\ points}}{Q} \tag{6}$$

where Q denotes the class number. And $AP(q)_{11\ points}$ is associated with:

$$AP(q)_{11\ points}=\frac{1}{11}\sum_{r=0}^{1.0}P_{interp}(r) \tag{7}$$

where $AP(q)_{11\ points}$ is a sum of the maximum precision P_{interp} (r) for eleven distinct recall numbers where r =[from 0 to 1] increment by 0.1.

4.3.2. Recall: The ratio of true streaks that are correctly detected as streaks and is associated with:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

4.3.3. Precision: The ratio of detected objects that are true streaks and is associated with:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

4.3.4. F1-score: The harmonic mean of the recall and the precision and is associated with:

$$\text{F1} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{10}$$

4.4. Competitive approaches

The performance of the EFPN-based framework [7] was compared with the FPN-based framework [4]. The performance of each framework was estimated by the metrics of mAP, precision, recall, and F1 score. We find that mAP is stable after 30 training epochs, which are used to train the two frameworks. After training, the two frameworks have been tested using the 500 astronomical images from the data set. Figure 5 displays the recall rate vs precision rate curve for each framework. The precision-recall curves calculate precision and recall over distinct thresholds, giving a demonstration of the framework's ability. It is shown that the EFPN-based framework [7] is better than the Faster R-CNN framework based on the FPN model [4].

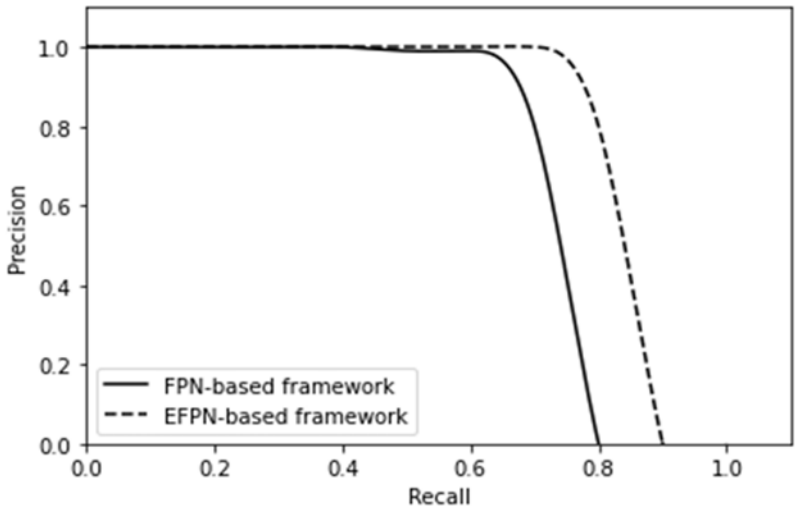

Figure 5. Precision–recall curve for each framework.

The results confirmed that the EFPN-based framework [7] achieves a significant improvement in streak detection. The resultant mAP, Recall, Precision, and F1-score values for the two frameworks are presented in Table I. It could be noticed that the EFPN-based framework [7] outperforms the FPN-based framework [4], in terms of mAP, recall, precision, and F1 score values by (3.8), (3.7), (5.8), and (5.1) respectively.

The significant improvement in precision indicates that the number of FP decreases and this leads to the ratio of detected objects that are true streaks increasing. Also, The significant improvement in recall indicates that the number of FN decreases and this leads to the ratio of true streaks that are correctly detected as streaks increase. This indicates that the ratio of missing streaks in astronomical images decreases and also the number of wrong streaks detected decrease. So the EFPN-based framework [7] demonstrates its ability in detecting streaks more precisely.

In Figure 6, we show the difference between the EFPN-based framework [7] and the FPN-based framework [4], we find that the EFPN-based framework [7] detects the faint streaks more precisely and also it increases the

detection ability of streaks that overlap with other streaks and streaks that overlap with other celestial objects. Also, the number of FP and FN decreased in the astronomical data set which was detected by the EFPN-based framework [7].

Table 1. The performance of the EFPN-based framework and FPN-based framework. Bold means the best performance for each metric.

framework	mAP	Recall	Precision	F1-score
Astro FRCNN w FPN [4]	71.1	85.8	75.3	80.0
Astro FRCNN w EFPN [7]	**74.9**	**89.5**	**81.1**	**85.1**

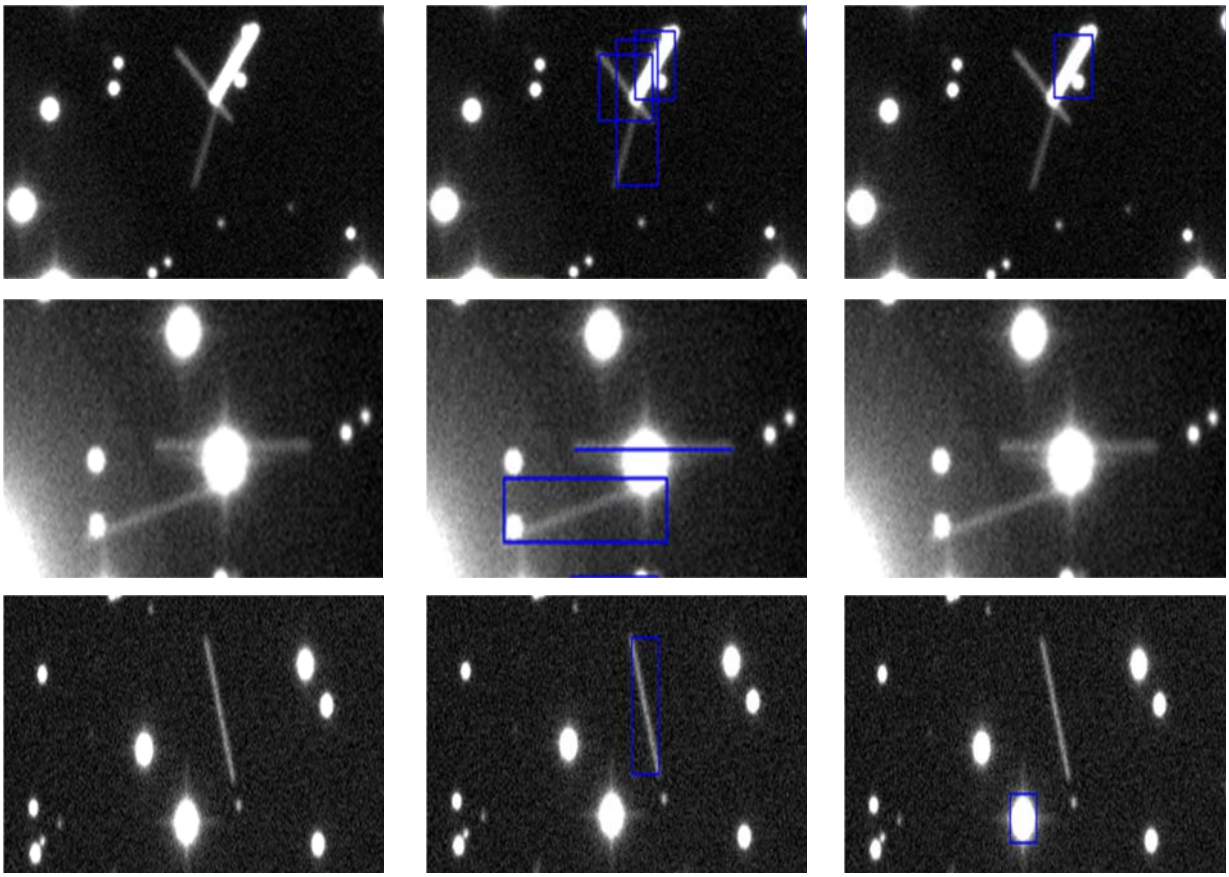

Figure 6. The left image in each row represents the original image, the central image in each row denotes the detection result of the EFPN-based framework [7], and the right image in each row denotes the detection result of the FPN-based framework [4].

5. Conclusion

In this paper, a comparison between the EFPN-based framework and the FPN-based framework shows that the EFPN-based framework is robust and has a good performance for streak detection in astronomical data sets. We use the simulated data set to train the neural network and compare the performance between the two frameworks by using mAP, recall, precision, and F1 score metrics. The EFPN-based framework achieves a significant improvement in streak detection than the Faster R-CNN framework based on the FPN model by 5.3% in terms of mAP, 4.3% in terms of recall, 7.7% in terms of precision, and 6.4% in terms of F1 score. The EFPN-based framework can detect faint streaks more precisely and also it increases the detection ability of streaks that overlap with other streaks and streaks that overlap with other celestial objects.

References

[1] Xi J, Xiang Y, Ersoy OK, Cong M, Wei X, Gu JJIA. Space debris detection using feature learning of candidate regions in optical image sequences. 2020;**8**:150864-77.
[2] Beckouche S, Starck J-L, Fadili JJA. Astronomical image denoising using dictionary learning. 2013;**556**:A132.
[3] Elhakiem AA, Ghoniemy TE, Salama GI. Astronomical image denoising based on Convolutional Neural Network. *2021 Tenth ICICIS*; 2021: IEEE.
[4] Jia P, Liu Q, Sun YJTAJ. Detection and classification of astronomical targets with deep neural networks in wide-field small aperture telescopes. 2020;**159**(5):212.
[5] Varela L, Boucheron L, Malone N, Spurlock N, editors. Streak detection in wide field of view images using Convolutional Neural Networks (CNNs). *Advanced Maui Optical and Space Surveillance Technologies Conference*; 2019.
[6] Duev DA, Mahabal A, Ye Q, Tirumala K, Belicki J, Dekany R, et al. DeepStreaks: identifying fast-moving objects in the Zwicky Transient Facility data with deep learning. 2019;**486**(3):4158-65.
[7] Deng C, Wang M, Liu L, Liu Y, Jiang YJIToM. Extended feature pyramid network for small object detection. 2021;**24**:1968-79.
[8] Ding S, Wang H, Chen D, Fu T, Gao M, editors. An improved method for dim space debris detection based on Hough transform. *2016 IEEE 13th ICSP*; 2016: IEEE.
[9] Diprima F, Santoni F, Piergentili F, Fortunato V, Abbattista C, Amoruso LJAA. Efficient and automatic image reduction framework for space debris detection based on GPU technology. 2018;**145**:332-41.
[10] Kong S, Zhou J, Ma WJIJoO. Effect analysis of optical masking algorithm for geo space debris detection. 2019;**2019**.
[11] Sun Q, Niu Z, Yao C, editors. Implementation of real-time detection algorithm for space debris based on multi-core DSP. *Journal of Physics: Conference Series*; 2019: IOP Publishing.
[12] Sease B, Flewelling B, editors. GEODETICA: A general software platform for processing continuous space-based imagery. *25th AAS/AIAA Space Flight Mechanics Meeting*; 2015.
[13] Mukhopadhyay P, Chaudhuri BBJPR. A survey of Hough Transform. 2015;**48**(3):993-1010.
[14] Krucinski J, Bienkowski A, Pattipati KR, editors. Machine Learning for Missile Streak Detection and Localization. *2021 IEEE Aerospace Conference (50100)*; 2021: IEEE.
[15] Tao J, Cao Y, Zhuang L, Zhang Z, Ding M, editors. Deep Convolutional Neural Network Based Small Space Debris Saliency Detection. *2019 25th ICAC*; 2019: IEEE.