Space Debris Detection Using Feature Learning of Candidate Regions in Optical Image Sequences

JIANGBO XI [ID][1], YAOBING XIANG[2], OKAN K. ERSOY[3], MING CONG[1], XIN WEI[4,5], AND JUNKAI GU[1]

[1]School of Geology Engineering and Geomatics, Chang'an University, Xi'an 710054, China
[2]School of Geography, Nanjing Normal University, Nanjing 210023, China
[3]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA
[4]Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences (CAS), Xi'an 710119, China
[5]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 10039, China

Corresponding authors: Jiangbo Xi (xijiangbo@chd.edu.cn) and Junkai Gu (dcgjk@chd.edu.cn)

ABSTRACT Space debris detection is important in space situation awareness and space asset protection. In this article, we propose a method to detect space debris using feature learning of candidate regions. The acquired optical image sequences are first processed to remove hot pixels and flicker noise, and the nonuniform background information is removed by the proposed one dimensional mean iteration method. Then, the feature learning of candidate regions (FLCR) method is proposed to extract the candidate regions and to detect space debris. The candidate regions of space debris are precisely extracted, and then classified by a trained deep learning network. The feature learning model is trained using a large number of simulated space debris with different signal to noise ratios (SNRs) and motion parameters, instead of using real space debris, which make it difficult to extract a sufficient number of real space debris with diverse parameters in optical image sequences. Finally, the candidate regions are precisely placed in the optical image sequences. The experiment is performed using the simulated data and acquired image sequences. The results show that the proposed method has good performance when estimating and removing background, and it can detect low SNR space debris with high detection probability.

INDEX TERMS Space debris detection, background estimation, candidate region extraction, deep learning.

I. INTRODUCTION

Space debris refers to useless artificial debris in orbit, which includes nonfunctional spacecraft, and abandoned space vehicle stages [1]. The amount of space debris is growing quickly, and its existence on a large scale poses a serious threat to satellites in orbit, space stations, and other space activities [2]. There was more than 139 million space debris by 2019 [3]. If space debris collides with spacecraft, it will cause equipment damage, mission failure, and so on. Therefore, the movement of space debris must be monitored to achieve an effective prediction of its activities and to avoid accidents. Space debris detection is a key step in monitoring space debris.

Space debris are mainly in geosynchronous orbit. The debris are far away from the sensor and the reflection area is small. When debris images appear through charge-coupled devices (CCD), the debris appears in the focal plane and spans only a few pixels. The total number of pixels covered by the debris is less than 0.15% of the 256×256 size image. At the same time, the reflectivity of the debris is low and its distribution is sparse. The corresponding energy of the pixels is very weak compared to the noise in the environment, and the SNR can even equal to 1. Therefore, according to the definition of SPIE, space debris consists of typical dim and small objects, and it is difficult to detect space debris with an extremely low SNR. A large number of classical methods have been

The associate editor coordinating the review of this manuscript and approving it for publication was Qichun Zhang [ID].

proposed to detect such dim small objects. Mohanty [4] proposed an algorithm to detect those objects based on the maximum likelihood ratio. Reed *et al.* [5] proposed a method that uses three-dimensional matched filtering to detect small objects. It can deal with the low SNR situation well through a group of filters to match the possible trajectories when the velocity and direction are known. Barniv [6] proposed a dynamic programming algorithm that uses speed and shape to detect objects moving in a straight line under low SNR, and this algorithm was subsequently developed by Buzzi *et al.* [7]. There are two main kinds of dim and small objects: streak-like, and point-like objects. They are acquired with different integration times. As the integration time becomes long compared with the velocity of the objects, the dim and small objects become increasingly streak-like from point-like. The current methods of space debris detection are manifested in the following works:

(1) Steak-like object detection: Kouprianov [8] proposed a point spread function (PSF) fitting method based on a model profile of an extended trail; Tagawa *et al.* [9] took the local sums of the streak along the moving direction to archive steak-like object detection; Virtanen *et al.* [10] used area-based grayscale intensity estimation and multiple-window shape analysis to implement an automated steak detection pipeline. The direction of the motion needs to be known as a priori knowledge or to be estimated with a complex fitting method. Blostein and Huang [11] proposed a multistage hypothesis testing (MHT) method, which can effectively search for dim and small objects at every pixel in the image sequence. Demos [12] proposed structured branch multistage hypothesis testing (SB-MHT) using sizable computational savings. Blostein and Richardson [13] proposed multiple multistage hypothesis test tracking (MMHTT) to detect dim and small objects. These methods always have an excellent performance, but usually need to search along a route or directions, which is also demanding substantial time and computing resources.

(2) Point-like object detection: Yanagisawa *et al.* [14] extracted low SNR objects using multiple, consecutive images of objects and median filtering based on the a priori motion information of the objects, which is very simple but the detection probability still needs to be improved. Nunez *et al.* [15] proposed an image deconvolution method based on the Richardson-Lucy (R-L) algorithm, which is based on the maximum likelihood solution but did not reach the maximum. Sun and Zhao [16] and Sun *et al.* [17] implemented a pipeline to detect dim objects by using median filtering, mathematical morphology, and global thresholding. These methods also require prior information of the objects, and have difficulty to detect space debris without the prior information. In recent years, some scholars have proposed new methods to implement high efficiency and low computation. Xi *et al.* proposed the three frame correlation method [18], and used the particle filter method [19] to detect space debris which can detect and track space debris with median SNRs. Xi *et al.* [20] proposed an

effective method to detect space debris with low SNR in optic image sequences by using the time-index filtering and time-index multistage quasi-hypothesis-testing (TMQHT) method. Ding *et al.* [21] proposed a detection method by using the prior orbit information and the Hough line transform. It can effectively detect small size space debris in the low earth orbit. Metrailler *et al.* [22] proposed a difference method to detect moving space debris by processing two successive exposures. The method can detect faint debris streaks down to peak SNR2 in less than 1 second of execution time. Diprima *et al.* [23] proposed a pipeline for automatic near real-time detection of space debris in optical data by using GPU technology. It is based on morphological filtering andthe Hough transform for line detection. Yang *et al.* [24] proposed an integration algorithm for the detection of weak space debris using incoherent scatter radar. Kong *et al.* [25] proposed an optical masking algorithm for GEO space debris detection. Its detection probability reaches 94% and false-alarm rate is below 2% when the SNR is higher than 3. Sun *et al.* [26] implemented a real-time detection algorithm for space debris based on multicore DSP. It takes approximately 600ms to detect objects on a 2048*2048 (gray 16bit) images. These recently proposed methods usually have a good performance, but the computation load, depending on the prior information, and the detection probability can still be improved.

Machine learning methods can learn the extracted features from different types of data to perform regression and classification [27], [28]. Recently, researchers have tried to introduce machine learning method into different research areas such as doing cyclic capacity, or future prediction of lithium-ion batteries [29], [30]. For images, deep learning has gained great success in image and computer vision tasks. Early in 1998, LeCun *et al.* [31] successfully used a convolutional neural network to identify the characters in documents. Recently, deep learning (DL) has been widely applied in image classification tasks [32]–[34]. It has also been successfully extended to object detection including two-step and one-step models. Two-step object detection framework is based on region proposals, classification and bounding box regression. R-CNN [35] obtains candidate regions by a selective search, and uses the CNN model to obtain the classification results. Fast R-CNN [36] uses convolution layers to produce feature maps, and obtains the projection of regions of interest (ROI). Faster R-CNN [37] uses region proposal a network (RPN) with 9 different anchors to obtain the candidate regions, and to speed up the process of object detection. The one steps object detection framework is based on global classification and bounding box regression. YOLO [38] uses the topmost features to predict both classes and obtains the bounding boxes by splitting the whole image into $S \times S$ grids. SSD [39] is proposed using a set of anchor boxes with different aspect ratios and scales to detect objects with different sizes and motion directions. Recently, researchers began to use image processing methods with deep learning models to detect dim and small objects. Hu *et al.* [40]

proposed a method using the bilateral filtering method and a long short term memory (LSTM) neural network to detect and track dim and small objects in infrared sequential images. Varela *et al.* [41] presented a streak detection method based on YOLO. They demonstrated that this method is better than the Hough transform and phase congruency transform. Kong *et al.* [25] proposed an effect analysis of the optical masking (EAOM) algorithm, which can detect GEO space debris with a low SNR based on a top-hat transformation, masking technique, and weighted algorithm. Jia *et al.* [42] built an astronomical object detection and classification pipeline based on the Faster R-CNN, ResNet-50 [43] and Feature Pyramid Network [44]. These frameworks have a strong ability to detect dim small targets with high accuracy. However, none of these frameworks combined the advantage of traditional methods and deep learning. It is challenging to use deep learning for space debris detection. The reasons are as follows: (1) Space debris is a dim and small object. The intensity of space debris is similar to that of the background noise, and space debris only have a few spatial features. (2) There are a large number of stars in the astronomical image sequences, and their energy distribution and patterns are very similar to those of space debris. Therefore, it is difficult to use deep learning models to classify or detect space debris directly.

In this article, we propose a space debris detection method using feature learning of candidate regions (FLCR) in optical image sequences. The framework is shown in Figure 1. First, image preprocessing methods are proposed to remove hot pixels, flicker noise, and nonuniform background information. Second, the space debris are detected using feature learning of candidate regions, which includes two parts: one is using the motion features among multiple sequential images and an improved threshold segmentation to reduce the amount of data and extract the candidate regions of the space debris, and the other is using the deep learning model SDdecNet to perform classification with the candidate regions. Finally, the locations of the detected space debris in the sequential images are computed using the weighted centroid localization. Similar to the recently proposed dim and small object detection method, which mainly uses traditional image processing or machine learning methods, the above method has difficulties when using the spatial features of dim and small objects sufficiently. The proposed FLCR method can greatly reduce a large number of false alarms caused by stars and noise, and reduce the computing load using the candidate regions. Additional, it can detect space debris more precisely and quickly by learning the spatial features of the extracted candidate regions without using exhaustive searching method to confirm the candidate objects.

The paper is organized as follows: Section 2 discusses the image preprocessing methods. Space debris detection using feature learning of candidate regions is described in Section 3. The experiments and results are given in Section 4. In Section 5, the conclusions are provided.

II. IMAGE PREPROCESSING
A. HOTPIXELS AND FLICKER NOISE REMOVAL
The image background on a charge-couple device (CCD) mainly includes moving targets, stars, starry sky background, flicker noise, dark current noise and damaged pixels. Visible light background images are subject to unevenness due to the different gray levels of the different channels of the camera. Therefore, we need to preprocess the image, to reduce the negative impact of background and noise on object detection.

Damaged CCD pixels due to space radiation and other reasons generally affect single pixels. The background is usually distributed unevenly; therefore, we can use local thresholds to detect the hot pixels in the image, and use the average of the gray values of the surrounding four neighboring pixels to replace it. The method is shown as follows:

1. Perform template matching detection. When the difference between gray value A and the gray value of the surrounding 4 neighborhoods is greater than a certain threshold, the pixel is initially determined as a thermal pixel, and the average gray value $m1$ of the four neighborhood pixels is calculated.

2. Within a certain range of surroundings, select four windows, calculate the average gray value of the pixels in each window, and then calculate the average gray value of the pixels in all windows to obtain the local average $m2$.

3. If $|m1 - m2| < b$, (b can be easily obtained by checking the mean uniform background using the acquired images, which is related to the optical system design, and the integrating time of the optical system.), this pixel is judged as a thermal pixel, and the pixel is replaced with the average gray value of the surrounding 4 neighborhood pixels. Otherwise, it is judged as a star and not a thermal pixel.

The flicker noise in the image appears as a single bright spot in the image, with a size of one or two pixels (flicker noise can occasionally appears, which can still be detected and eliminated because the mean of 4 pixels is computed to compare with the threshold), and its appearance is similar to that of a thermal pixel. Flicker noise distinct from thermal pixels because its position appears random and its position changes with time. In the experiment, it is processed using the same method that is applied to the thermal pixels.

B. NONUNIFORM BACKGROUND ESTIMATION AND REMOVAL
As the stray light from a celestial body enters the image, the system needs to estimate and subtract the nonuniform background. At the same time, for the original observation image, because the CCD sensor is actually composed of two signal transmission channels, the images collected by the two channels have a certain degree of nonuniformity. The Gaussian mixture model (GMM) is usually a good method for data modeling, but it is not suitable for this nonuniform background estimation. We propose a one-dimensional mean iteration method to estimate the multichannel nonuniform background, and to be remove the background from the
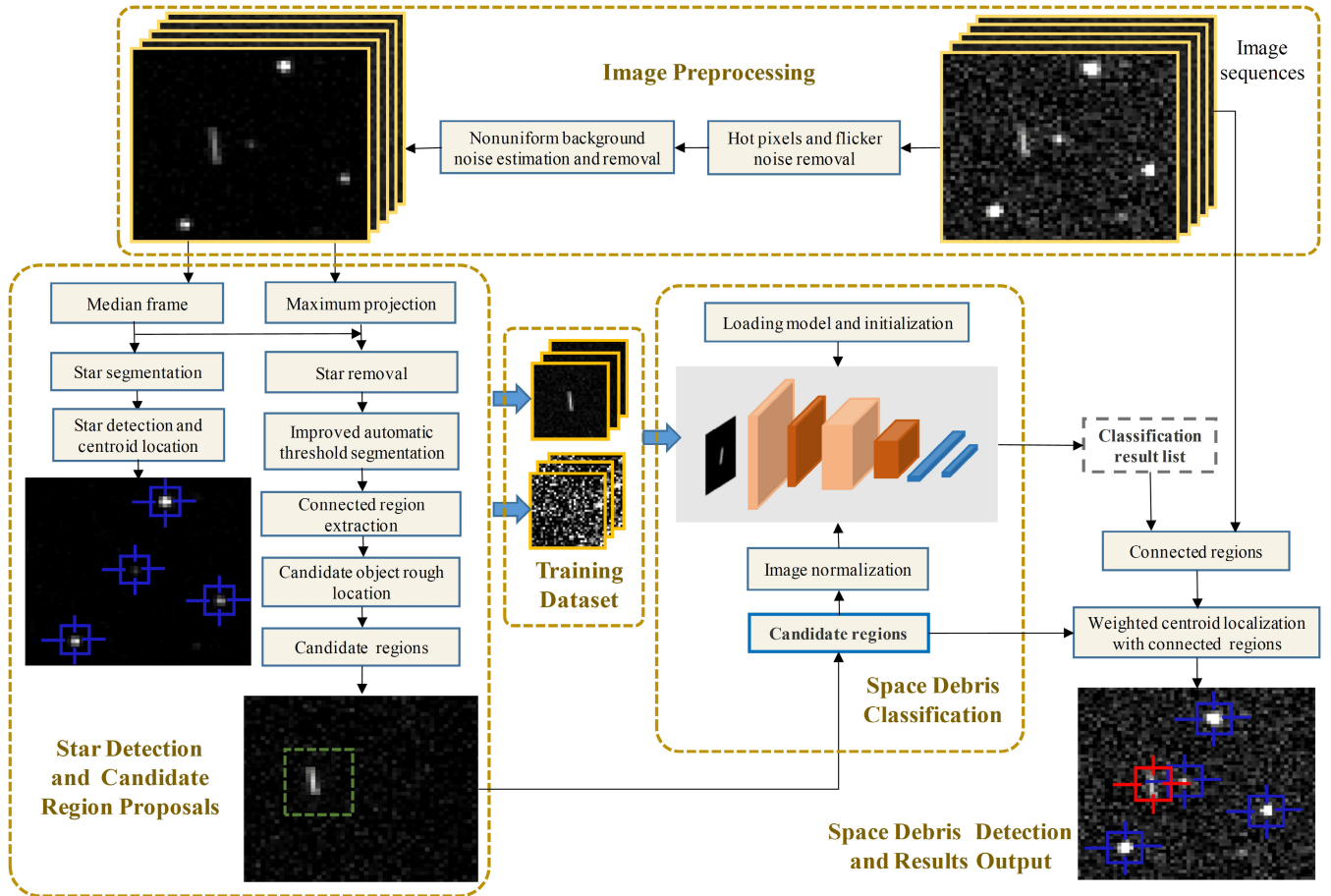
FIGURE 1. Diagram of space debris detection. First, the optical image sequences are preprocessed to remove hot pixels and flicker noise and to remove the nonuniform background. Then, stars are detected and removed, and the candidate regions of the space debris are extracted. These regions are classified by a trained deep learning model using a large number of simulated space debris with different SNRs and motion parameters, instead of using real space debris. The advantage is that the model does not need to extract a sufficient number of real space debris with diverse parameters in the optical image sequences. Finally, the correctly classified space debris are located precisely in the optical image sequences, and they are output together with the detected stars.

images, which is important for space debris detection when using threshold segmentation. Suppose the original image is $f_0(i, j)$, and the initial window size is W_0. The window size of the mth iteration is

$$W_m = \left[\frac{W_0}{m}\right] + C_0 \qquad (1)$$

where $[\cdot]$ denotes rounding the number, and C_0 is a constant added to ensure that the size of the window is not reduced to zero as the iteration is performed.

The background estimation for the first iteration is

$$\hat{B}_1(i, j) = \frac{1}{M} \sum_{l=j-\frac{W_1-1}{2}}^{l=j+\frac{N_1-1}{2}} f_0(i, l) \qquad (2)$$

Before the next iteration, the energy of the stars is reduced by

$$f_m = \min\left(f_{m-1}, \hat{B}_m + 2\sigma_n\right) \qquad (3)$$

where σ_n is the standard deviation of the image noise.

The background estimation after the mth iteration is

$$\hat{B}_m(i, j) = \frac{1}{W_m} \sum_{l=j-\frac{W_m-1}{2}}^{l=j+\frac{W_m-1}{2}} f_{m-1}(i, j) \qquad (4)$$

For the image with two channels, the background estimation is

$$\hat{B} = \hat{B}_L + \hat{B}_R \qquad (5)$$

where, $\hat{B}_L$, and $\hat{B}_R$ are the background estimations of the left and right channels of the image, respectively.

III. SPACE DEBRIS DETECTION WITH FEATURE LEARNING OF CANDIDATE REGIONS USING DEEP LEARNING

The detailed space debris detection flow chart is shown in Figure 2. This section includes three subsections. The first subsection describes the proposed method for extracting candidate regions of space debris including the star detection and removal. The second subsection presents the SDdecNet
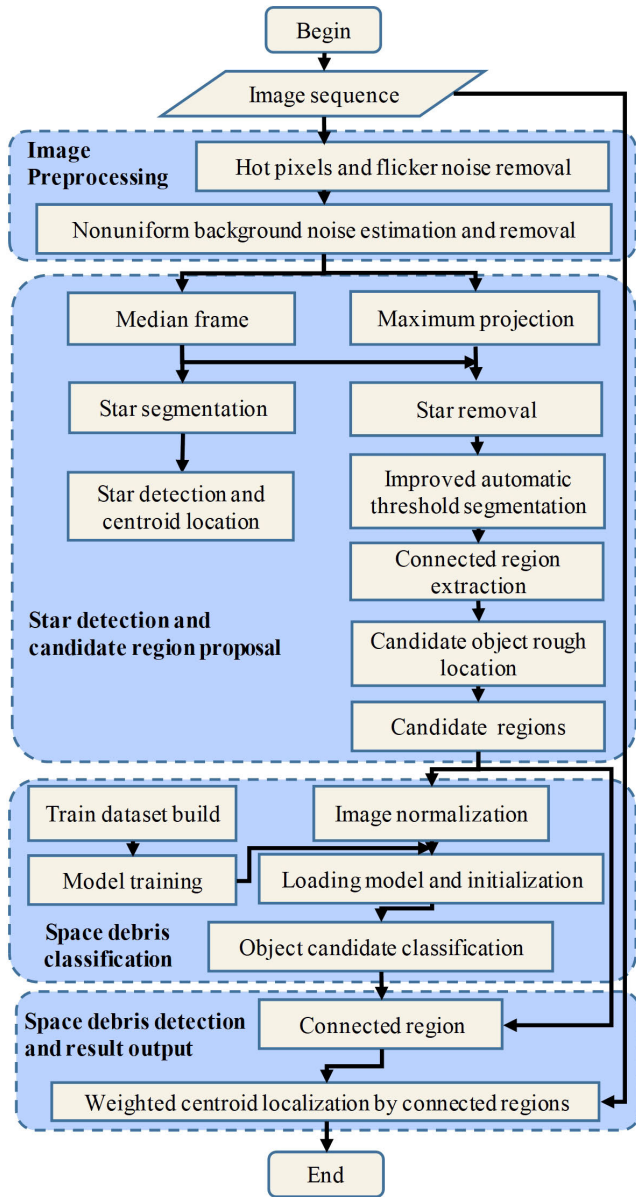
FIGURE 2. The flowchart of space debris detection.

for space debris classification, which is trained using a large number of simulated space debris images. It can effectively overcome the difficulty of segmenting enough real space debris images with different SNRs and motion parameters. Finally, the candidate regions of space debris are classified and located precisely in the optical image sequences using the subpixel location method.

A. CANDIDATE REGION ESTIMATION

The energy distribution of the stars is very similar to that of space debris without motion blur. Therefore, the stars will be removed first to reduce the number of candidate region estimation of space debris. A maximum projection is used to reduce the data from three dimensions to two dimensions, and the median frame is used to perform star detection. The

maximum and median frames are computed as

$$f_{max}(i,j) = \max\left[\hat{f}(i,j,k), k \in [1,K]\right], \quad (6)$$

$$f_{med}(i,j) = median\left[\hat{f}(i,j,k), k \in [1,K]\right], \quad (7)$$

where K is the number of images in an image sequence, and $K = 5$ is chosen to assure that the dim and small space debris are detected stably and effectively. The maximum frame includes both the space debris and stars, and the median frame only contains stars. When the maximum frame is generated, the time index image is obtained simultaneously as

$$t(i,j) = \arg\min_{k\in[1,K]} f_{max}(i,j) = \sum_{k=1}^{K} t_0(i,j,k), \quad (8)$$

where $t(i,j)$ is the time index that records the frame numbers of the maximum intensity at each pixel, and they can be rewritten as a summation of the time indices of each frame in an image sequence.

Most of the star intensity can be removed by

$$f_{res}(i,j) = f_{max}(i,j) - \alpha f_{med}(i,j), \quad (9)$$

where α is the intensity enhancement factor used to reduce the intensity of bright stars to a large extent [20].

The intensity of the space debris is left in the residual frame $f_{res}(i,j)$. Therefore, it can be used to extract the candidate regions of space debris, and $f_{med}(i,j)$. Then, the improved adaptive threshold method is used to extract the candidate regions of space debris, and detect stars.

First, the adaptive threshold method is used to compute the threshold of the image as

$$T_{rough} = \mu_0 + \beta_0 \sigma_0, \quad (10)$$

where μ_0 and σ_0 are the mean value and standard deviation of the image, and β_0 is a coefficient, which is usually chosen from 3 to 5 [45] to assure that a large number of noise pixels are eliminated according to the property of Gaussian distribution. With the rough threshold and supposing the image here is $\hat{f}(i,j)$, a corresponding binary mask is generated as

$$b(i,j) = \begin{cases} 1, & \hat{f}(i,j) \geq T_{rough} \\ 0, & \hat{f}(i,j) < T_{rough}. \end{cases} \quad (11)$$

The exclusive mask is obtained by

$$e(i,j) = 1 - b(i,j). \quad (12)$$

Then, the precise threshold can be computed in the image as

$$\hat{f}_{noise}(i,j) = \hat{f}(i,j)\, e(i,j). \quad (13)$$

The precise threshold is obtained by

$$T_{precise} = \mu_1 + \beta_1 \sigma_1, \quad (14)$$

where μ_0 and σ_0 are the mean value and standard deviation of the image $\hat{f}_{noise}(i,j)$, and β_1 is a coefficient, which is usually chosen from 3 to 5.

With the improved adaptive threshold, the precise thresholds and corresponding binary masks used to segment the foreground and background can be obtained for both $f_{max}(i,j)$ and $f_{med}(i,j)$, respectively, and they are denoted as T_{max}, T_{med}, $b_{max}(i,j)$, and $b_{med}(i,j)$.

The candidate regions can be extracted using a mask that only contains the candidate objects of the current frame. It is generated using the time index image as follows:

$$b_{res}(i,j,k) = \begin{cases} 1, & t(i,j) = k \\ 0, & t(i,j) \neq k. \end{cases} \quad (15)$$

The image that only contains the candidate regions of space debris in the current frame is

$$f_{CR}(i,j,k) = f_{residual}(i,j)\, b_{res}(i,j,k). \quad (16)$$

The candidate regions can be extracted and located through the following steps:

1. Binary morphological filtering. Through the combination of binary morphological corrosion and expansion operations, the isolated noise points in the binary image are filtered.

2. Connected region extraction. Extract the connected region from the filtered binary image. Each connected area corresponds to an object. We use the extracted connected domain to mask the original image. The pixels of the candidate region are extracted and arranged in order to form an $n \times 3$ vector. The three dimensions store pixel gray value F, abscissa i, and ordinate j.

3. Rough candidate centroid calculation: To simplify the calculation and ensure sufficient positioning accuracy, we use the vector obtained in step 2 to calculate the target centroid position using the weighted centroid method:

$$x_0 = \lfloor \frac{\sum_{k=1}^{n} i_k}{n} \rfloor$$
$$y_0 = \lfloor \frac{\sum_{k=1}^{n} j_k}{n} \rfloor \quad (17)$$

where i is the abscissa of the pixel, j is the ordinate of the pixel, n is the total number of pixels in the target mask area, and (x_0, y_0) is the coordinate position integer of the object.

4. Clip the subimage from the clear image: We extend 16 pixels horizontally and vertically from the target position, and select a 33×33 subimage as the candidate region.

B. DETECTION OF SPACE DEBRIS USING SPACE DEBRIS DETECTION NEURAL NETWORK (SDdecNet)

1) ARCHITECTURE OF SDdecNet

The space debris recognition network is a convolutional neural network model based on the LeNet5 [31] architecture, and is used to identify space debris. After the candidate regions first extracted from the image sequences. A large number of false alarms caused by stars can be eliminated. These candidate regions are then fed into the proposed SDdecNet to learn the features of space debris efficiently and automatically. The samples in the space debris detection task only include space debris and noise. Their features mainly include low level
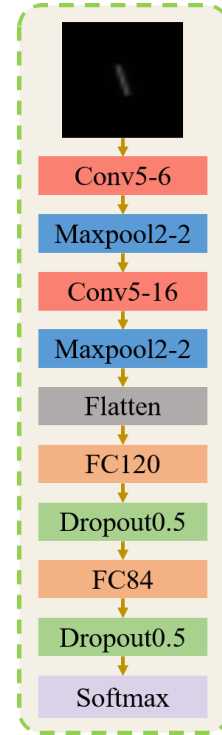

FIGURE 3. Architecture of the SDdecNet.

features such as edges and points. Two convolutional layers are sufficient for this task. The network model structure is shown in Figure 3. The input size of the network is $33 \times 33 \times 1$. The first convolution layer has 6 convolution kernels with size 5×5. The convolution process uses zero padding to keep the size of the output unchanged. The rectified linear units (ReLU) are chosen as the activation function to effectively avoid the vanishing gradient problem [46]. This function also has balanced performance on computation cost and accuracy. A pooling layer that uses a 2×2 pooling kernel is connected later, and the data are pooled in steps of 2×2. The second convolution layer has $16 \times 5 \times 5$ convolution kernels with zero padding to keep the size of the output unchanged. The activation function is ReLU. A pooling layer using a 2×2 pooling kernel is connected after the convolution layer, and the data are pooled in steps of 2×2. The convolution layer is followed by a fully connected neural network classifier. First, a two dimensional matrix is converted into a one-dimensional matrix using a flattening layer. Later, a hidden layer containing 120 neurons is used, and the activation function is ReLU. Then a hidden layer containing 84 neurons is used, and the activation function is also ReLU.

Dropout regularization is used for each fully connected layer. It randomly turns off 50% neurons on the fully connected layers during the training processing. This helps prevent the model from being overfit and improve the performance of the model. The output layer uses Softmax as the activation function. Since the data is divided into space debris and stars, two output neurons are used.

2) TRAINING OF SDdecNet USING SIMULATED SPACE DEBRIS

Although there is a large amount of space debris, only a few debris can be observed in a single optical image. It is difficult to obtain sufficient instances of space debris to train the SDdecNet. The simulated space debris is introduced to train the SDdecNet. The intensity distribution of the debris without motion blur is

$$I(x, y) = I_0 \left(\frac{2J_1 \left(\sqrt{x^2 + y^2} \right)}{\sqrt{x^2 + y^2}} \right)^2 \qquad (18)$$

where I_0 is the central intensity of the debris, and J_1 is the 1st order Bessel function. Supposing the motion is in the x direction, then the point spread function of motion blur is:

$$PSF_{mx}(x, y) = rect \left(\frac{x}{d_{mx}} \right) \delta(y) \qquad (19)$$

$$rect \left(\frac{x}{d_{mx}} \right) = \begin{cases} 1 & |x| < \frac{d_{mx}}{2} \\ \frac{1}{2} & |x| = \frac{d_{mx}}{2} \\ 0 & |x| > \frac{d_{mx}}{2} \end{cases} \qquad (20)$$

where d_{mx} is the motion distance in the x direction on the focal plane, and the motion in the y direction is the same as that in the x direction, which is denoted as $rect \left(\frac{x}{d_{my}} \right)$. The noise is simplified to Gaussian noise $N(x, y)$ with background mean u_1 and σ_1, which are computed using the improved adaptive threshold. Therefore, the simulated space debris can be written as

$$S_{sim}(x, y) = I(x, y) * PSF_m(x, y) + N(x, y) \qquad (21)$$

$$PSF_m(x, y) = PSF_{mx}(x, y) * PSF_{my}(x, y) \qquad (22)$$

After the discrete sampling of the intensity of the simulated space debris with background noise, the training samples can be obtained.

3) SUBPIXEL LOCATION OF SPACE DEBRIS AND STARS

The subpixel location of space debris and stars can be divided into two steps:

1. The connective regions of classified space debris, and stars extracted from the median frame image are obtained.

2. Subpixel object centroid calculation: To simplify the calculation and ensure sufficient position accuracy, we use the vector obtained in the candidate region estimation to calculate centroid position of the object subpixel using the weighted centroid method:

$$x_1 = \frac{\sum_{k=1}^{n} F^2(i_k, j_k) i_k}{\sum_{k=1}^{n} F^2(i_k, j_k)}$$

$$y_1 = \frac{\sum_{k=1}^{n} F^2(i_k, j_k) j_k}{\sum_{k=1}^{n} F^2(i_k, j_k)} \qquad (23)$$

where i is the abscissa of the pixel, j is the ordinate of the pixel, and $F(i, j)$ is the gray value at (i, j). n is the total number of pixels in the target mask area. (x_1, y_1) is the coordinate position of the space debris centroid.

The basic location method for the location of stars, and dim and small objects mainly utilizes (1) the interpolation method and (2) the fitting method. The basic location method for interpolation is the centroid method, also known as center of mass method. In additional, the threshold centroid method and weight centroid method are also commonly used [47]. The fitting method includes the Gaussian surface fitting method and paraboloid fitting [48]. Alexander analyzed the systematic error of the subpixel location in the frequency domain [49], and Xi *et al.* used a phase transfer function to compute the location of the stars in the frequency domain [50]. The fitting method and frequency method usually have good location accuracy, but their computing requirement are demanding. The centroid method is very simple, but the performance still needs to be improved. Considering the balance between location accuracy and computing speed, the weighted centroid is the proper method.

4) COMPUTATIONAL EFFORTS OF FLCR

The computational efforts of our designed feature learning methods mainly include three measures. The first measure uses maximum projection, which reduces the three-dimensional image sequences into two-dimensional image. The second measure uses the candidate region extraction method based on the improved adaptive threshold method, which eliminates most of the false alarms mainly caused by stars. Therefore, the number of these candidate regions is highly reduced, as is the computational load. The third measure uses SDdecNet to recognize space debris instead of using the exhaustive searching method to conform the detection results of candidate space debris. The chosen threshold depends on the requirement of the detection tasks by setting β_0 and β_1.

IV. EXPERIMENT

The experimental hardware was an Intel CPU i7-8700K, and the CPU memory was 32 GB.

A. SIMULATION OF SPACE DEBRIS AND STARS FOR LEARNING

Space debris appears as a series of point-like or streak-like objects depending on the different exposure times in the image sequences. The simulated region of the space debris is set to 33×33 pixels, and its motion direction is random from $(-45, 45)$. The moving direction is from the bottom to the top on the images, which is based on the motion properties of real space debris. The summation of all the gray levels of the space debris is used with different SNRs to generate simulated space debris, and the energy concentration is 80% in the 2×2 window. The center offset is a random number between 0 and 1 with an interval of 0.1, which indicates that the centers of the space debris are at different pixel location. The intensities are distributed based on the above settings. It is difficult to acquire a large number of real space debris with

different SNRs along different motion directions.Therefore, we need to generate the simulated space debris to train the detection network sufficiently. Considering the background of the images, a Gaussian distribution with a mean equal to 313 and a standard deviation equal to 7.5 (which is computed with the acquired real sequential images) is generated as a simplified background for the simulated space debris. The negative background samples were collected directly from the optical image sequences.

1) TRAINING SET

The training set is composed of high SNR space debris samples, low SNR space debris samples and background samples.

High SNR space debris samples: The magnitude level ranges are from M8 to M14 and the corresponding total digital number value are 249530, 99339, 39548, 15744, 6268, 2495 and 993. The center of the space debris are changed with a small random offset with an interval of 0.1 from 0 to 1 in the x and y directions. A total of 2000 space debris samples of each level are generated and the dataset totally includes 14000 high SNR space debris samples.

Low SNR space debris samples: The SNR of space debris is defined as follows:

$$\text{SNR} = \frac{\frac{S}{n_1 \times n_2}}{\sqrt{\frac{S}{n_1 \times n_2} + B}} \qquad (24)$$

where S is the total intensity of the space debris; n_1 and n_2 are the width and the height of the window, respectively; and B is the background mean. The space debris with $SNR = 6$, 4, 3, 2.5, 2.0, 1.5, 1.3, 1.2, 1.0, 0.8, 0.5, 0.4, 0.3 are selected, and for each SNR, 2000 space debris samples are generated. Finally, a low SNR space debris dataset of 26000 samples is generated. The space debris with different magnitudes or SNRs are shown in Figure 4.

Background samples: It is much easier to obtain a large number of training samples of background regions than those of space debris. Therefore, 10000 background samples were clipped from the image sequences directly instead of using simulation. Then, data augmentation was used, and each sample was flipped horizontally, vertically and in both directions. Finally, we obtain 40000 noise background samples, some of which are shown in Figure 5. The total training set includes 80000 samples.

2) TEST SET

There are a group of test sets with different SNRs to evaluate the classification performance. Each test set has 2000 samples composed of 1000 samples of space debris and 1000 samples of background.

a: SPACE DEBRIS SAMPLES

Space debris samples with different SNRs were generated including 6, 4, 3, 2.5, 2.0, 1.5, 1.3, 1.2, 1.0, 0.8, 0.5, 0.4, and 0.3. There are 1000 test samples of space debris for each value of the SNR.
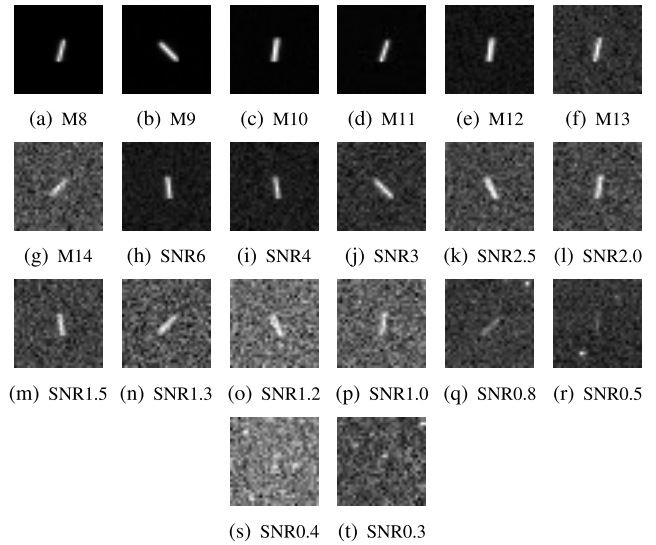

FIGURE 4. The simulated dataset with different magnitudes and SNRs of space debris. M denotes the magnitude. For better visibility, all displayed samples have been stretched with contrast.
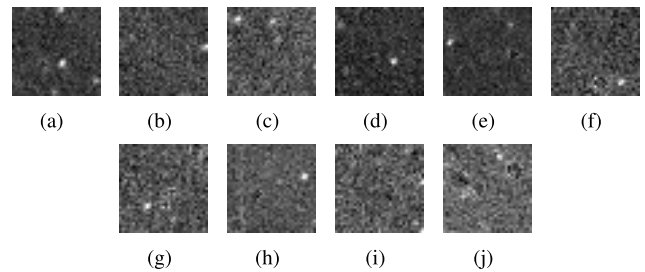

FIGURE 5. The dataset of background noise obtained from the real image. For better visibility, all displayed samples have been stretched with contrast.
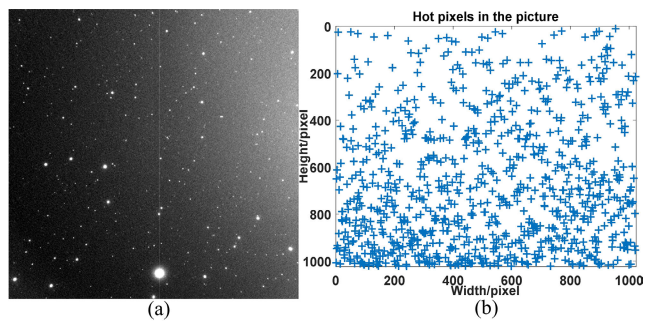

FIGURE 6. Original image and hot pixels detection results. (a) Original image, (b) Hot pixel.

b: BACKGROUND SAMPLES

A total of 1000 background samples were randomly clipped from the image sequences and they never appeared in the training set. These same 1000 background samples have different value of SNR.

B. IMAGE PREPROCESSING

The real astronomical image with a size of 1024 × 1024 pixels used in this experiment is shown in Figure 6 (a).
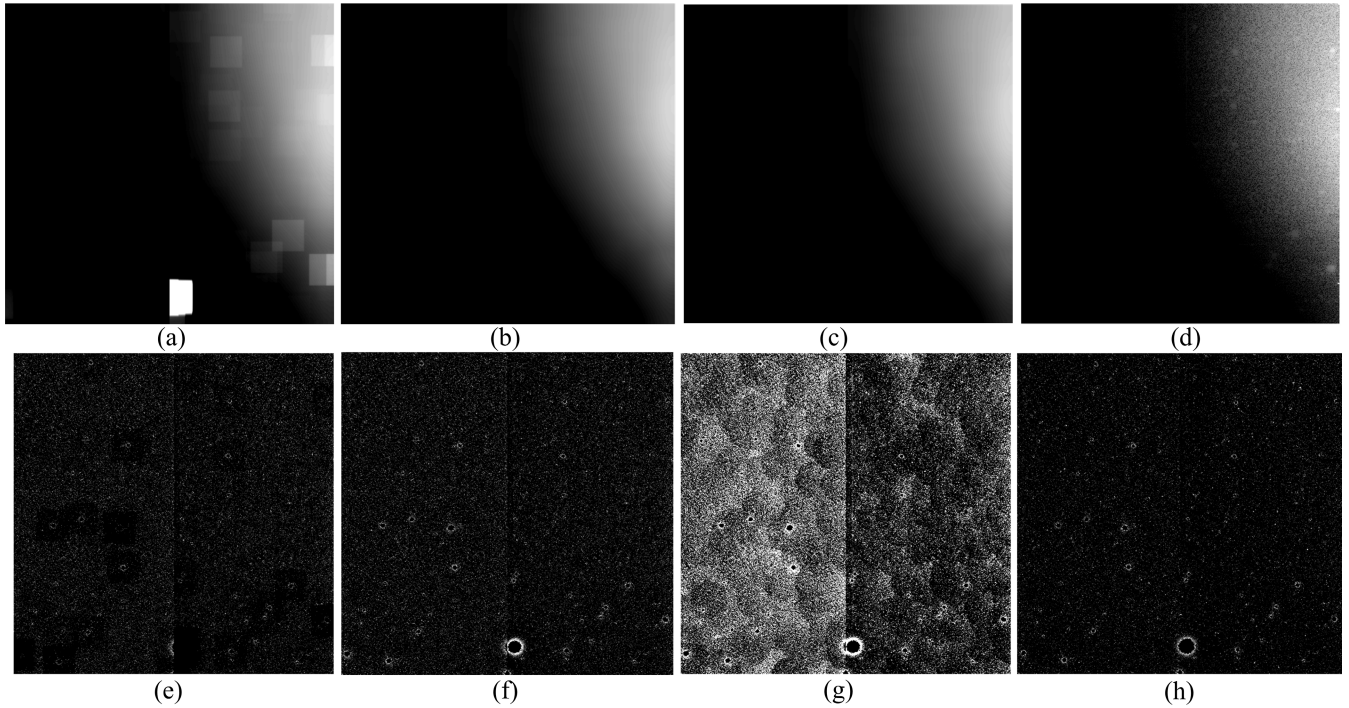
FIGURE 7. Results of background estimation and background residual errors. (a)-(d) are the estimated background images using the mean, median, top-hat, and proposed 1D mean iteration methods, respectively. (e)-(h) are the residual errors after using the mean, median, top-hat, and proposed 1D mean iteration methods, respectively.
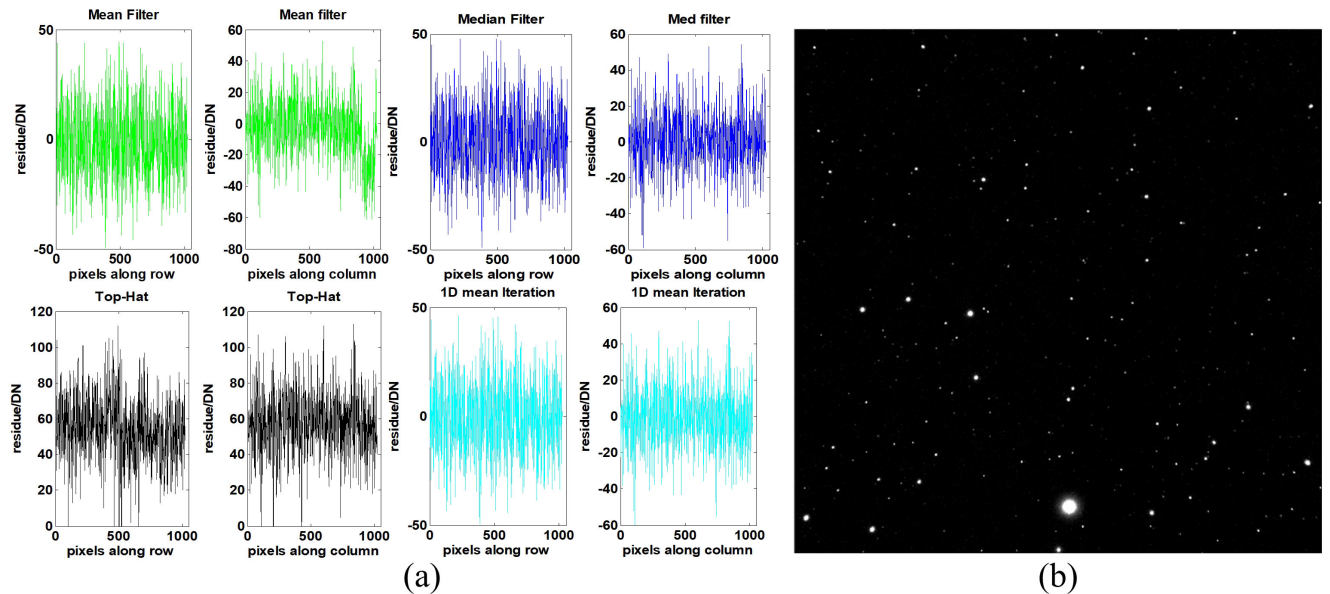

FIGURE 8. (a) Results of background residual errors in the 50th row and line using the mean, median, top-hat, and 1D mean iteration methods. (b) The preprocessed image after hot pixel removal, and background removal using 1D mean iteration.

The background estimation results using the one-dimensional mean iteration method are compared with those of the mean filter (5 iterations), median filter, and top-hat methods. The width and height are set to 100 pixels for the first three methods, and the diameter of the disk structure element set to 50 pixels for the top-hat method. The results are shown in Figures 7, and 8 (a). Suppose the estimated background

is $\hat{B}(i, j)$. The image after removing the background is

$$\tilde{f}(i, j) = f(i, j) - \hat{B}(i, j) \qquad (25)$$

The elusive zone $e_B(i, j)$ computed using the improved adaptive threshold can be used to obtain the residual image

$$R(i, j) = \tilde{f}(i, j)e_B(i, j) \qquad (26)$$

TABLE 1. Residual errors of different background estimation methods.

	Mean Filter	Med Filter	Top-hat	1D Mean
Mean/DN	10.30	9.70	54.55	**4.52**
Std/DN	7.75	6.60	13.78	**2.87**

and the residual errors are computed as shown in Table 1.

It is observed that the proposed one dimensional mean iteration method has the best background estimation results. After removing hot pixels and nonuniform background, the image in Figure 8 (b) shows the effectiveness of the proposed image preprocessing method. After image registration, these images can be directly used in target detection algorithms to detect space debris.

C. SPACE DEBRIS CLASSIFICATION PERFORMANCE WITH DIFFERENT SNRs

The model training uses classification cross-entropy as the loss function, and the Adam optimizer as the training method with the training batch size equal to 500. The loss function is shown below:

$$Loss = -\frac{1}{batch\_size} \sum_{j=1}^{batch\_size} \sum_{i=1}^{n} y_{ji} \log \widehat{y}_{ji} \qquad (27)$$

where *Loss* means the loss value in the batch, y_{ji} is the sample label and $\widehat{y}_{ji}$ is the predicted result. The back-propagation algorithm has been used to optimize the parameters by batch samples.

The dataset is normalized to speed up the model convergence. We normalized each image in the training set and the test set with equation 28.

$$f_c' = \frac{f_c - \min(f_c)}{\max(f_c) - \min(f_c)} \qquad (28)$$

where f_c is the instance with size 33×33. $\max(f_c)$ and $\min(f_c)$ are the largest and smallest values in each instance, respectively. The validation set (10% of the training set) is used for early stopping and the maximum reaching time is set to 10 iterations; namely, if the cross entropy loss value does not decrease 10 consecutive iterations, the model training process will be stopped. The model that corresponds to the lowest cross-entropy loss value on the validation set was chosen as the final model. The relevant training curves are shown in Figure 9. The model converges after approximately 10 epochs. After that, the model overfit, and the accuracy in the training set still increased as the epoch increased, but the validation accuracy did not increase. Therefore, the weights and biases at a training epoch of 10 were used for prediction.

Considering the stability of the proposed method, the experiment uses five metrics to comprehensively access the classification performance of the model:

1) ACCURACY
The proportion of correct classifications:

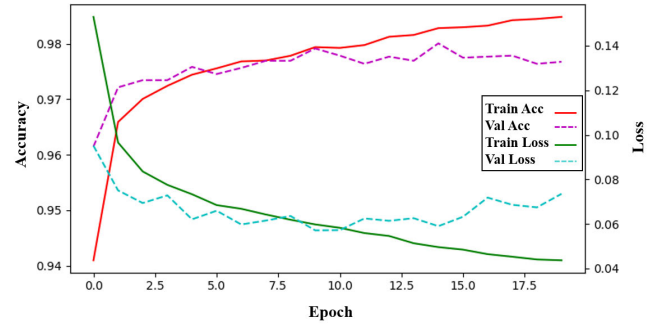$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \qquad (29)$$

FIGURE 9. Accuracy and loss of SDdecNet.

TABLE 2. Comparisons of different sizes of convolutional kernels.

Kernels	Metrices			
	Accuracy	Recall	Precision	F1-score
3×3	0.9743	0.9649	**0.9875**	0.9761
5×5	0.9757	**0.9705**	0.9844	0.9774
7×7	**0.9760**	0.9695	0.9866	**0.9780**

where *TP*, *FP*, *TN*, and *FN* represents numbers of true positive, false positive, true negative, and false negative, respectively.

2) RECALL
The proportion of true space debris that are correctly classified as space debris:

$$R = \frac{TP}{TP + FN} \qquad (30)$$

3) PRECISION
The proportion of images classified as space debris that are true space debris:

$$P = \frac{TP}{TP + TN} \qquad (31)$$

4) F1-SCORE
It measures the performance of the binary classification model, and takes into account the accuracy and recall of the classification model:

$$F1 = \frac{2PR}{P + R} \qquad (32)$$

5) AUC
The area under the receiving operating characteristic (ROC) curve.

The values of the first three metrics depend on the threshold we choose. The default choice is to classify a candidate as space debris if its predicted probability is above 0.5. Different kernel sizes have been tested by the validation set, and the evaluation results on entire test set are shown in Table 2. It is observed that the accuracy, recall, and F1-score obtained when a 5×5 kernels is used are better than those obtained when a 3×3 kernel is used. The difference between 5×5 and 7×7 kernels is almost negligible. Considering the computing

TABLE 3. Comparison of the metrices of different SNR.

Metrices	SNR					
	1.2	1.0	0.8	0.5	0.4	0.3
Accuracy	0.9910	0.9910	0.9900	0.9655	0.9550	0.8690
Recall	1.0000	1.0000	0.9980	0.9490	0.9280	0.7560
Precision	0.9823	0.9823	0.9823	0.9814	0.9810	0.9767
F1-score	0.9911	0.9911	0.9901	0.9649	0.9538	0.8523
AUC	0.9999	0.9998	0.9996	0.9944	0.9924	0.9753
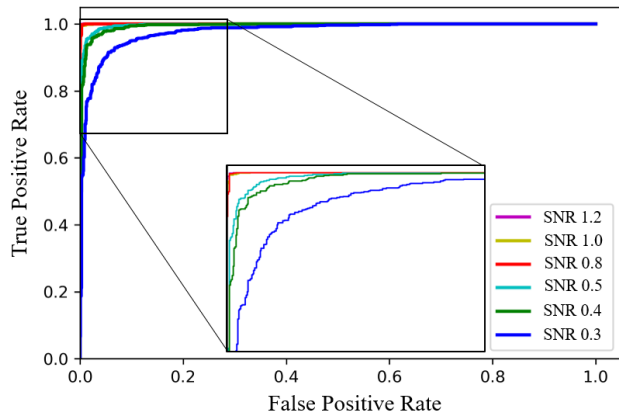
FIGURE 10. The ROC curves of classification model using different SNRs.

FIGURE 11. The confusion matrices of the classification results with different SNRs.

time and the test performance, a kernel size of 5×5 is chosen. The final classification accuracy of the obtained global test set is 0.9745, and the classification time for a single sample is 180 us. Table 3 shows the accuracy, recall, precision, F1-score and AUC, of the test sets with SNR $\leq$ 1.2. When the SNR $\geq$ 1.3, all the test samples are classified correctly.

If the SNR is higher than 1.0, all five metrics are near 1. When the SNR is lower, the metrics value decrease, and can be as low as 0.3, with the recall value decreasing to 0.7560 and the accuracy value decreasing to 0.869. The results show that our model has excellent classification performance for space debris.

Figure 10 shows an ROC curve with the convolutional neural network with 6 different SNRs with the testing set. The closer a curve is to the top left corner of the ROC curve, the better the performance of the model is. Sometimes, ROC curves may crossover with different SNRs, and it is difficult to compare them with each other. The area under the curve (AUC) can be used in such case. The area under the curves is shown in Tabel 3 line 5. For each level, every point on its curve corresponds to a different choice of classification threshold. The true positive rate (TPR) means the proportion of the true space debris that is correctly classified as space debris in all space debris samples, also denoted by "Recall". The false positive rate (FPR) means that the proportion of the true space debris that is correctly classified as noise in all noisy samples. The curves of SNR1.2, SNR1.0, and SNR0.8 were very steep and close to each other. The
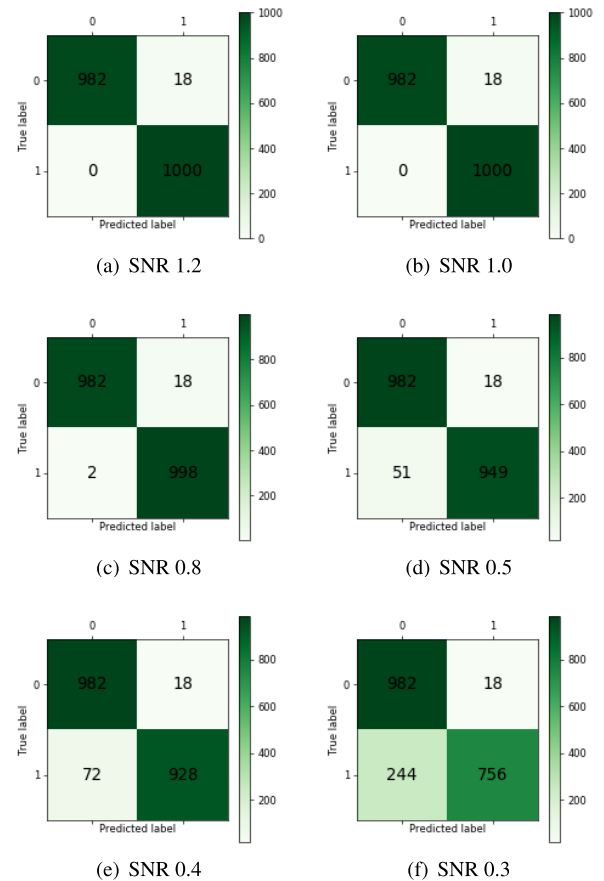
remaining curves decreased as the SNR decreased. The blue curve ($SNR = 0.3$) has the lowest score.

Figure 11 shows the confusion matrix for the convolutional architecture model with the six different SNR test sets. In the confusion matrix, the horizontal axis represents the predicted labels and the vertical axis means true labels. The value 0 at the axis represent noise (negative sample) and the value 1 represents space debris (positive sample). Therefore, (0, 0), (0, 1), (1, 0) and (1, 1) represent TN, FN, FP and TN, respectively. In the matrices, the number of background noise samples which are misclassified as space debris is 18. The result proves the stability of classification with different SNRs. The number of space debris wrongly classified as noise increased from 0, 2, 51, to 244. When the SNR $\leq$ 0.5, many space debris were misclassified as background noise. The samples in Figure 4 (r), (s), and (t) are very noisy; therefore, classifying these objects correctly is very difficult.

The results in Table3, and Figures 10 and 11 show that, for SNR $\geq$ 0.8, the proposed learning model has excellent classification performance. If the SNR $\leq$ 0.3, it is difficult to distinguish space debris and background noise, and the model nearly losses its classification ability. For space debris, especially those with very low SNR,
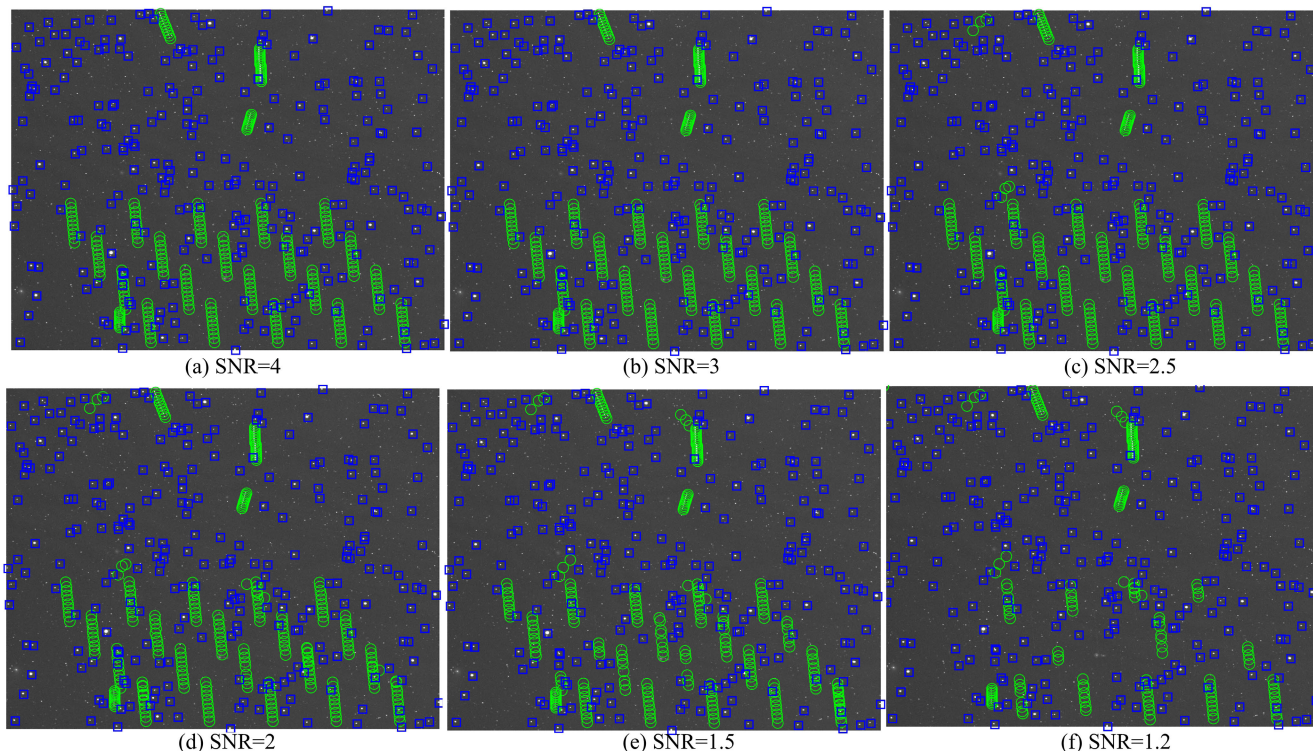
(a) SNR=4 (b) SNR=3 (c) SNR=2.5

(d) SNR=2 (e) SNR=1.5 (f) SNR=1.2

FIGURE 12. Detection results of space debris with different SNRs in acquired real images sequences (containing 20 simulated space debris, and 5 real space debris).

TABLE 4. Detection results with different SNRs in the image sequences.

SNR	Total Num.	Detected Num.	Detection Probability	False Alarm Num.	False Alarm Rate
4	200	200	100%	0	0%
3	200	200	100%	0	0%
2.5	200	200	100%	6	2.9%
2	200	200	100%	9	4.3%
1.5	200	171	85.5%	11	5.2%
1.2	200	83	41.5%	11	5.2%

TABLE 5. Detection results of different methods in the image sequences.

Methods	Detection Probability (%)			
	SNR=6	SNR=3	SNR=2	SNR=1.5
MHT [51]	97.5	95.0	86.7	75.0
EAOM [25]	97.25	94.75	-	-
FLCR	100	100	100	85.5

the model performance in extracting candidate regions is another important factor for the final space debris detection performance.

D. DETECTION PERFORMANCE OF SPACE DEBRIS IN OPTICAL IMAGE SEQUENCES

In this experiment, we tested the performance of space debris detection in optical images. There are 10 sequential images used in the experiments, which were captured by the CCD telescope; the size of each image is 1024×1024 pixels with 16 bits of gray scales. The optical system collects 80% of the energy of the space debris and stars in a window with a size of 2×2. Twenty space debris with SNRs 4, 3, 2.5, 2, 1.5, and 1.2 were generated and added to the 10 sequential images. Each group of space debris is evenly spaced in the images. In addition, there are also 5 pieces of real space debris in the images.

The detection results are shown in Fig. 12, and Table 4. When the SNR is greater than or equal to 2, the detection probability is 100%. When the SNR is 1.5, the detection probability can still reach 85.5%. There are no false alarm in the experiment when the SNR is higher than 2.5. This is because a higher segmentation threshold is used. When the SNR is less than or equal to 2.5, the number of false alarms is slightly large. These false alarms can be easily removed using the continuity of trajectory in the succeeding image sequences [20]. Finally, the proposed method is compared with the MHT method for streak-like dim and small object detection [51] with $SNR = 6, 3, 2,$ and 1.5. We also compared this method with the EAOM algorithm with $SNR = 6, 3,$ [25]. The result is shown in Table 5, and it is observed that the proposed method has a better detection performance than that of the MHT and EAOM methods. Jia *et al.* [52] proposed a method that uses a neural network for optical transient object classification. This method can reach an accuracy of 97.20% in streak-like objects with minimal $SNR = 3.10$. Our proposed method can reach an accuracy of 100.00% with

$SNR \geq 3$, which means that our FLCR method has a better performance.

V. CONCLUSION

Space debris detection is important in space asset protection. In this article, we proposed a space debris detection method using feature learning of candidate regions in astronomical optical image sequences. First, image preprocessing is performed including hot pixels and flicker noise removal, and nonuniform background estimation and removal. Second, the feature learning of candidate regions method is proposed to detect space debris, in which the stars are removed to reduce the false alarm, the candidate regions are extracted, and deep learning is utilized to learn the spatial features to discriminate the space debris and the background. The classification results show that space debris can be detected and located in the image sequences. The experimental results also show that the proposed method can detect low SNR space debris effectively with high accuracy.

ACKNOWLEDGMENT

REFERENCES

[1] M. Garcia. (Apr. 2015). *Space Debris and Human Spacecraft*. [Online]. Available: http://www.nasa.gov/mission_pages/station/news/orbital_debris.html

[2] S. Lemmens, "ESA annual space environment report," ESA Space Debris Office, Darmstadt, Germany, Tech. Rep. GEN-DB-LOG-00271-OPS-SD, Jul. 2019.

[3] (2019). *Space Debris-The Threat Hanging Over Our Heads*. [Online]. Available: https://astroscale.com/space-debris/

[4] N. C. Mohanty, "Computer tracking of moving point targets in space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 5, pp. 606–611, Sep. 1981.

[5] I. S. Reed, R. M. Gagliardi, and H. M. Shao, "Application of three-dimensional filtering to moving target detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 6, pp. 898–905, Nov. 1983.

[6] Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 1, pp. 144–156, Jan. 1985.

[7] S. Buzzi, M. Lops, and L. Venturino, "Track-before-detect procedures for early detection of moving target from airborne radars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 937–954, Jul. 2005.

[8] V. Kouprianov, "Distinguishing features of CCD astrometry of faint GEO objects," *Adv. Space Res.*, vol. 41, no. 7, pp. 1029–1038, Jan. 2008.

[9] M. Tagawa, T. Hanada, H. Oda, H. Kurosaki, and T. Yanagisawa, "Detection algorithm of small and fast orbital objects using faint streaks; application to geosynchronous orbit objects," in *Proc. 40th COSPAR Sci. Assembly*, vol. 40, 2014, Paper PEDAS-1.

[10] J. Virtanen, J. Poikonen, T. Säntti, T. Komulainen, J. Torppa, M. Granvik, K. Muinonen, H. Pentikäinen, J. Martikainen, J. Näränen, J. Lehti, and T. Flohrer, "Streak detection and analysis pipeline for space-debris optical images," *Adv. Space Res.*, vol. 57, no. 8, pp. 1607–1623, Apr. 2016.

[11] S. D. Blostein and T. S. Huang, "Detecting small, moving objects in image sequences using sequential hypothesis testing," *IEEE Trans. Signal Process.*, vol. 39, no. 7, pp. 1611–1629, Jul. 1991.

[12] G. C. Demos, "Applications of MHT to dim moving targets," in *Signal Data Processing of Small Targets*, vol. 1305. Bellingham, WA, USA: SPIE, 1990, p. 297.

[13] S. D. Blostein and H. S. Richardson, "A sequential detection approach to target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 1, pp. 197–212, Jan. 1994.

[14] T. Yanagisawa, H. Kurosaki, H. Banno, Y. Kitazawa, M. Uetsuhara, and T. Hanada, "Comparison between four detection algorithms for GEO objects," in *Proc. Adv. Maui Opt. Space Surveill. Technol. Conf.*, vol. 1114, 2012, p. 9197.

[15] J. Núñez, A. Núñez, F. J. Montojo, and M. Condominas, "Improving space debris detection in GEO ring using image deconvolution," *Adv. Space Res.*, vol. 56, no. 2, pp. 218–228, Jul. 2015.

[16] R. Sun and C. Zhao, "A new source extraction algorithm for optical space debris observation," *Res. Astron. Astrophys.*, vol. 13, no. 5, p. 604, 2013.

[17] R.-Y. Sun, J.-W. Zhan, C.-Y. Zhao, and X.-X. Zhang, "Algorithms and applications for detecting faint space debris in GEO," *Acta Astronautica*, vol. 110, pp. 9–17, May 2015.

[18] J. Xi, "A frame correlation and Kalman filtering algorithm for space debris detection," *J. Inf. Comput. Sci.*, vol. 11, no. 15, pp. 5317–5325, Oct. 2014.

[19] J. Xi, "A maximum projection and particle filtering algorithm for space debris detection," *J. Inf. Comput. Sci.*, vol. 12, no. 1, pp. 161–169, Jan. 2015.

[20] J. Xi, D. Wen, O. K. Ersoy, H. Yi, D. Yao, Z. Song, and S. Xi, "Space debris detection in optical image sequences," *Appl. Opt.*, vol. 55, no. 28, p. 7929, Oct. 2016.

[21] S. Ding, H. Wang, D. Chen, T. Fu, and M. Gao, "An improved method for dim space debris detection based on Hough transform," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 1534–1538.

[22] L. S. C. Metrailler, A. Vananti, T. Schildknecht, J.-N. Pittet, J. Utzmann, and T. Flohrer, "The difference method: A simple and effective on-board algorithm for space debris detection," Adelaide, SA, Australia: International Astronautical Federation (IAF), Sep. 2017, pp. 1–9.

[23] F. Diprima, F. Santoni, F. Piergentili, V. Fortunato, C. Abbattista, and L. Amoruso, "Efficient and automatic image reduction framework for space debris detection based on GPU technology," *Acta Astronautica*, vol. 145, pp. 332–341, Apr. 2018.

[24] S. Yang, Z. Ding, and L. Dai, "A fast integration algorithm for detection of weak space debris using incoherent scatter radar," in *Proc. 12th Int. Symp. Antennas, Propag. EM Theory (ISAPE)*, Hangzhou, China, Dec. 2018, pp. 1–4.

[25] S. Kong, J. Zhou, and W. Ma, "Effect analysis of optical masking algorithm for GEO space debris detection," *Int. J. Opt.*, vol. 2019, pp. 1–8, Mar. 2019.

[26] Q. Sun, Z. D. Niu, and C. Yao, "Implementation of real-time detection algorithm for space debris based on multi-core DSP," *J. Phys. Conf. Ser.*, vol. 1335, Oct. 2019, Art. no. 012003.

[27] K. Liu, X. Hu, Z. Wei, Y. Li, and Y. Jiang, "Modified Gaussian process regression models for cyclic capacity prediction of lithium-ion batteries," *IEEE Trans. Transport. Electrific.*, vol. 5, no. 4, pp. 1225–1236, Dec. 2019.

[28] X. Tang, K. Liu, X. Wang, F. Gao, J. Macro, and W. D. Widanage, "Model migration neural network for predicting battery aging trajectories," *IEEE Trans. Transport. Electrific.*, vol. 6, no. 2, pp. 363–374, Jun. 2020.

[29] K. Liu, Y. Li, X. Hu, M. Lucu, and W. D. Widanage, "Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3767–3777, Jun. 2020.

[30] K. Liu, Y. Shang, Q. Ouyang, and W. D. Widanage, "A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery," *IEEE Trans. Ind. Electron.*, early access, Mar. 18, 2020, doi: 10.1109/TIE.2020.2973876.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[33] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2012, *arXiv:1206.5538*. [Online]. Available: http://arxiv.org/abs/1206.5538

[34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[36] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[38] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[40] J. Hu, Y. Hu, and X. Lu, "A new method of small target detection based on neural network," *Proc. SPIE*, vol. 10608, Feb. 2018, Art. no. 106080H.

[41] L. Varela, L. E. Boucheron, N. Malone, and N. Spurlock, "Streak detection in wide field of view images using convolutional neural networks (CNNs)," in *Proc. Adv. Maui Opt. Space Surveill. Technol. Conf.*, 2019, p. 89.

[42] P. Jia, Q. Liu, and Y. Sun, "Detection and classification of astronomical targets with deep neural networks in wide-field small aperture telescopes," *Astronom. J.*, vol. 159, no. 5, p. 212, 2020. [Online]. Available: https://iopscience.iop.org/article/10.3847/1538-3881/ab800a

[43] S. Xie, G. Ross, D. Piotr, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.

[44] T. Lin, D. Piotr, G. Ross, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[45] C. C. Liebe, "Accuracy performance of star trackers—A tutorial," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 587–599, Aug. 2002.

[46] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, p. 9, 2011.

[47] J. Xi, Z. Song, and W. Gao, "Sub-pixel location with phase transfer function for star tracker," in *Proc. 8th Int. Symp. Precis. Eng. Meas. Instrum.*, vol. 8759, 2013, Art. no. 87590U.

[48] J.-Q. Sun, J. Zhou, Z. Zhang, and Y.-P. Zhang, "Centroid location for space targets based on energy accumulation," *Opt. Precis. Eng.*, vol. 19, no. 12, pp. 3043–3048, 2011.

[49] B. F. Alexander and K. C. Ng, "Elimination of systematic error in subpixel accuracy centroid estimation [also Letter 34(11)3347-3348(Nov1995)]," *Opt. Eng.*, vol. 30, no. 9, pp. 1320–1332, 1991.

[50] J. Xi, "Rapid star extraction and high accuracy location in star image," *J. Inf. Comput. Sci.*, vol. 12, no. 8, pp. 2929–2937, May 2015.

[51] M. Li, C. Yan, C. Hu, C. Liu, and L. Xu, "Space target detection in complicated situations for wide-field surveillance," *IEEE Access*, vol. 7, pp. 123658–123670, 2019.

[52] P. Jia, Y. Zhao, G. Xue, and D. Cai, "Optical transient object classification in wide field small aperture telescopes with neural networks," *Astronom. J.*, vol. 157, no. 6, p. 250, 2019. [Online]. Available: http://arxiv.org/abs/1904.12987

YAOBING XIANG received the B.S. degree in survey and mapping from Chang'an University, China, in 2019. He is currently pursuing the M.S. degree in photogrammetry and remote sensing with Nanjing Normal University. His research interests include the remote sensing image processing and machine learning.

OKAN K. ERSOY is currently a Professor of electrical and computer engineering with Purdue University. He also holds five patents. His research interests include neural networks, machine learning and pattern recognition, decision analytics, digital signal/image processing and recognition, transform and time–frequency methods, diffractive optics, and their applications. He has published approximately 250 articles, two books, and several book chapters in his areas of research. He is a Fellow of the Optical Society of America.

MING CONG received the Ph.D. degree from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, China, in 2015. He is currently a Lecturer with the School of Geological Engineering and Surveying, Chang'an University, Xi'an, China. He has published many articles on *Remote Sensing* journals. His research interests include remote sensing image processing and photogrammetry.

XIN WEI received the bachelor's degree in microelectronics from Jilin University, China, in 2014. He is currently pursuing the Ph.D. degree in signal and information processing (SIP) with the University of Chinese Academy of Sciences (UCAS). His research interests include the embedded systems, star identification, and machine learning.

JIANGBO XI received the B.S. degree in electronic and information science and technology from Jilin University, China, in 2008, and the Ph.D. degree in signal and information processing from the University of Chinese Academy of Sciences (UCAS), China, in 2017. He was a joint Ph.D. Student in electrical and computer engineering with Purdue University, USA, from April 2015 to April 2017. He is currently an Assistant Professor with the School of Geology Engineering and Geomatics, Chang'an University, China. His current research interests include deep learning, machine learning, and objects detection in optical sequential images.

JUNKAI GU received the B.S. and M.S. degrees in photogrammetry and remote sensing from Wuhan University, China, in 2003 and 2006, respectively, and the Ph.D. degree in environment remote sensing from Chang'an University, China, in 2015. He is currently an Engineer with the College of Geological Engineering and Geomatics, Chang'an University. His research interests include the photogrammetry, computer vision, and deep learning.

• • •