IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 10, 2019, accepted July 5, 2019, date of publication July 16, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927684

RPNet: A Representation Learning-Based Star Identification Algorithm

LIKAI XU, JIE JIANG, AND LEI LIU
School of Instrumentation Science and Engineering, Beihang University, Beijing 100191, China
Key Laboratory of Precision Opto-Mechatronics Technology, Ministry of Education, Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beijing 100191, China

Corresponding author: Jie Jiang (jiangjie@buaa.edu.cn)

This work was supported by the National Neural Science Fund of China 61725501.

ABSTRACT A novel star identification network (RPNet) based on representation learning is proposed in this paper. Unlike other pattern-based stars identification algorithms, the RPNet does not require the creation of an elaborate pattern, nor does it need to search among patterns. Instead, a star pattern generator (SPG) in the RPNet helps in finding the best pattern that can distinguish different stars clearly. A star pattern classifier (SPC) in the RPNet is utilized to recognize the pattern generated before. The simulations show that the RPNet is extremely robust toward star position noise, star magnitude noise, and false stars. The performance on simulation images outperforms almost all other pattern-based stars identification algorithms. On average, it achieves an identification rate of 99.23% in simulated star images. The identification rate on real star images is higher than 98%. Moreover, the algorithm achieves this performance with lesser memory and faster speed compared to polygon algorithms.

INDEX TERMS Neural networks, pattern recognition, representation learning, star identification.

I. INTRODUCTION

The navigation system is an indispensable part of a spacecraft and includes attitude determination, speed measurement, and positioning. The star sensor [1] is an important device in the navigation system that can determine its three-axis attitude without any prior information. Its high accuracy makes it widely used in spacecraft. The star sensor has two operating modes: The initial attitude acquiring mode and tracking mode. Though most of the time it works on tracking mode, a star identification process is needed for a star sensor to obtain current attitude of the spacecraft once it loses in space.

Star identification algorithms aim at finding the correlation between observed and cataloged stars. It mainly falls into two categories: the subgraph isomorphism algorithm and pattern-based identification algorithm.

The subgraph isomorphism-based algorithms borrow ideas from graph theory. Stars are treated as vertexes, angular distances between star pairs are the weight of the edge, a subgraph is constructed from a stellar image and the complete graph is formed from the star catalog; Identification problems can be reduced to find the most similar subgraph in the entire graph. The triangle algorithm [2] is the most typical and widely used one; it uses three stars and their angular distances between each other to generate a triangle feature. The algorithm is quite robust even though there are only three stars in a stellar image, but it is time-consuming because the number of triangle features formed in a star catalog have to be searched and matched. To accelerate the search process, Quine [3] modified the triangle algorithm by filtering the redundant triangles of each star, Kruijff [4] assigned each star a probability based on the magnitude of detected stars to select the most reliable triangle, and Mortari *et al.* [5] proposed the pyramid algorithm to solve the false star condition. Wang *et al.* [6] used a hash map to improve the searching time of the pyramid algorithm. Another type of subgraph isomorphism algorithm utilizes a voting strategy based on the count of angular distances [7]; the algorithm proves to be tolerant to position noise but behaves terribly with the increase of false stars. Li *et al.* [8] improved the geometric voting algorithm by adding an iterative process to eliminate false stars, but cannot converge sometimes when the true stars in star images are not sufficient.

Pattern-based star identification algorithms construct a unique pattern for each guiding reference star (GRS) chosen from the star catalog; this pattern of a GRS usually contains

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

information from its guiding neighbor stars (GNSs). The matching process is to find the most similar pattern among all of the pre-calculated patterns. The most representative pattern-based approach is the grid algorithm [9] proposed by Padgett; it has a remarkable tolerance for position noise, but once the nearest neighbor star is missing or wrongly detected, the pattern calculated and the identification would be incorrect. Zhang *et al.* [10] put forward an algorithm on the basis of radial and cyclic features; the pattern is with rotation invariance and it overcomes the shortcomings of the grid algorithm. In addition, the algorithm takes a step-by-step matching strategy to improve the identification accuracy, but it still suffers a lot from plenty of false stars and missing stars. Jiang *et al.* [11] utilized a redundant-coded radial and cyclic pattern to conduct the identification. Based on the fundamental of the former, Sun *et al.* [12] created a Hidden Markov model (HMM) based pattern to accelerate the matching process. In [13], the K-L transform is taken to form a rough matching, then a star walk procedure is performed to identify stars precisely. However, when the number of false stars increases, the identification rate decreases severely.

With the boom of artificial intelligence, neural networks [14]–[16] and some advanced algorithms such as genetic algorithms [17] were applied on star identification. However, the complicated network structure and insufficient robustness of these algorithms resulted in lower identification rates and slower identification speed. Therefore, this paper proposes a representation learning based star identification network RPNet whose network structure is simple but highly efficient. The RPNet utilizes a star pattern generator (SPG) to generate star patterns for the GRS, then a star pattern classifier (SPC) is applied for the identification of the pattern generated previously. Finally, a weight searching strategy is exerted to guarantee identification accuracy. By learning from a huge amount of data, the pattern generated by RPNet is pretty robust to position noise, star magnitude noise, and false stars; it outperforms other pattern-based algorithms and AI algorithms on average. Meanwhile, using a neural network to do the inference is much more time efficient than other searching-based algorithms. The proposed algorithm behaves well in real star images too.

The paper is organized as follows: In section 2, the concrete implementation of the algorithm is described. In section 3, comparisons between RPNet and other algorithms on simulated and real star images are made. Finally, the paper makes a conclusion in section 4.

II. METHODOLOGY

The paper proposes an end-to-end and representation learning based neural network model RPNet for fast, efficient, and robust star identification tasks. RPNet learns a unique star pattern R-Pattern from a huge amount of simulated star images for each GRS in a catalog, then a classification is made on these star patterns learned before. RPNet consists mainly of two parts: (1) a SPG based on the star pattern generation network to generate unique star patterns for star image samples, and (2) a SPC to classify the star patterns generated previously. Compared with other pattern-based star identification algorithms, RPNet does not need to manually design star patterns; instead, it is driven by big data, and it generates star patterns by self-learning. These star patterns are more discriminating than man-made star patterns. Therefore, it has a lower requirement for backend SPC, only a relatively simple feed-forward neural network can produce satisfactory results. It greatly reduces the overhead of the training and prediction of the network, and it avoids the over-fitting caused by a complicated network structure. Meanwhile, it provides convenience for later implementation on embedded systems. A weight searching strategy is also proposed in the paper for filtering and verification of GRSs identified by RPNet, which improves tremendously the identification ability for a single star image. The flow chart of the entire algorithm is shown in Fig. 1.

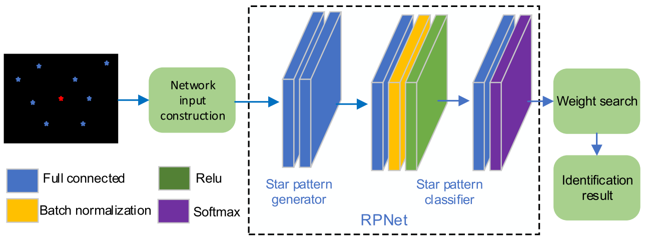


FIGURE 1. **Flow chart of the proposed algorithm.**

The methodology to construct the input of RPNet by GRSs and GNSs within a pattern radius is introduced in this chapter. The structure and training process of the RPNet, as well as the generation of the training data set, is subsequently described in detail. Finally, a weight searching method is put forward to verify the identification results.

A. RPNET INPUT CONSTRUCTION

The size of the star catalog influences directly the efficiency and accuracy of the star identification, thus a screening process for the original star catalog is imperative and the remaining stars subsequently become GRSs. GRSs should meet the following requirements: (1) GRSs should be observable under a certain degree of star magnitude noise. (2) The number of GRSs in a star image should not be too small on average. (3) The angular distance between two GRSs cannot be too close in case of wrong identification. Based on the criteria above, the Smithsonian Astrophysical Observatory Star Catalog (SAO) [18] was chosen. A limiting magnitude that can be detected by a star sensor is set to 6.0 Mv. Some stars in dense areas are also deleted selectively and one of the ‘‘double stars’’ whose angular distance between each other is too close is retained randomly. Finally, only star magnitude, right ascension, and declination of GRSs are kept.

Through the filtering process above, a total of 5,045 GRSs whose star indices are renumbered from 0 to 5,044. The new star catalog formed by these GRSs is called the nSAO star catalog. The identification for GRSs by RPNet is a typical classification problem. The data and its characteristics determine the upper bound of the classification accuracy; the modules and algorithms merely approach the bound. RPNet generates patterns of GRSs via a SPG. Though it does not need to manually design the pattern, the input of the SPG still should contain sufficient information to guarantee to produce a well discriminative pattern. So, it is quite critical to construct the initial input for RPNet according to the GRS and its GNSs within pattern radius.

It is feasible to determine GNSs of a GRS by limiting the angular distance between the GRS and other stars in the celestial coordinate system. Considering that various types of noises are generated from the image level, the GRS and GNSs are transformed from the celestial coordinate system into the image coordinate system. Assuming that the position of a GRS denoted by S in the celestial coordinate system is (α_s, β_s) (with α_s as right ascension and β_s as declination), let the attitude angle of the star sensor be $(\alpha_s, \beta_s, \phi_s)$, where φ_s is the roll angle (whose value has no effect on the relative relationship between S and its GNSs) that can be set to a random value in the range $[-2\pi, 2\pi]$. For another star N whose coordinates in the celestial coordinate system are (α_n, β_n), the projection [19] from the celestial coordinate system into the image coordinate system can be represented via (1):

$$s\begin{pmatrix} X_n \\ Y_n \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha_n \cos\beta_n \\ \sin\alpha_n \cos\beta_n \\ \sin\beta_n \end{pmatrix}, \quad (1)$$

where s is the scale factor, (X_n, Y_n) is the projected position of N in the image coordinate system, f is the optical focus, (u_0, v_0) is the main point, and $a_{11} \sim a_{33}$ are elements of the transformation matrix. The angular distance between S and N can be calculated via (2):

$$d_{sn} = \arccos\left(\frac{v_s \cdot v_n}{\|v_s\| \cdot \|v_n\|}\right)$$
$$v_s = \frac{1}{\sqrt{X_s^2 + Y_s^2 + f^2}}\begin{pmatrix} X_s \\ Y_s \\ -f \end{pmatrix},$$
$$v_n = \frac{1}{\sqrt{X_n^2 + Y_n^2 + f^2}}\begin{pmatrix} X_n \\ Y_n \\ -f \end{pmatrix} \quad (2)$$

where N is considered the GNS of S when d_{sn} is within the pattern radius ρ. After obtaining all GNSs of S, the methodology to construct the initial input for RPNet according to S and geometric distribution of its GNSs is of great importance. The initial input should meet the following demands of the self-learning property of RPNet and the distribution of GNSs: (1) The length of the initial input should be fixed so that the number of GNSs does not affect the size of the initial input. (2) The construction procedure of the initial input should be simple and efficient; it should contain sufficient information to perform the identification. (3) The initial inputs of different GRSs should be discriminative. Based on the criteria above, the initial input construction of S is shown in the following steps:

1. Choose the right ascension and declination of S and a random roll angle as the boresight pointing of the star sensor. Project S and neighboring stars from the celestial coordinate system into the image coordinate system. Calculate angular distances between S and neighboring stars and determine the GNSs of S that are within the range of pattern radius ρ.
2. Filter out GNSs far away from main star S. Only keep m nearest GNSs, if the number of GNSs is smaller than m. For all GNSs retained, denote these selected GNSs by s-GNSs. Calculate the angular distances between S and s-GNSs and the pair-wise angular distance of s-GNSs, respectively. Arrange two groups of angular distances as vector d_{sn} and vector d_{nn} separately.
3. Discrete d_{sn} and d_{nn}; the discretization factors are e and $2e$, respectively. For the maximum angular distances in d_{sn} and d_{nn}, these are ρ and 2ρ, the length of the discretized vectors D_{sn} and D_{nn} are both $\lfloor \rho/e \rfloor$. For d_1 in d_{sn} and d_2 in d_{nn}, the discretization process can be formulated by (3):

$$id_1 = d_1/e, \quad D_{sn}[id_1] = D_{sn}[id_1] + 1$$
$$id_2 = d_2/2e, \quad D_{nn}[id_2] = D_{nn}[id_2] + 1. \quad (3)$$

4. Concatenate D_{sn} and D_{nn} to form a 1-D array S_{ori}, as the initial input fed into RPNet.

The concrete construction procedure of S_{ori} is shown in Fig.2.

B. STRUCTURE OF RPNET

RPNet is composed mainly of two modules: the front end is a SPG based on a pattern generation network; followed by a SPC formed by a feed-forward neural network. For the initial input S_{ori} of a GRS, a unique pattern RP_S is generated via the encoding of SPG to S_{ori}, then SPC classifies this unique pattern to gain the probability of each star that the GRS may belong to in the nSAO catalog. The structure of the RPNet is shown in Fig.3.

The description of the two modules are as follows.

1) STAR PATTERN GENERATOR

The SPG consists of two fully connected layers and a single activation function layer. It is the encoder part of a pattern generation network. The encoder part of a well-trained pattern generation network can encode a unique pattern for the initial input and map the input from the $2\lfloor \rho/e \rfloor$ dimension into N-dimensional output pattern RP_S.The high discriminative pattern it generates guarantees the simplicity of the subsequent SPC.

2) STAR PATTERN CLASSIFIER

It is based on the typical feed-forward neural network. A recently proposed batch normalization layer [20] is added to gain a more robust performance. The SPC plays a role in

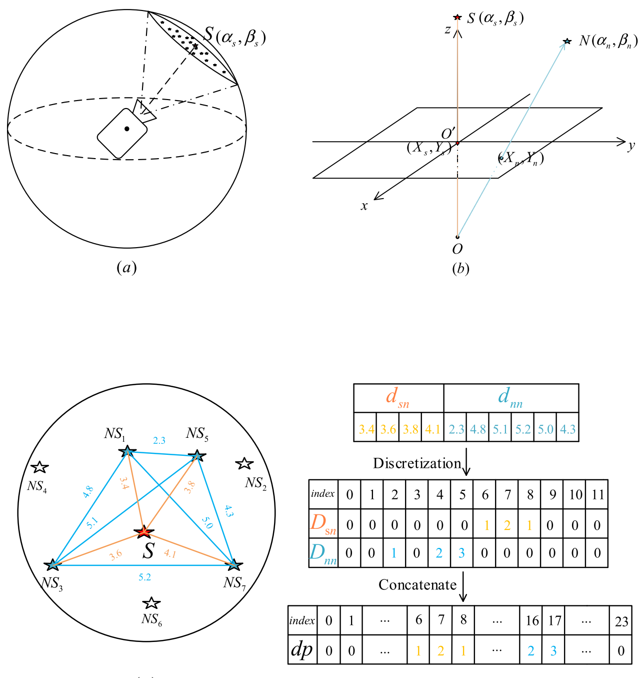


FIGURE 2. Initial input construction of RPNet for S. (a) S and its neighbor stars observed by the star sensor at the boresight pointing of $(\alpha_S, \beta_S, \phi_S)$. (b) Project S and its neighbor stars onto the image plane. (c) Calculate d_{sn} and d_{nn} based on S and its m nearest GNSs. (d) Discretization. For simplicity and better visualization, parameters are set as $\rho = 6, e = 0.5°, m = 4°$.

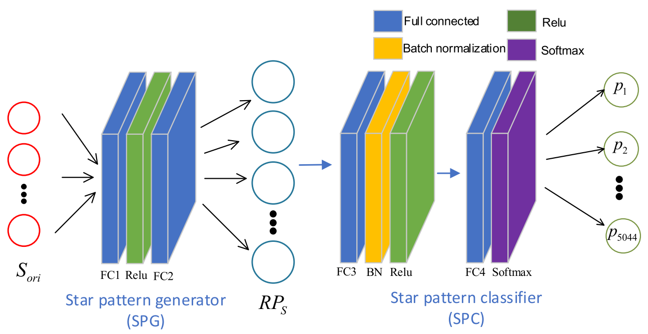


FIGURE 3. Structure of RPNet.

assigning probabilities for stars in the nSAO catalog that the GRS may belong to for later identification.

The specific hierarchical structure of RPNet is shown in Table 1.

After the normalization process, the output of RPNet obeys a probability distribution of

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad j = 0, 1, \ldots K-1, \tag{4}$$

where K denotes the size of the nSAO catalog and the output p_j meets the demand $\sum_{j=0}^{K-1} p_j = 1$, which denotes the predicted probability of the j-th star in the nSAO catalog.

TABLE 1. Architecture of RPNet.

Layer	Parameters
FC1	Weight 400×512
FC2	Weight 512×1024
FC3	Weight 1024×512
BN	Batch normalization, enables the input to allow a distribution of N(0, 1).
Relu	Activation function, enables the network to have nonlinearity.
FC4	Weight 512×5045 , maps into the space of the nSAO catalog.
Softmax	Normalize to a probability distribution.

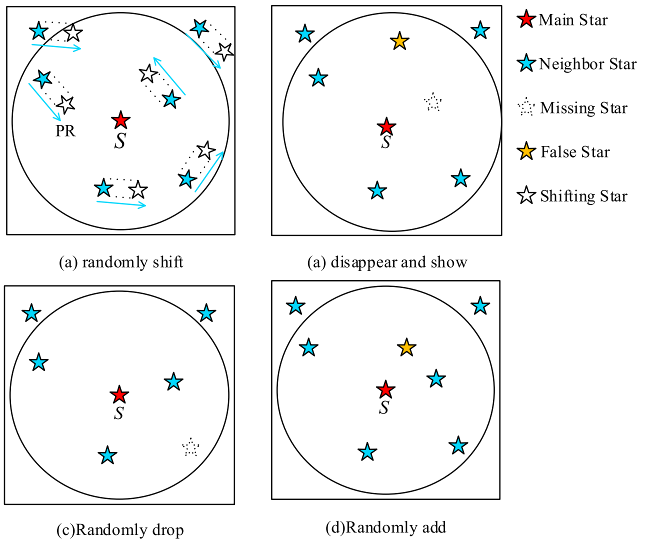


FIGURE 4. Methods for adding different types of noise.

C. GENERATION OF TRAINING DATA

Both pattern generation and pattern classification networks need to complete learning through training on adequate samples. The pattern generation network requires noise samples and corresponding clean samples for training. The training of a pattern classification network requires patterns generated from a well-trained SPG previously and their related labels.

For a GRS named S, a simulated star image I without noise can be generated as described above in this chapter. The clean sample of S can be acquired directly by constructing initial input of S from I. Noise images are generated by adding different types and levels of noise into the image I. Noise samples can be generated through the initial input construction of noise images. For the pattern generation network, types and quantities of noise samples determine the discrimination of the patterns it generates. The more noise samples it consumes, the higher the accuracy of the identification. However, because they are limited by hardware condition and efficiency, noise types and number of noise samples are not infinite. The types of noise and the way they are added are shown in Fig. 4.

The details of the generating procedure for the four types of noise samples are described below.

1) POSITION NOISE SAMPLES

Position noise results in shifts in the stars' coordinates in image I. For all of stars except S in image I, Gaussian noise of mean of 0 and standard deviation of σ_x and σ_y are added into their X and Y coordinates, respectively, which is $pn_X \sim N(0, \sigma_x^2), pn_Y \sim N(0, \sigma_y^2)$. Set $\sigma_x = \sigma_y$ for symmetry. The range of the standard deviation is 0.2 pixel to 1.6 pixels, at a step of 0.2 pixel. Ten noise samples are assigned for each level of position noise and 80 position noise samples of S are generated in total.

2) MAGNITUDE NOISE SAMPLES

Magnitude noise leads to variations in a star's brightness. For stars with magnitudes around the limited detectable magnitude of a sensor, magnitude noise may make it darker, causing it to disappear. For stars that cannot be detected before, magnitude noise may brighten it to make it appear. Therefore, a 7.0 Mv limited detectable magnitude is set as the threshold, then a star image I is simulated according to the threshold. Magnitude noise is added into each star except S in the image I, then the stars darker than 6.0Mv are filtered out. Finally, the noise sample can be constructed from the noisy image generated before. Star magnitude noise is subject to a Gaussian distribution of $N(0, \sigma^2)$, where $\sigma \in [0.323, 0.484, 0.646, 0.807, 0.969]$. For each level of magnitude noise, 16 noise samples are generated, which adds up to 80 noise samples.

3) MISSING STAR SAMPLES

Some stars that should have appeared disappear at random because of circuit noise or optical system imaging noise. For stars except for S in image I, 1 to 5 stars were dropped randomly to produce the missing star samples. Eighty samples of this kind of noise were taken.

4) FALSE STAR SAMPLES

False stars are quite common in real star images. The planet, cosmic dust, background white noise, and even the flame from the tail of an aircraft may be considered as false stars. Subgraph isomorphism-based algorithms are allergic to false stars; however, the proposed algorithm has good resistance to false stars. The appearance of false stars is random and thus are considered to be distributed uniformly on the image plane. One to five false stars were added randomly with a magnitude of 0 Mv to 6 Mv into the area within the pattern radius of S to obtain 80 noise samples.

For a GRS, 320 noise samples versus one clean sample are generated. Therefore, the noise sample database X' contains $5,045 \times 320 = 1,614,400$ samples, the corresponding labels y' of X' are indices of noise samples in the nSAO catalog. Similarly, clean sample database X includes 5,045 samples in total, the ground truth label set of X is y. The training dataset of RPNet is {X′, y′; X, y}.

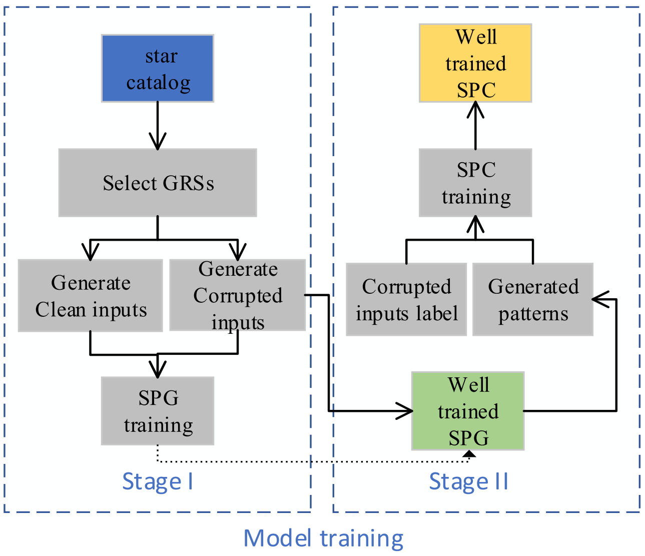


FIGURE 5. Training process of RPNet.

D. TRAINING OF RPNET

The training of the RPNet can be divided into two stages. In the first stage, the pattern generation network learns from noise and clean samples to gain a star pattern generator. In the second stage, noise samples are fed into the previously trained star pattern generator to produce unique star patterns for noise samples. Then, a star pattern classifier is trained with previously generated star patterns and their corresponding labels. The training process is shown in Fig.5.

The pattern generation network is based on the denoising auto encoder (DAE) [21]. It accepts corrupted noise samples as input and trained to undo this corruption to reconstruct the initial input, the hidden layers part can encode the input features thus can be used as a star pattern generator. The detailed architecture and training process of the pattern generation network are shown in Fig.6.

Both the star pattern generator and star pattern decoder contain two fully connected layers: a Relu [22] activation function, followed by the first fully connected layer. When creating the initial input of RPNet, the parameters are $\rho = 6°$, $e = 0.03°$ which are chosen by referring the discretization procedure of the other pattern-based star identification algorithms and many experiment results. Hence, the input dimension of RPNet is $2\lfloor \rho/e \rfloor = 400$. The detailed parameters of the pattern generation network are listed in Table 2.

The pattern generation network is adapted from the DAE. Three key points of improvement are proposed to combine the multi-classification property of the star identification: (1) A new noise-adding strategy is proposed. The traditional way of adding noise is to obtain the initial input from clean simulated star images, then discretize the initial input and finally add noise. This method of adding noise does not correspond to the meaning of the actual noise situation. By contrast, different types of noises are added directly into clean simulated star images first; then, noise samples are

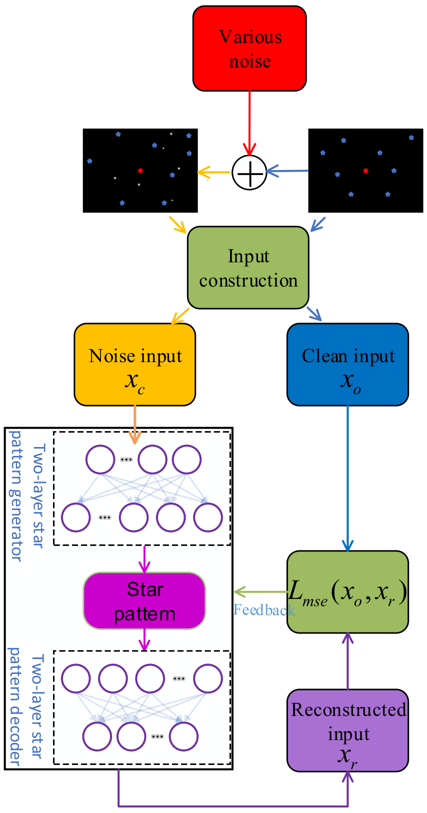


FIGURE 6. Architecture of the pattern generation network and its training strategy.

TABLE 2. Parameters of pattern generation network.

	Input	Layer1	Layer2	Output
SPG	(400,)	W_1 : 400×512 b_1 : 512×1	W_2 : 512×1024 b_2 : 1024×1	(1024,)
SPD	(1024,)	W_2' : 1024×512 b_2' : 512×1	W_1' : 512×400 b_1' : 400×1	(400,)

constructed from these corrupted images. It is a reasonable way of generating various types of noise samples, such as position noise samples, magnitude noise samples, missing star samples, and false star samples. (2) Due to the multi-classification property of the pattern-based star identification algorithm, tens of thousands of categories require that the network has sufficient capacity. An extra hidden layer was added into the encoder and decoder of the traditional DAE to improve its capability to generate a more representative pattern. (3) A sparse encoding [23] strategy is taken to map the initial input into a high-dimensional feature space. Compared with the dimensionality reduction of the traditional DAE that drops some information when encodes, sparse encoding will generate a completer and more robust pattern.

The loss function of pattern generation network is MSE loss, which minimizes the mean-squared error of clean input x_o and reconstructed input x_r:

$$L(x_o, x_r) = \|x - x_r\|_2 . \tag{5}$$

TABLE 3. Hyper-parameters of RPNet training.

	Lr	Epoch	Batch size	L2	Weight decay	Early stop
PGN[1]	0.0001	15	512	0.0001	No	No
PCN[2]	0.001	10	512	0.0001	Yes	Yes

Star pattern classifier in the backend is trained by 10-fold cross-validation strategy to select the best hyper-parameters. All samples are then put into training. The loss function of this part is cross-entropy loss:

$$L(p, y_i) = -\log p_{y_i}, \tag{6}$$

where y_i is the label of the sample and p_{y_i} denotes the probability of the sample the network predicts belongs to y_i. The hyper-parameters of two components of RPNet are listed in Table 3.

E. WEIGHT SEARCHING STRATEGY

For pattern-based star identification algorithms, recognition of a single star image can be simplified to identify the GRS closest to the image center. The pattern of the GRS is completer because it contains most GNSs within the pattern radius compared with other GRSs. Once the specific GRS is identified, its GNSs can be identified easily. However, the wrong identification of the GRS will result in the failure of the star image identification or it will take a huge amount of time to conduct verification and re-identify another GRS. To conquer this problem, a weight searching strategy based on the characteristic of outputting probabilities of the star pattern classifier is proposed. By choosing three stars nearest to the image center, two are selected and verified according to their distances away from the center and the probabilities of their candidate stars. This strategy increases tremendously the confidence ratio of the identified stars and improves the identification rate for GNSs of the main star as well. That is, the strategy enhances significantly the identification rate and processing efficiency for a single star image. The schematic diagram of the algorithm is shown in Fig.7.

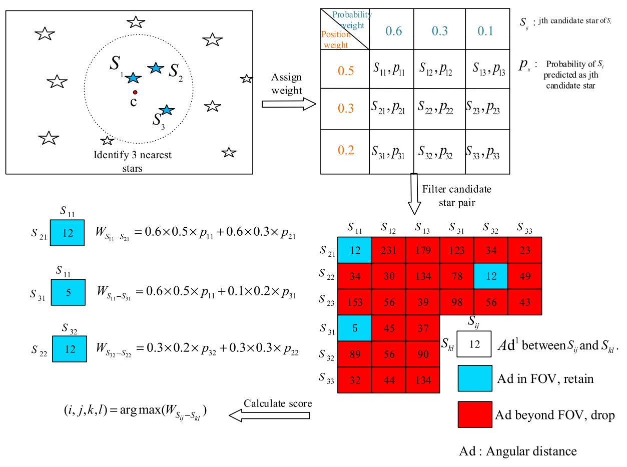
Suppose there are three stars closest to the image center c as S_1, S_2, and S_3. The distances meet the following relationship:

$$S_1c < S_2c < S_3c. \tag{7}$$

The initial inputs constructed by S_1, S_2, and S_3 are fed into the RPNet. For each of the three stars, the RPNet will generate the probability of each star in the nSAO catalog it predicted, then choose three of the most likely stars as candidate stars. Take S_1 for example: the corresponding probabilities of its candidate stars S_{11}, S_{12}, and S_{13} outputted by RPNet are p_{11}, p_{12}, and p_{13}, respectively. They satisfy the following formula:

$$p_{11} > p_{12} > p_{13} > \ldots \tag{8}$$

[1]Pattern generation network
[2]Pattern classification network



Identify 3 nearest stars
Assign weight
Probability weight
Position weight
0.6 0.3 0.1
0.5 0.3 0.2
S_{ij} : jth candidate star of S_i
p_{ij} : Probability of S_i predicted as jth candidate star
Filter candidate star pair
$W_{S_{11}-S_{21}} = 0.6\times0.5\times p_{11} + 0.6\times0.3\times p_{21}$
$W_{S_{11}-S_{31}} = 0.6\times0.5\times p_{11} + 0.1\times0.2\times p_{31}$
$W_{S_{32}-S_{22}} = 0.3\times0.2\times p_{32} + 0.3\times0.3\times p_{22}$
$(i,j,k,l) = \arg\max(W_{S_{ij}-S_{kl}})$
Calculate score
Ad^1 between S_{ij} and S_{kl}.
Ad in FOV, retain
Ad beyond FOV, drop
Ad : Angular distance

FIGURE 7. Weight search algorithm.

The position weights po_1, po_2 and po_3 are assigned to candidate stars from near to far; the probability weight pr_1, pr_2, and pr_3 are assigned to candidate stars based on the probabilities RPNet generates, too. Two groups of weights meet the following relationship:

$$\sum_{i=1}^{3} po_i = 1, \quad \sum_{i=1}^{3} pr_i = 1 \tag{9}$$

Two stars S_i and S_j, are selected randomly and then the candidate stars of S_i and S_j are chosen randomly and denoted by S_{ij} and S_{kl}, where $i, j, k, l \in \{1, 2, 3\}$ and $i \neq k$. The total number of candidate star pairs are $C_3^2 \cdot C_3^1 \cdot C_3^1 = 27$. Assume the attitude of S_{ij} and S_{kl} are (α_1, β_1) and (α_2, β_2), respectively. The angular distance between two candidate stars are as follows:

$$d_{ijkl} = \arccos[\cos(\beta_1) \cdot \cos(\beta_2) \cdot \cos(\alpha_1 - \alpha_2) + \sin(\beta_1) \cdot \sin(\beta_2)] \tag{10}$$

If d_{ijkl} is beyond the maximum angular distance of field of view (FOV), it indicates the chosen candidate star pair is wrong. Otherwise, calculate the score of the qualified candidate star pair based on the following equation:

$$W_{S_{ij} \leftrightarrow S_{kl}} = po_i \cdot pr_j \cdot p_{ij} + po_k \cdot pr_l \cdot p_{kl} \tag{11}$$

Finally, the candidate star pair $S_{ij} \leftrightarrow S_{kl}$ with the highest score is chosen as "confident stars" for guiding stars S_i and S_j, then their GNSs are identified.

III. EXPERIMENT RESULTS AND ANALYSIS

The assessment and verification of the algorithm is performed on a series of simulated star images with different types of noise; Thus, the fundamental parameters of the simulation platform and the key parameters of the algorithm are introduced first. Then, comparisons of performances on simulated star images between RPNet and other pattern-based algorithms are made, and time and space complexities are analyzed as well. Finally, tests on real star images are carried out.

TABLE 4. Parameters of the simulation platform.

Parameters	Value
Focal length	49.277mm
Field of view (FOV)	20°×20°
Resolution	1024×1024pixel
Pixel size	0.012mm
Max. magnitude	6.0 Mv

TABLE 5. Weight distribution.

	i=1	i=2	i=3
Position Weight po_i	0.45	0.30	0.25
Probability Weight pr_i	0.60	0.25	0.15

A. PARAMETERS SETTING

When the boresight pointing of the star sensor is given, a series of star images with different kinds of noise can be produced through the simulation platform. By altering the boresight pointing randomly, simulated star images can be considered as distributed uniformly over the entire celestial sphere. We chose the simulated parameters of some typical pattern-based star identification algorithms for better comparisons. The detailed parameters of the simulation platform are listed in Table 4.

The key parameters of the algorithm and their selection are described below:

(1) Pattern radius ρ. Considering the FOV of a star sensor typically ranges from $\Phi 10°$ to $\Phi 20°$, so the limitation is $2\rho < 20°$. In the meanwhile, original patterns should cover as much information as possible so that ρ cannot be too small, ρ is set to 6° for compromise.

(2) Number of selected GNSs m. The number of neighbor stars within the pattern radius varies from a few to hundreds. To reduce the time and space required to generate the initial input, select as few GNSs as possible under the premise of sufficient information contained for identification. Multiple experiment results indicate the most suitable value is $m = 10$.

(3) Discretization factor e. Discretization factor turns the continuous angular distance into discrete form. A smaller e results in a more elaborate representation but takes more space to store while a larger e may lead to poor discrimination of the input. According to many results of the experiment, $e = 0.03°$ achieved the best performance.

(4) Position weight po_i, $i = 1, 2, 3$ and probability weight pr_i, $i = 1, 2, 3$. Two groups of parameters are selected using grid search algorithm [23]. The best weights obtained are listed in Table 5.

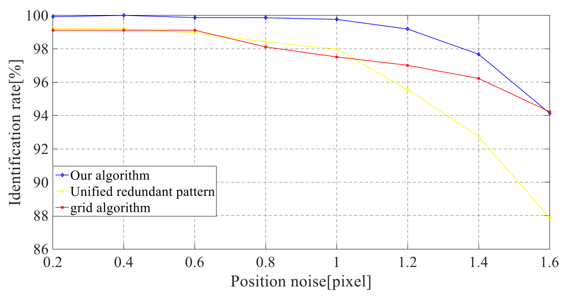


FIGURE 8. Identification rate for different levels of position noise.

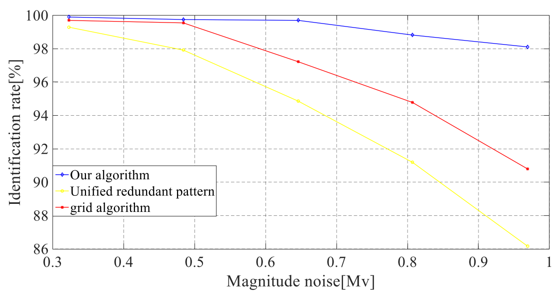


FIGURE 9. Identification rate for different levels of magnitude noise.

B. SIMULATION RESULTS AND COMPARISONS

The proposed algorithm is verified and evaluated under different noise conditions. Two typical pattern-based star identification algorithms are adopted as references for a better comparison: one is the unified and redundant pattern [11] algorithm that concatenates the radial features and the neighbor star features to form a 1-D pattern. It takes the advantages of the polestar algorithm [24] and radial and cyclic pattern algorithm [10] and uses redundant coding strategy to gain a more robust performance. The second is the grid algorithm [9], which constructs a 2-D pattern based on the distribution of neighbor stars. The grid algorithm is re-implemented and tuned to adjust to the simulation platform; the results of the unified and redundant pattern algorithm are taken from related papers.

1) PERFORMANCE TOWARDS POSITION NOISE

The test set is built in the same way as the training data set. For each noise level with a mean of 0 and a standard deviation of 0.2 to 1.6 pixels, 10,000 simulated star images are generated randomly as test data. The performances of the three algorithms are shown in Fig. 8.

The identification rate is obviously near 100% and almost has no attenuation when position noise is less than 1 pixel, indicating that the proposed algorithm has a considerable robust performance towards tiny deformation. Meanwhile, the identification rate of the proposed algorithm is steadily 1% higher than the other two algorithms. When the position continues to increase, the unified redundant pattern algorithm has a cliff fall in identification rate. Although the identification rate of the proposed algorithm drops to 94% when position noise increases to 1.6 pixels, it still outperforms the other two algorithms.

2) PERFORMANCE TOWARDS STAR MAGNITUDE NOISE

The distribution of the magnitude noise is also a Gaussian distribution with a mean of 0 and a standard deviation ranging from 0 to 1Mv. The noise is applied to the magnitude of each star and the number of noise samples for each level of magnitude noise is 10,000. The identification rate of three algorithms can be found in Fig.9.

The identification rate of the proposed algorithm drops from 99.90% to merely 98.11% when the magnitude noise increases from 0.323 Mv to 0.969 Mv. By contrast, the unified and redundant pattern algorithm undergo a large decline from 99.29% to 86.17% while the grid algorithm falls from 99.70% to 90.80% in the same noise condition. Both of algorithms decreased severely when the noise is greater than 0.5 Mv, but the performance of the proposed algorithm remains favorable. The huge gap between the proposed algorithm and the other two algorithms may be caused by their different way of generating patterns. The proposed algorithm involves taking the N closest GNSs to form the initial input; the initial input suffers a little but the other two algorithms are averse to this kind of noise because both algorithms use brightness information to some extent.

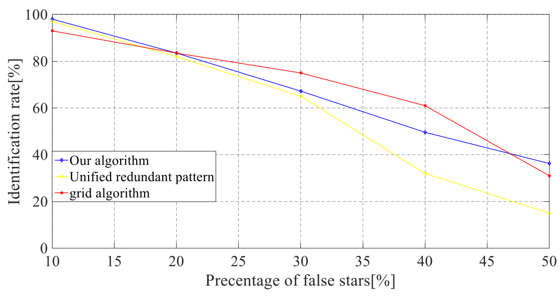


FIGURE 10. Identification rate for different ratios of false stars.

3) PERFORMANCE TOWARDS FALSE STARS

The level of this kind of noise is measured by the percentage of false stars p. For a randomly generated star image, the number of star N_t in the image should be determined and then the number of false star N_f can be calculated by

$$\frac{N_f}{N_f + N_f} = p \tag{12}$$

Finally, N_f false stars are added randomly to form the noise sample. The number of noise samples for each level of false star noise is also 10,000. The performances of the three algorithms are shown in Fig.10.

The performance of the proposed algorithm is better than the two other algorithms when the ratio of false stars is relatively small (p < 20%); the identification rate remains higher than 80%. With the increase in ratio p, the grid algorithm is slightly better than the proposed algorithm. However, when the ratio p reaches 50%, the proposed algorithm outperforms

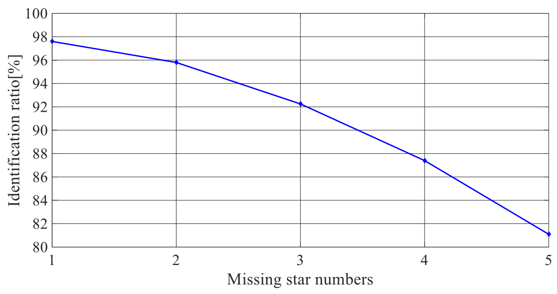


FIGURE 11. The identification rate for different number of missing stars.

the other two algorithms and maintains an identification rate of 35%.

4) PERFORMANCE TOWARDS MISSING STARS

The missing star sample can be obtained easily by dropping the stars in the simulated star image randomly. One to five missing stars were tested and each level contained 10,000 noise samples. However, only the proposed algorithm is estimated because of the lack of experiment results of the other two algorithms. The results for the proposed algorithm are shown in Fig. 11.

The identification rate dropped from 97.6% to 81.09% when the number of missing stars increased from 1 to 5. Only 1 or 2 missing stars are acceptable because too many missing stars may lead to insufficient information for recognition, especially when the total number of stars in an image is scarce.

C. ALGORITHM COMPLEXITY ANALYSIS

The implementations and platforms of star identification algorithms are different, which makes it difficult to compare the running time and actual memory. However, the performance of the algorithms can be judged by analyzing the time and space complexities. Assuming that the size of the star catalog is N, for the most representative subgraph-isomorphism-based algorithm, the triangle algorithm and its varieties construct $C_N^3 = N(N-1)(N-2)/6$ triangle features. The identification procedure involves finding the most similar triangle in the triangle feature database and thus the time and space complexities of this kind of algorithms are both $O(N^3)$. For pattern-based algorithms such as the grid algorithm, for each star in the catalog, a unique pattern should be generated, which means there are N patterns in total. Suppose a pattern takes $O(c)$ space, the space complexity of the algorithm will be $O(cN)$ and the identification can be reduced to a pattern recognition process, and thus, the time complexity of the grid algorithm is $O(N)$. For the algorithm proposed in this paper, patterns are not stored explicitly but in the form of the network parameters which takes $O(c)$ space. However, the star catalog must be stored which means the space complexity of the algorithm is $O(c) + O(N) = O(N)$. The identification involves performing an inference through the networks and thus, the time complexity is $O(1)$.

TABLE 6. Time and space complexities of different algorithms.

	Triangle algorithm	Grid algorithm	RPNet
Time complexity	$O(N^3)$	$O(cN)$	$O(N)$
Space complexity	$O(N^3)$	$O(N)$	$O(1)$

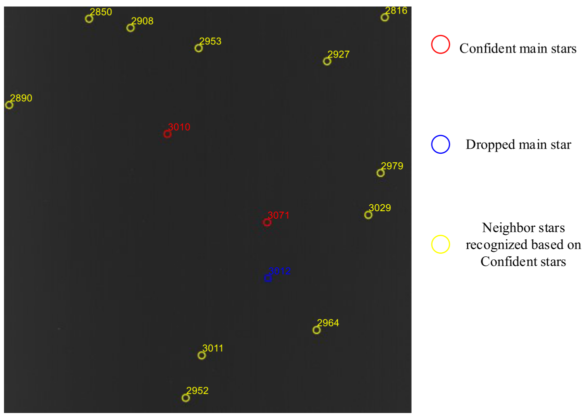


FIGURE 12. Identification results for a real star image.

TABLE 7. Parameters of the ZY3 star sensor.

Parameters	Value
Focal length	43.258 mm
Field of view (FOV)	$20^\circ \times 20^\circ$
Resolution	1024×1024 pixel
Pixel size	0.015 mm
Max. magnitude	6.0 Mv
Principle point (width, height)	(515.247, 513.320) pixel

TABLE 8. Identification rate on ZY-3 real star images.

	Grid	Unified redundant pattern	RPNet
Identification rate	98.21%	94.64%	98.45%

Table 6 lists the time and space complexities of different algorithms for a better comparison.

D. PERFORMANCE ON REAL STAR IMAGE

A total of 323 real star images downloaded from the Chinese civilian satellite ZY-3 were tested to verify the performance in a real scene. The parameters of the star sensor are shown in Table 7 and the recognition results of one real star image are shown in Fig.12.

Five images were failed to be identified because of the low accuracy of the star centroids positioning algorithm. Many stars in these images were too dim to be extracted. The proposed algorithm achieved a 98.45% identification accuracy on this dataset. The comparison with other algorithms is listed in TABLE 8 [11], [12].

IV. CONCLUSION

A representation learning based neural network (RPNet) for robust and efficient star identification tasks is proposed in this paper. The RPNet learns from a huge amount of data and creates unique patterns for GRSs and then classifies the previously created patterns. The characteristics of the RPNet are listed below:

(1) The algorithm is quite robust towards small deformations of the star image and star magnitude noise hardly affects the identification rate.

(2) A few false stars or missing stars do not hinder the algorithm from maintaining an acceptable performance. However, too many false stars or missing stars will significantly decrease the identification rate. This issue is a common problem for pattern-based algorithms.

(3) The algorithm is fast in theory; it achieves a time complexity of only O(1) and the space complexity is O(N). These numbers are far smaller than the triangle algorithm whose time and space complexities are both O(N ^{3}).

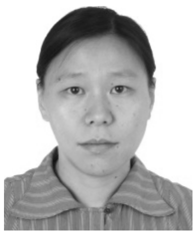
(4) The architecture of the RPNet is relatively simple. After training on the computer, RPNet is feasible and easy to re-implement on embedded systems such as the FPGA and DSP.

In conclusion, the algorithm proposed in this paper is fast and highly efficient. The identification rate of the algorithm outperforms other pattern-based algorithms on average. The RPNet is also a bold and successful attempt of artificial intelligence in the aerospace field.

REFERENCES

[1] C. C. Liebe, "Accuracy performance of star trackers—A tutorial," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 587–599, Apr. 2002.
[2] C. C. Liebe, "Pattern recognition of star constellations for spacecraft applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 7, no. 6, pp. 34–41, Jun. 1992.
[3] B. Quine and H. F. Durrant-Whyte, "Rapid star-pattern identification," *Proc. SPIE*, vol. 2739, pp. 351–360, Jun. 1996.
[4] E. J. Van der Heide, "American institute of aeronautics and astronautics 54th international astronautical congress of the international astronautical federation, the international academy of astronautics, and the international institute of space law—Bremen, Germany (29 Sep. 2003 - 03 October 2003)]," in *Proc. 54th Int. Astron. Congr. Int. Astron. Fed., Int. Acad. Astronaut., Int. Inst. Space Law*, 2003, p. A-5.
[5] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, 2004.
[6] G. Wang, J. Li, and X. Wei, "Star identification based on hash map," *IEEE Sensors J.*, vol. 18, no. 4, pp. 1591–1599, Feb. 2018.
[7] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 441–456, Apr. 2008.
[8] J. Li, X. Wei, and G. Zhang, "Iterative algorithm for autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 536–547, Jan. 2015.
[9] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 202–213, Jan. 1997.
[10] G. Zhang, X. Wei, and J. Jiang, "Full-sky autonomous star identification based on radial and cyclic features of star pattern," *Image Vis. Comput.*, vol. 26, no. 7, pp. 891–897, 2008.
[11] J. Jiang, F. Ji, J. Yan, L. Sun, and X. Wei, "Redundant-coded radial and neighbor star pattern identification algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 2811–2822, Oct. 2015.
[12] L. Sun, J. Jiang, G. Zhang, and X. Wei, "A discrete HMM-based feature sequence model approach for star identification," *IEEE Sensors J.*, vol. 16, no. 4, pp. 931–940, Feb. 2015.
[13] Y. Zhao, X. Wei, J. Li, and G. Wang, "Star identification algorithm based on K–L transformation and star walk formation," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5202–5210, Jul. 2016.
[14] J. Hong and J. A. Dickerson, "Neural-network-based autonomous star identification algorithm," *J. Guid., Control, Dyn.*, vol. 23, no. 4, pp. 728–735, 2000.
[15] C. S. Lindsey, T. Lindblad, and A. J. Eide, "Method for star identification using neural networks," *Appl. Sci. Artif. Neural Netw.*, vol. 3077, pp. 471–479, Apr. 1997.
[16] P. Alvelda and M. A. S. Martin, "Neural network star pattern recognition for spacecraft attitude determination and control," in *Proc. Adv. Neural Inf. Process. Syst.*. 1989, pp. 314–322.
[17] L. Paladugu, E. Seisie-Amoasi, B. G. Williams, and M. P. Schoen, "Intelligent star pattern recognition for attitude determination: The 'Lost in space' problem," *J. Aerosp. Comput., Inf., Commun.*, vol. 3, no. 11, pp. 538–549, 2006.
[18] *SAO Star Catalog*. [Online]. Available: http://tdc-www.harvard.edu/catalogs/sao.html
[19] G. Zhang, *Star Identification: Methods, Techniques and Algorithms*. Springer, 2019, pp. 48–53.
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Feb. 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167
[21] P. Vincent, H. Larochelle, and Y. Bengio, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 1096–1103.
[22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[23] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 801–808.
[24] E. Silani and M. Lovera, "Star identification algorithms: Novel approach & comparison study," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1275–1288, Oct. 2006.

LIKAI XU received the bachelor's degree from the University of Electronic Science and Technology of China (UESTC), in 2016. He is currently pursuing the Ph.D. degree with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University (BUAA), China. His research interests include image processing and pattern recognition.



JIE JIANG received the bachelor's, master's, and Ph.D. degrees from Tianjing University, in 2000. She is currently a Professor with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University (BUAA), China. Her research interests include image processing and star sensors.

LEI LIU received the bachelor's and master's degrees from the University of Science and Technology Beijing (USTB), in 2014. He is currently pursuing the Ph.D. degree with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University (BUAA), China. His research interests include image processing and pattern recognition.