Design and Optimization of Star Recognition Algorithm
Based on Hierarchical CNN

Wentao Lu¹, Gengxin Hua*, Yunfu Zhao¹, Jiantao Zhou¹

¹Beijing Institute of Control Engineering, China Academy of Space Technology, Beijing, 100094, China
*Corresponding author’s e-mail: huagengxin@cast.cn

Abstract. With the rapid-development of AI technology, artificial intelligence algorithms for the aerospace applications have shown very good simulation performance in many areas. Among the spaceborne application fields, star identification can be seen as a typical pattern recognition process. It’s also the key part of attitude determination of the satellites, which requires the algorithm to be robust and efficient due to the limited computing and storing resources of the spaceborne computers. Nevertheless, most of the previous algorithms are not possible to be applied in practical due to the reasons above. This article proposes a strategy of constructing ‘net-structure’ images of stars to build the datasets for training and testing. Besides, a hierarchical convolutional neural network (CNN) with a small size is also designed. It performs good results on robustness and efficiency in the experiments. In the end, a method of fusing the Conv layers and the batch normalization (BN) layers is also adopted to further accelerate the algorithm.

1. Introduction
Artificial intelligence (AI) algorithms for the aerospace applications have shown very good simulation performance in many areas such as fault diagnosis [1], rendezvous and docking.

Star map identification [2][3] is a core technique for star sensors to determine and modify its attitude while executing the space missions such as earth observation, celestial exploration and so on. As the of star map images are in greyscales, the processing of the images also can be seen as a image pattern recognition process. Some algorithms for star map identification based on AI methods such as neural networks were already proposed [4][5]. But they failed to meet of requirements of robust and real-time performance because the information of the greyscale star images is too scarce, which makes the constructing of data sets demands for a strategy to enhance and increase the messages shown in the images. And in the environment of aerospace, it’s possible that the star sensor gets noise information such as missing stars or fake stars, so these aspects shall also be considered in dataset’s setting to enhance the robust of the algorithm. A method of ‘net-structure’ and it’s construction will be shown in the article. Besides, it’s even harder for the algorithms to achieve the requirements on the spaceborne platform that equipped with highly limited computing and storing resources. So the scale of the network shall be limited, and the speed of interfering is also a key point to be considered. As shown in figure1.
2. Construction of ‘net-structure’ images and datasets

Due to the messages of the star maps are too scarce, due to the images are shown in greyscale, some of the neural network algorithms turns the star messages to be extracted in to vectors[6] to the inference part to finish matching. But it’s obvious that the CNN is more capable in recognizing images[7], according to this, this article constructs the main star of a star image and its neighbourhood stars in to a ‘net-structure’ image with different colors of links that can be recognized by the CNN.

2.1. Construction of images

The original star coordinates are based on the SAO [9](Smithsonian Astrophysical Observatory) All Sky Catalog of 2500 primary star points with magnitudes below 6 and their neighbourhood stars (at least four) within 12°*12°of the FOV field of view. On this basis, the datasets are constructed as follow steps:

Step 1: Using white lines to connect the main star to every neighbouring stars in turn. As in Figure 2.

Step2: Determine the coordinates (X1,Y1) and (X2,Y2) of the equatorial latitude and longitude of the neighbourhood stars (represented by stars A and B) in the celestial coordinate system, and to calculate their direction vectors s1 and s2 in the image plane coordinate system, using the following formula.

\[
s_1 = \frac{1}{\sqrt{x_1^2 + y_1^2 + f^2}} \begin{pmatrix} x_1 \\ y_1 \\ -f \end{pmatrix}
\]

Figure 2. step 1

Figure 1. Flowcharts of the proposed algorithm
Where $f$ is the focal length as:

$$f = \frac{N_x \cdot d_h}{2 \tan(FOV_x / 2)} = \frac{N_y \cdot d_v}{2 \tan(FOV_y / 2)}$$

Step 3: Based on the direction vectors of the neighbourhood stars (presenting as star A, star B), the angular distance between these two neighbourhood stars can be calculated with the following formula.

$$d_{12}^{m} = \cos^{-1}\left(\frac{s_{1} \cdot s_{2}}{|s_{1} \cdot s_{2}|}\right)$$

Step 4: The adjacent neighbourhood stars are also connected in series with lines in clockwise order. The color of the lines is no longer white, but to connect the stars with lines of different colors according to the range of the angular distance between the two adjacent stars, and the correspondence between the size of the angular distance and the color of the line is as follows.

**Table 1. Correspondence Between Angle Distances and Colors**

| Angle distance/° | 0~1.5 | 1.5~3 | 3~5 | 5~7 | 7~9 | 9~11 | 11+ |
|------------------|-------|-------|-----|-----|-----|------|-----|
| Color            | red   | green | blue| yellow| brown| purple| pink|

**Figure 3. an example of constructed image**

### 2.2. Construction of datasets

Finishing the method of drawing the ‘net-structure’ star map images, what comes as follow is the construction of the relevant datasets for the neural network, which includes the training set and the test set.

The training set is used to modify the model to the neural network model. According to the design criteria of the star map recognition algorithm, the 2500 ‘net-structure’ model maps described in the previous section are to be rotated angularly, and the design approach taken by this article is to rotate each star map continuously by 30°, as shown in Figures 4 until it is rotated by 360°, resulting in a total of 12 training sets with different angular rotations.
Based on this, for each different angle of the star map, one or two neighboring stars are randomly deleted or pseudo-stars are added at any position (as in Figure 4). Five random operations are performed for each different angle of the mesh model, and the process in 2.1 is repeated to redraw the mesh model map after the changes occur.

After the above process, each original star images produced 60 ‘net-structure’ images as the training datasets, and all of the 2500 original star maps produced a total of 150,000 ‘net-structure’ images as the training datasets.

The test set is constructed by aiming on the robustness of the neural network model to angular rotation after fitting the training set, so that each original star map is rotated 70° three times (Figure 5) to avoid overlapping with the training set, and on this basis, each rotated star map is adjusted by adding fake stars or missing neighbourhood stars randomly for three times and reconstruct the ‘net-structure’ star images. Thus, a total of 12 ‘net-structure’ star images are generated for each star map as
the testing dataset, and a total of 30,000 ‘net-structure’ star images are constructed from 2500 original star maps as the training set.

3. Designing of hierarchical convolutional neural network model

3.1. Structure of the CNNs

Convolutional neural network (CNN) is firstly proposed by Lecun et al. [8], which is a typical feed-forward structured neural network with structural features that can be superimposed on multiple layers to complete learning, and has shown outstanding capabilities in image classification, segmentation, and object-specific detection. A standard convolutional neural network usually consists of a convolutional layer, a pooling layer, a batch normalization layer, an activation layer, and a fully connected layer. Compared with traditional neural networks, it has a smaller number of weights and a simpler structure, and maintains good robustness in recognizing rotations, translations, and light distortions of images.

In the process of designing the neural network for star map recognition in this paper, due to the limitation of types and numbers of mathematical operations that can be performed by the spaceborne computers. Therefore, the number of network layers should not be too much, the convolution kernel in the process of convolution and operation should not be too large, and the functions selected for the pooling and activation layers should not be too complex. Such a design idea will lead to a slight limitation of the accuracy of star map recognition, but the inference speed and portability will be guaranteed.

The initial design of the network structure is: 5 convolutional layers, with 3*3 convolutional kernels, Batch Normalization after each convolutional layer, and a Relu function is adopted for the activation function.

\[ f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \quad (5) \]

The structure of the convolutional neural network is constructed as below(figure 7.):
3.2. Building of the hierarchical CNN

In the process of drawing, rotating, adding fake stars, and losing neighbourhood stars of the ‘net-structure’ images. It is likely to lead to "similar" images, and these similar images are likely to create error results in the process of matching recognition. Therefore, this article considered to select an appropriate feature information from the 2500 labels as a kind of clustering classification, so as to build different subnets according to the different categories of the feature. It integrates the design idea of building subnets based on clustering features.

After analysis on different clusters of the ‘net-structure’ images, we selected the angle distance between the main star and the nearest neighbour star which is the nearest to it, as with the minimum angle distance, as the proof of clustering analysis of the CNN. The star maps are classified into 10 groups, and the CNN is modified into a hierarchical CNN with 10 subnets. The previous 9 subnets contains 256 labels in the softmax layer. And the last one contains 296, each two adjacent nets has 10 coincident stars to enhance robustness. The classification table is as follow:

| Category number | class1       | class2       | class3       | class4       | class5       |
|-----------------|--------------|--------------|--------------|--------------|--------------|
| Angular distance/° | 0.206-0.486 | 0.471-0.408 | 0.701-0.913  | 0.905-1.129  | 1.122-1.329  |
| Number of main stars | 256        | 256          | 256          | 256          | 256          |

| Category number | class6       | class7       | class8       | class9       | class10      |
|-----------------|--------------|--------------|--------------|--------------|--------------|
| Angular distance/° | 1.317-1.532 | 1.529-1.813  | 1.805-2.157  | 2.148-2.623  | 2.611+       |
| Number of main stars | 256        | 256          | 256          | 256          | 296          |

In the process of constructing the hierarchical network, the idea of completely retraining the 10 subnetworks to build them separately cannot be used in order to ensure that the overall weight volume of the network cannot be too large. Instead, the weight parameters of the completed training network in 3.1 are kept unchanged, and the calibration of the angular distance information of the main and nearest neighbourhood stars of the test set is added to the softmax classifier of the corresponding class in the inference process. Thus A 10-subnet hierarchical neural network is constructed.

3.3. Fusion of the Conv layers and the BN layers

The first Batch Normalization (BN) layer settings is a method proposed in 2015 with the following role and rationale:
It can speed up the convergence and alleviate the problem of gradient disappearance, which plays a regularization effect to some extent. BN layer is used to make normalization on the network input with the formulas as in Eqs. (6) to (9).

\[
\mu = \frac{1}{m} \sum_{i=1}^{m} x_i
\]  

\[
\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2
\]  

\[
\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}
\]  

\[
\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}
\]

From this, it can be seen that the computation process of BN layer is ① calculating the mean value \(\mu\); ② calculating the variance \(\sigma^2\); ③ normalize the data; ④ introduce the parameters \(\gamma\) and \(\beta\) to do scaling and translation, which are learned by the network.

A fully connected layer with 200 filters is set up in the classification stage, and the final output unit of the classifier is a Softmax function for the final numerical processing and label output, corresponding to a total of 2500 labels.

Fusing the Conv layer and BN layer is to bring the convolutional layer formula into the BN layer formula to obtain equation (10), from the equation we can see that the fusion of the Conv layer and BN layer only replaces the convolutional kernel weights and biases, and does not increase the computation of the convolution. The computation of the BN layer is eliminated, and the computation of the BN layer is reduced. The reduction of computation is achieved without any loss.

\[
\hat{x} = \frac{\gamma_1 x + \beta - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta
\]  

Figure 9. comparison of before and after fusion

The left plot is the calculation before optimization, and the right plot is the calculation after optimization.

The left figure includes Conv, BN, ReLU, and Pooling layers, where the Conv, BN, and ReLU layers are fused into the Conv layer in the right one. To reduce the overhead of GPU core function calls. It reduces 3 GPU calls to 1 GPU call, and since each GPU call requires copying data from CPU to GPU and then copying data from GPU back to CPU after the call is completed, the number of GPU calls is reduced, so the data copying between CPU and GPU is also reduced, thus reducing the time consumption and accelerating the forward inference of the network.
4. Experiments and Analysis
According to 3.1~3.3, this article proposed 3 different CNNs based on ‘net-structure’ star images as follow: 1). The CNN before clustering analysis. 2). The hierarchical CNN. 3). The fused hierarchical CNN. All of them were implemented with the same environment of Python on a 16-core Intel I7 processor, which has a frequency of 3.2GHz, RAM of 24GB, and L3 cache for 8MB.

![Figure 10. Comparation between accuracy of origin and hierarchical CNN](image)

According to the Figure 10 we can see that the accuracy of inference increased from 88.92% to 95.12%, achieved a huge amount of improvement and achieved the requirement for robustness, besides, the memory of consumption is 2.38MB, also fits the limitation of computation resources of spaceborne computers. The average identification time for the hierarchical CNN algorithm is 20.7ms, the method of fusion the Conv layers and the BN layers is taken up to improve this index.

![Figure 11. Comparation between accuracy of all three CNNs](image)

From Figure 11 we can find that after fusion the Conv layers and the BN layers, the accuracy of the whole CNN and the 10 subnets are decreased for a little degree less than 1%. But the average inference time for identification decreased from 20.7ms to 12.8ms, with a high scale accelerate for 38.2%, and the occupation for memory decreased to 2.32MB. The tradeoff between accuracy and efficiency meets a balance by the fusion method.
5. Conclusion
This article studies on the construction and optimization of a star recognition algorithm. The method of creating training datasets and testing datasets is to turn the star map image into a ‘net-structure’ star image, and the CNN was optimized by two steps including turning into a hierarchical structure and adopted a method of layers fusion. By these ways, the accuracy of the algorithm is enhanced up to 95.12%. Besides, the average time of referring is also shortened to 12.8ms. These facts made this algorithm fits the requirements of accuracy and efficiency of the star identification. In further research, how to improve the accuracy and robustness of the algorithm while keep the speed and efficiency will be a meaningful work.

References
[1] Jiao J, Zhao M, Lin J, Liang K. A Comprehensive Review on Convolutional Neural Network in Machine Fault Diagnosis[J]. Neurocomputing, 2020, 417.
[2] Zhang Guangjun. Star Map Recognition [M]. Beijing:National Defense Industry Press, 2011.
[3] Padgett C, Kreutz-Delgado K (1997) A grid algorithm for autonomous star identification.IEEE Trans Aerosp Electron Syst 33(1):202-213
[4] Lindsey C, Lindblad T (1997) A method for star identification using neural networks. SPIE 3077 : 471-478
[5] Li C, Li K, Zhang Y et al. (2003) Star identification based on neural networks. Chin Sci Bull 48(9):892-895
[6] J. Feilong, J. Jie, and Y. Jinyun A redundant-coded radial and neighboring star pattern identi- Fication IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 4, pp. 2811–2822, Oct. 2015.
[7] Liu L, Jiang J, Zhang G. Star Identification Based on Spider-Web Image and Hierarchical CNN[J]. IEEE Transactions on Aerospace and Electronic Systems, 2019, PP(99):1-1.
[8] C. Farabet, C. Poulet, J. Y. Han, and Y. LeCun, “CNP: An fpga-based processor for con- volutional networks,” in Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on, pp. 32–37, IEEE, 2009.
[9] M.W.Myers,Sande,“Sky2000-masterstarcatalog-starcatalogdatabase,”Bulletin- AmericanAstronomicalSociety,vol.191,no.128,1997.