1-1-2022

Fine-grained ship image recognition based on BCNN with inception and AM-Softmax

Zhilin Zhang

Ting Zhang

Zhaoying Liu

Peijie Zhang

Shanshan Tu

See next page for additional authors

10.32604/cmc.2022.029297

Zhang, Z., Zhang, T., Liu, Z., Zhang, P., Tu, S., Li, Y., & Waqas, M. (2022). Fine-grained ship image recognition based on BCNN with inception and AM-Softmax. CMC-Computers, Materials and Continua, 73(1), 1527-1539. https://doi.org/10.32604/cmc.2022.029297

Computers, Materials & Continua
DOI: 10.32604/cmc.2022.029297
Article

Tech Science Press

Fine-grained Ship Image Recognition Based on BCNN with Inception and AM-Softmax

Zhilin Zhang[1], Ting Zhang[1], Zhaoying Liu[1,*], Peijie Zhang[1], Shanshan Tu[1], Yujian Li[2] and Muhammad Waqas[3]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2]School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin, 541004, China
[3]School of Engineering, Edith Cowan University, Perth WA 6027, Australia
*Corresponding Author: Zhaoying Liu. Email: zhaoying.liu@bjut.edu.cn
Received: 01 March 2022; Accepted: 01 April 2022

Abstract: The fine-grained ship image recognition task aims to identify various classes of ships. However, small inter-class, large intra-class differences between ships, and lacking of training samples are the reasons that make the task difficult. Therefore, to enhance the accuracy of the fine-grained ship image recognition, we design a fine-grained ship image recognition network based on bilinear convolutional neural network (BCNN) with Inception and additive margin Softmax (AM-Softmax). This network improves the BCNN in two aspects. Firstly, by introducing Inception branches to the BCNN network, it is helpful to enhance the ability of extracting comprehensive features from ships. Secondly, by adding margin values to the decision boundary, the AM-Softmax function can better extend the inter-class differences and reduce the intra-class differences. In addition, as there are few publicly available datasets for fine-grained ship image recognition, we construct a Ship-43 dataset containing 47,300 ship images belonging to 43 categories. Experimental results on the constructed Ship-43 dataset demonstrate that our method can effectively improve the accuracy of ship image recognition, which is 4.08% higher than the BCNN model. Moreover, comparison results on the other three public fine-grained datasets (Cub, Cars, and Aircraft) further validate the effectiveness of the proposed method.

Keywords: Fine-grained ship image recognition; Inception; AM-softmax; BCNN

1 Introduction

Fine-grained image recognition (FGIR) refers to the recognition of different subclasses of the same category [1], for example, the recognition of "freighters" and "merchant ships". Currently, traditional image recognition tasks have achieved great success. However, due to the small inter-class and large intra-class differences, the performance of fine-grained image recognition is not so satisfying. As a major carrier of marine traffic and transport, fine-grained ship image recognition has attracted

more and more attention, it has been widely applied for maintaining maritime safety, such as maritime traffic monitoring and maritime search, thus to improve the capability of coastal defense and early warning [2,3]. However, for ship targets, the shapes and structures are similar from one category to another, and there are also rich diversity components between the same classes, thus making fine-grained ship recognition a very challenging task.

Traditional methods of fine-grained image recognition of ships mainly use manually designed feature extraction algorithms for feature matching [4,5]. They cannot fully utilize the information contained in the dataset to extract the distinctive features of the objects, resulting in the performance of fine-grained recognition is limited. Furthermore, all of these methods have a low generalization capacity. With the development of deep learning techniques, many deep learning models have been developed to improve the accuracy by learning better feature representations from the dataset automatically based on convolutional neural networks (CNN) [6]. Among those deep models, the bilinear convolutional neural network (BCNN) [7] demonstrates satisfying performance for fine-grained image recognition. The BCNN typically utilizes two parallel branches of the VGGNet network [8] to retrieve the features of each image position, then an outer product operation to integrate features, and update the training network by end-to-end. However, BCNN has the following two deficiencies. 1) The two branches of the network only consist of 3×3 convolutional kernels, and generally small convolutional kernels ignore certain global information [9]. 2) The BCNN uses the Softmax loss function, which has a weak ability to activate subtle features, and it is likely to misclassify certain images with particularly small inter-class differences [10,11].

To enhance the performance of fine-grained recognition of ships image, we developed a fine-grained image recognition network for ships based on BCNN with Inception and AM-Softmax, which improved the BCNN from two perspectives. First, to gather global information, we replaced one branch of the BCNN with Inception module, this is helpful to aggregate feature information on a large scale and increase the ability of global information extraction. Second, to activate the distinctive characteristics between different classes, and extend the inter-class distance while reducing the intra-class distance, we introduced the AM-Softmax function, which effectively activate the differences between the ship classes by adding an additive margin to different decision boundaries. Moreover, we construct a fine-grained ship image dataset containing 47,300 images belonging to 43 categories. The key advantages and major contributions of the proposed method are:

- To extract global information, we design Inception modules and use them to replace a branch of the BCNN network.
- To better activate the features between fine-grained images, we introduce AM-Softmax, which can by adding an additive margin to different decision boundaries.
- Based on the existing dataset, we construct a richer ship dataset.

The rest of the paper is organized as follows. Section 2 summarizes related work. The proposed method is described in Section 3. Detailed experiments and analysis are conducted in Section 4. Section 5 concludes the paper.

2 Related Work

In recent years, many fine-grained image recognition methods have been developed, and these methods can be roughly classified into three main paradigms, i.e., fine-grained recognition with localization-classification subnetworks, with end-to-end feature encoding and with external information. Fine-grained with localization-classification subnetworks approaches design a localization

subnetwork for locating these key parts [12], while later, a classification subnetwork follows and is employed for recognition of the key parts, such as Part-based CNN [13], Mask-CNN [14]. Those approaches are more likely to find distinguished parts [15,16], and require more annotation information. End-to-end feature coding methods, by designing powerful models, learn a more discriminative feature representation. The most representative method among them is BCNN. Beyond the two paradigms, another paradigm is to leverage external information, such as web data and multi-modality data, to further assist fine-gained recognition [17,18].

The BCNN extracts feature via a network of two parallel branches, each of which is VGG16, and an outer product operation is performed on the two outputs. The outer product operation completes the feature fusion at each location, which can capture discriminative features. The structure of the VGG16 network is relatively simple, and each layer of the network uses small-sized convolutional kernels. By increasing the depth of the network, rich feature information can be obtained and the overall performance of the network can be improved. However, small convolutional kernels ignore some global information when extracting feature layer by layer, and just increasing the depth of the network brings a few problems, such as overfitting, gradient vanishing, and training difficulties. The Inception network [19] proposed by Szegedy is wider and more efficient, it uses larger scale convolutional kernels to extract global information, and reduces the number of parameters by exploring the factorization of convolutional kernels.

In recent years, besides the commonly used Softmax, there are various loss functions [20] have been proposed that can optimize the distance between classes. The L-Softmax [21] was first proposed as an angle-based loss function, which can reduce the angle between the feature vector and the corresponding weight vector by introducing a parameter m. The A-Softmax [22] performs a normalization operation on the weights, and by adding a large angle margin, the network more focused on optimizing the angles of features and vectors. The Cosface [23] reformulates the Softmax as a cosine loss, then it can remove radial variations by L_2 normalizing both features and weight vectors, and can further maximize the decision margin in the angular space by introducing a cosine margin term. By introducing additive angles to the decision boundary, the Arc-Softmax [24] maximizes the classification bounds in the angle space.

3 The Proposed Method

In this paper, based on the BCNN framework, we design a fine-grained ship image recognition network by introducing Inception and AM-Softmax. By adding the Inception module to a branch of BCNN, it is helpful to enhance the ability of the whole network to extract global information. Meanwhile, the network uses the AM-Softmax function to learn decision boundaries among the different classes, which can increase the inter-class distance and reduce the intra-class distance.

The architecture of the proposed method is illustrated in Fig. 1. There are two parallel branches, one of them uses VGG16 to extract features from local information, and the other branch introduces the Inception module to extract features from global information. The results of both branches are combined using the outer product and average pooled to get the bilinear feature representation. Then the bilinear vector is passed through a linear classifier and AM-Softmax layer to obtain class predictions. Finally, the cross-entropy loss function is used to guide and optimize the training of the network.

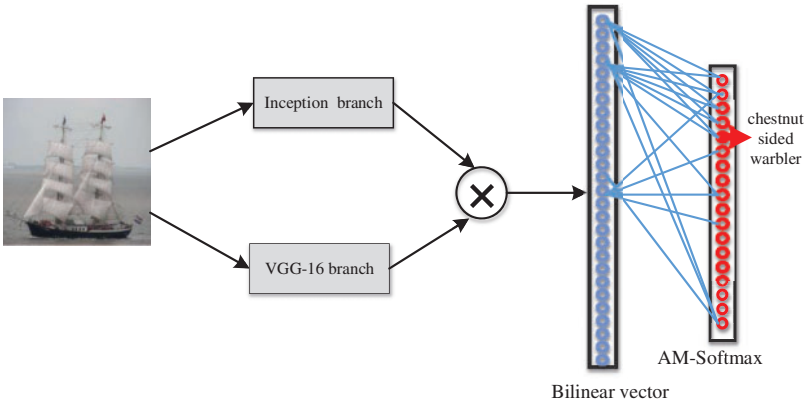


Figure 1: Network architecture

3.1 The Inception Branch

In the VGG16 network, small scale convolutional kernels are used to get local feature information more easily, but it is difficult to extract global information features. Meanwhile, network with larger scale convolutional kernels usually requires a large amount of calculation. According to the literatures, we know that the Inception network can extract global information and reduce the calculation consumption while using larger scale convolutional kernels during the network. Inspired by the Inception network, we design three modules, IncepA, IncepB, and IncepC, to extract global information. These modules, as shown in Fig. 2, have a convolutional kernel size of 3×3, 5×5, and 7×7, respectively.

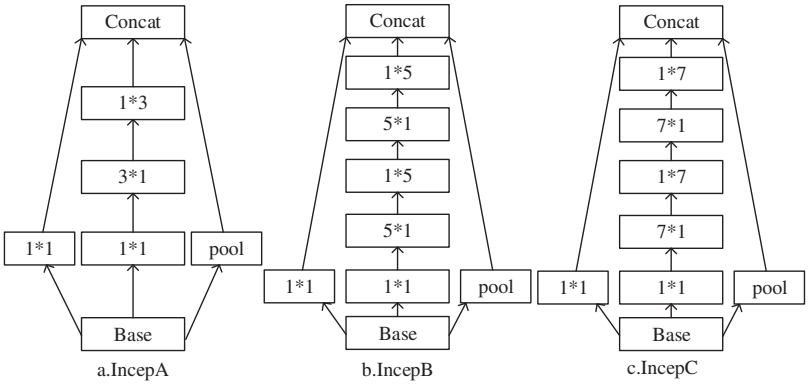


Figure 2: Three Inception modules

In all three modules, 1×1 convolutional kernel and a pool operation are performed, which can help the modules reduce the amount of calculation. Then 3×3 convolutional kernel is decomposed into 1×3 and 3×1 vector kernels. In the IncepB and IncepC, 1×5 or 1×7 vector kernels are stacked two times. Finally, all the three components are concatenated. Those cascaded vector kernels can roughly achieve the effect of large-scale convolution kernels. By decomposing the large-scale kernels, it can effectively reduce the total number of parameters without increasing the calculation consumption.

Once the three modules have been designed, the Inception branch network is built as shown in Fig. 3. The VGG16 branch network is shown in Fig. 4, and it consists of 13 convolutional layers and

3 fully connected layers. In the Inception branch network, we also used 13 convolutional layers, just using 3 modules instead of 9 convolutional layers in VGG16.

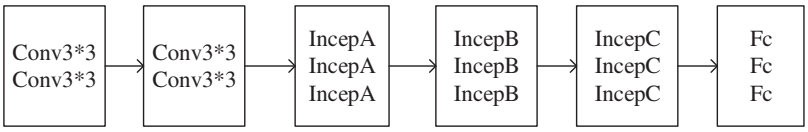


Figure 3: Inception branch network

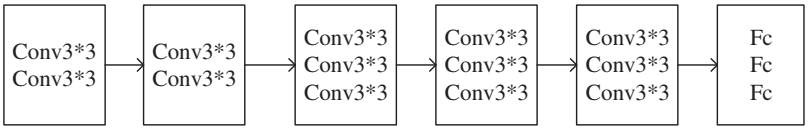


Figure 4: VGG16 branch network

By using Inception modules in the Inception branch, large-scale convolutional kernels are added to this network. Furthermore, the decomposed kernels help the network to extract much richer global features without increasing the overall computational effort.

3.2 Additive Margin Softmax

The BCNN uses the original Softmax loss function. If ignoring the bias, the formulation of the original Softmax loss is defined as

$$L_{\text{softmaxLoss}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_i^T x_i}}{\sum_{j=1}^{c} e^{W_j^T x_i}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{||W_i||.||x_i||\cdot\cos\left(\theta_{y_i,i}\right)}}{\sum_{j=1}^{c} e^{||W_j||.||x_i||\cdot\cos\left(\theta_{y_j,i}\right)}} \tag{1}$$

where x_i denotes the feature of the i-th sample, belonging to the y_i-th class, W_j is the j-th column weight of the last fully connected layer. The $W_i^T x_i$ is called as the target logit of the i-th sample, θ represents the angle between the weight and input value. Then, the weights and inputs in the above Eq. (1) are normalized (making $||W_j||$ and $||x_i||$ to be 1), we obtain the modified expression as

$$L_{\text{modified}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\cos\left(\theta_{y_i,i}\right)}}{\sum_{j=1}^{c} e^{\cos\left(\theta_{y_j,i}\right)}} \tag{2}$$

If a two-dimensional feature is used as an example and the feature is represented in a circle, a geometric interpretation of the above equation can be clearly illustrated as shown in Fig. 5. Where, W_1 and W_2 can be considered as center vectors of the two class; θ_1 and θ_2 represent the angles between sample vector x and the two center vectors. P_0 represents the decision boundary generated by the Softmax function for the two classes, and accordingly, P_1 and P_2 are generated by AM-Softmax. If $\cos(\theta_1) < \cos(\theta_2)$, the feature can be identified as category 1. As a result, the decision boundary between the two classes has only one P_0, i.e.,$\cos(\theta_1) = \cos(\theta_2)$. If there is only one decision boundary, some special samples, with the intra-class distance being larger than the inter-class distance, can be easily misclassified.

The ship images are characterized by small differences between classes and large differences within classes. It is necessary to design an activation function which increases the distance between classes and decreases the distance within classes.

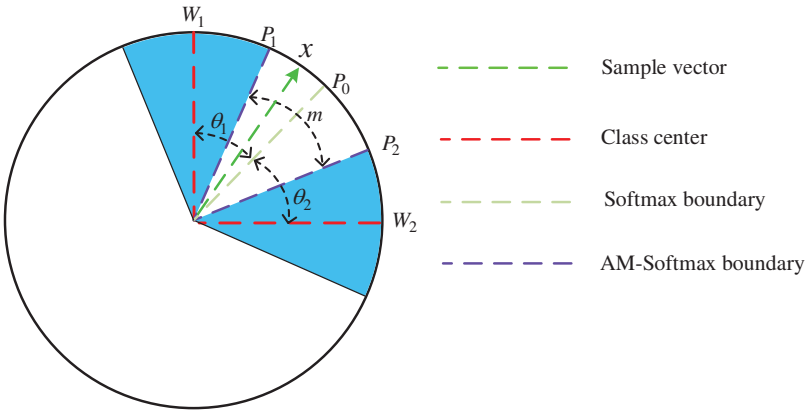


Figure 5: Decision boundaries of Softmax and AM-Softmax

To increase the inter-class distance and decrease the intra-class distance, a margin m can be explicitly added to the decision boundaries of the categories. That is, based on the Softmax loss function, the decision boundary has two decisional surfaces P_1 and P_2. The boundary for category 1 is $\cos\theta_1 - m = \cos\theta_2$, category 2 is $\cos\theta_1 = \cos\theta_2 - m$. Then, in this paper, we assume that the norm of both W_{y_i} and x_i are normalized to 1, the Additive Margin Softmax loss function can be designed as, which is denoted as AM-Softmax loss function,

$$L_{\text{AM-Softmax}} = -\frac{1}{N}\sum_i \log\frac{e^{(\cos(\theta_{y_i})-m)}}{e^{(\cos(\theta_{yi})-m)} + \sum_{j=1,j\neq y_i}^{n} e^{\cos\theta_j}} = -\frac{1}{N}\sum_i \log\frac{e^{(W_{y_i}^T x_i - m)}}{e^{(W_{y_i}^T x_i - m)} + \sum_{j=1,j\neq y_i}^{n} e^{W_{y_i}^T x_i}} \tag{3}$$

3.3 The Overall Procedure of the Proposed Method

By adding the Inception branch network and the AM-Softmax loss function, the network can extract features with local and global information, and optimize the differences between different classes of ships. The whole procedure of the proposed method is described in details as below.

(1) The input image I is cropped to 448×448, and it is horizontally flipped, randomly rotated, randomly cropped, etc. The processed image is noted as X.

(2) The processed images are input to the proposed network, and the feature extraction process is denoted as $W * X$, where $*$ represents a series of convolutional, Relu and pooling operations, W_A and W_B represent all the parameters of the two branches. f_A and f_B are the extracted feature maps, and each of shape is $28 \times 28 \times 512$.

$$f_A = W_A * X \tag{4}$$

$$f_B = W_B * X \tag{5}$$

(3) At the same position l of the two feature maps, f_A and f_B have a 1×512 vectors, i.e., $f_A(l, X)$ and $f_B(l, X)$, the outer product operation yields a 512×512 matrix $b(l, X)$.

$$b(l, X) = f_A^T(l, X) f_B(l, X) \tag{6}$$

(4) Perform an average pooling operation on the matrixes b^X.

$$b^X = \text{averagepooling}(\sum b(l, X)) \tag{7}$$

(5) The bilinear vector B^X is obtained by vectorising b^X.

$$B^X = \text{vector}(b^X) \tag{8}$$

(6) The obtained bilinear vector B^p is normalized as follows.

$$B^p = sign(B^X)\sqrt{|B^X|} \tag{9}$$

(7) Then L_2 normalization of the above features is performed as follows, z is the input of the next layer.

$$z = B^p / ||B^p||_2 \tag{10}$$

(8) z is input into the fully connected layer and is predicted to s_j by using the AM-Softmax function.

$$s_j = \frac{e^{w_{y_i}^T z_i - m}}{e^{w_{y_i}^T z_i - m} + \sum_{j=1, j \neq y_i}^{n} e^{w_{y_i}^T z_i}} \tag{11}$$

(9) The loss is calculated using the AM-Softmax loss function, then the loss is back propagation for network optimization, and the network parameters are updated.

$$L_{\text{AM-Softmax}} = -\frac{1}{N} \sum \log s_j \tag{12}$$

4 Experimental Results

To validate the performance of the proposed method, we conduct experiments on the constructed dataset and three other public datasets. And comparison experiments are performed with other four popular methods to further verify the effectiveness of the proposed method. In the following parts, we will present the dataset, details of the training process, the ablation experiments, and the comparison results in details.

4.1 Dataset

The Ship-43 dataset is a fine-grained image dataset constructed by our group independently. Some of its images and labels come from the website CNSS (www.cnss.com.cn). The Ship-43 dataset contains 43 categories, each containing approximately 1,100 images. Some examples are shown in the Fig. 6. In each category, 1,000 images were used for training and the other 100 images were used for testing. In addition, to validate the generalization capacity of the proposed method, three commonly used public datasets for fine-grained image recognition are also used, including the Cub dataset [25], the Car dataset [26], and the Aircraft dataset [27]. The Cub dataset contains 11,788 images of 200 bird species, where each category contains a relatively balanced set of 30 training images and 29 test images. The Car dataset contains 16,185 images of 196 categories of cars, the cars' key features include vehicle manufacturer, car make, and model, etc. The Aircraft dataset contains 102 categories, there are 100 images in each class, of which two-thirds are used for training and the other images are used for testing.

4.2 Training Details

Experimental frameworks and devices. This paper chooses PyTorch framework for experiments. The experiments use four NVIDIA Tesla V100, each of them with a memory size of 32G.

Figure 6: Examples of ship images in Ship-43

Network training. This paper adopts the transfer learning approach for the training of the model, and the network is pre-trained on ImageNet. In the first stage, all parameters of the network except the fully connected layer are frozen, and the parameters of the fully connected layer are learned in the fine-grained dataset using a larger learning rate. In the second stage, the entire network is fine-tuned with a smaller learning rate.

Image size. Image size will affect the ultimate accuracy of the experiment. Taking into account the machine memory, the image size is set to 448×448.

Learning rate. The learning rate is set to 1e−2 in the first stage, then it is set to 1e−3 when the network is fine-tuned.

Batch Size. When defining this parameter, we consider the size of the dataset and the computer's memory, so the batch size is 256 in this paper.

4.3 Fine-grained Ship Recognition Results

To evaluate the performance of the proposed model, ablation experiments and comparison experiments are carried out on the above datasets. Firstly, ablation experiments are performed to verify the influence of the Inception branch and the AM-Softmax for fine-grained ship image recognition, respectively. Then, comparison experiments with four well-known methods are performed to validate the effectiveness of the proposed method.

4.3.1 Ablation Experiment for Inception Branch

Based on the Softmax loss function, to verify the efficiency of the Inception branch network, we design three different networks: (1) BCNN: the two branch networks use the VGG16 network. (2) BCNN-I: both networks use the Inception branch. (3) BCNN-II: one branch uses the VGG16 network, and another uses the Inception branch network. The experimental results are presented in Fig. 7.

From the experimental results, the network merging the Inception branch and the VGG branch has the highest accuracy in all datasets. On the Ship-43 dataset, our method improves by 2.06% compared to the BCNN network. It also improves by 1.14% compared to the network of the two Inception branches. It states that the network extracted simultaneously the global information and local information is more appropriate. Meanwhile, it can be seen that our proposed network also improves the accuracy on three general fine-grained image datasets.

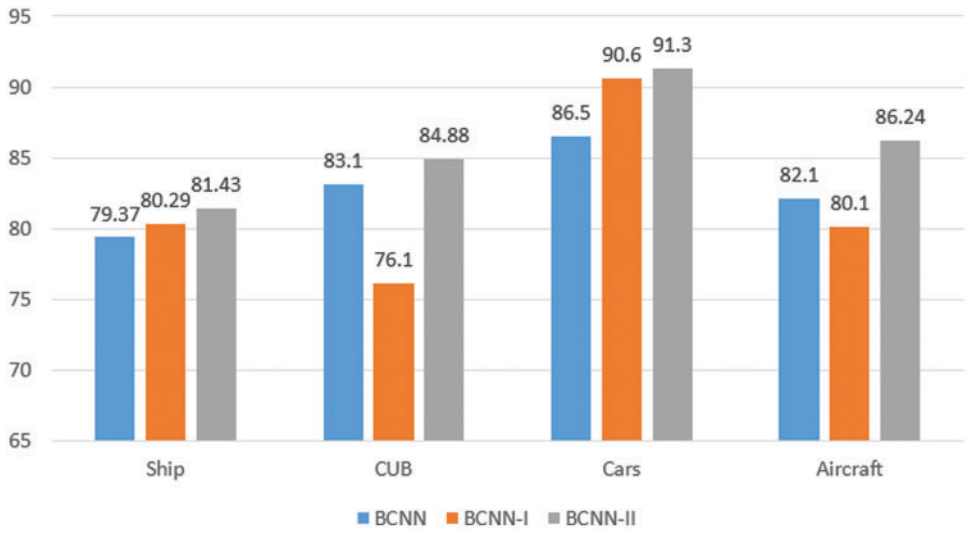


Figure 7: Accuracy of different networks on different datasets

4.3.2 Ablation Experiment for AM-Softmax

To properly assess the influence of the AM-Softmax, the benchmark network for all the experiments in this section is BCNN. The influence of AM-Softmax function on ship fine-grained recognition will be analyzed in two parts: the influence of different additive margin values m, and the comparison of accuracy between different loss functions.

A. The influence of different additive margin m values

To explore how a margin can be added manually to assist the network in achieving better accuracy, the m values are set to 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6 in this section, and the accuracy is shown in Tab. 1.

Table 1: Results of different m values

m values	Ship-43	Cub	Car	Aircraft
0.1	80.86	82.41	88.47	82.63
0.2	80.59	81.28	88.34	81.38
0.3	80.15	82.31	87.42	81.76
0.4	80.97	83.68	**91.94**	84.07
0.5	**81.76**	**83.96**	90.89	**84.11**
0.6	81.24	83.51	89.53	83.17

From Tab. 1, we can see that margin value is a hyperparameters, and different margin values result in different accuracy. Moreover, for different dataset, the best margin value is different. For example, when $m = 0.5$, three datasets, Ship-43, Cub, and Aircraft, obtain the best result compared with other margin values. Compared with the Cub and Car dataset, different values has less effect on the Ship-43 datasets, this may be because the Ship-43 dataset has a relatively small number of categories and a large number of pictures.

B. The comparison of accuracy between different loss functions

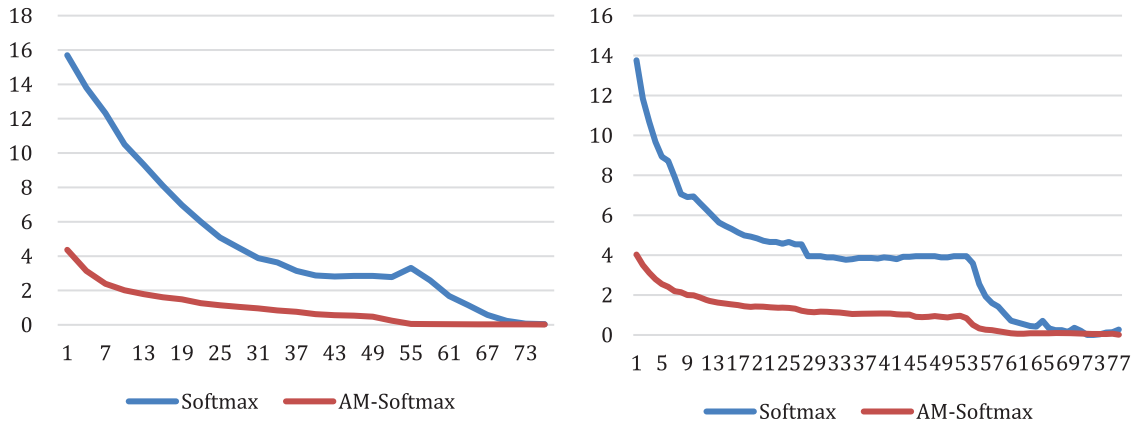
Based on the analysis of different margin values in AM-Softmax, $m = 0.5$ is selected for the following experiments. Meanwhile, the default optimal hyper-parameters are used for A-Softmax and Arc-Softmax, respectively. To demonstrate the advantages of the AM-Softmax loss function, comparison experiments are conducted with other commonly used loss functions, and the comparison results are shown in Tab. 2.

Table 2: Comparison results of different loss functions

Loss function	Ship-43	Cub	Cars	Aircraft
Softmax	79.37	83.10	86.50	82.10
A-Softmax	80.89	83.35	89.35	75.26
Arc-Softmax	81.95	79.05	**91.94**	82.21
AM-Softmax	**82.15**	**83.96**	91.47	**84.11**

On the Ship-43 dataset, compared to the Softmax function, these modified functions (A-Softmax, Arc-Softmax and AM-Softmax) improve the recognition accuracy, and especially the model with the AM-Softmax function improves the accuracy by 2.78%. Moreover, AM-Softmax achieves the highest accuracy on both the Cub and Aircraft datasets.

The Fig. 8 shows the trend of the loss value of AM-Softmax and Softmax.



(a) Loss of the training process

(b) Loss of the validation process

Figure 8: Loss graph

We can see that the loss value of the AM-Softmax is always much smaller than the loss value of the Softmax during the training process. In addition, using AM-Softmax loss function, the network can not only improve the convergence speed but also improve the accuracy. Meanwhile, during the validation process, the AM-Softmax loss value is also smaller than the Softmax loss value, which further indicates that AM-Softmax is more accurate for fine-grained recognition. Due to the two-step training method, the learning rate decay is used to fine-tune the network in the later stage, which makes the loss value of the loss function further decrease.

4.3.3 Comparison Results

To further verify the effectiveness of the proposed method. We conducted comparison experiments with four popular models for fine-grained image recognition, that are the compact bilinear pooling network (CBP) [28], the low-rank bilinear pooling network (LRBP) [29], the BCNN with Softmax function and BCNN with AM-Softmax function. The CBP obtains feature representation by designing novel convolutional kernel based on BCNN. The LRBP can compress the model through the co-decomposition of the larger classifiers. These networks are also frequently used for fine-grained recognition tasks. Because the benchmark framework of this paper is BCNN, in order to be fair, on the same framework, using the variant network of BCNN and our method to compare, it can be reflected that the improvement for different links has significantly different effects.

As shown in Tab. 3, our method achieves the highest accuracy on the Ship-43 dataset. Compared with CBP and LRBP, it exceeds 2.29% and 0.88%, respectively. Compared to the BCNN with the Softmax or the AM-Softmax, the maximum improvement is 4.08%. Our method achieves the highest accuracy on the Cub and Aircraft datasets. In addition, our method only improves on the backbone network and loss function, so our method has a similar computational cost to BCNN. Overall, the effectiveness and generalizability of the method described in this paper for fine-grained recognition is further validated.

Table 3: Recognition accuracy of different models

	Ship-43	Cub	Cars	Aircraft
BCNN + Softmax	79.34	83.10	86.94	82.10
CBP	81.13	84.01	90.83	**87.40**
LRBP	82.54	84.21	90.92	87.31
BCNN + AM-Softmax	82.15	83.96	**91.47**	84.11
Our method	**83.42**	**85.32**	90.63	86.81

5 Conclusion

In this paper, to improve the performance of fine-grained ship image recognition, we modify the BCNN network in two aspects. Firstly, by adding Inception branches to the feature extraction network, the network can merge local and global feature information from different scale kernels. Secondly, by adding margin values to the decision boundary, the AM-Softmax function can optimize the difference between ship classes, and can better activate different categories. Moreover, we construct a fine-grained ship image dataset. Ablation experiments and comparison result on the fine-grained ship dataset and three other fine-grained datasets demonstrate that our method is effective and has high generalization ability. And the proposed method can be applied to many fine-grained applications, such as bird species identification, cars identification, aircraft type identification, online plant identification. Our future work will focus on designing end-to-end models that can extract more distinguishable details to further improve the accuracy of fine-grained ship image recognition.

Acknowledgement: We express our thanks to Professor Li Yujian for providing devices.

Funding Statement: This work is supported by the National Natural Science Foundation of China (61806013, 61876010, 62176009, and 61906005), General project of Science and Technology Plan

of Beijing Municipal Education Commission (KM202110005028), Beijing Municipal Education Commission Project (KZ201910005008), Project of Interdisciplinary Research Institute of Beijing University of Technology (2021020101) and International Research Cooperation Seed Fund of Beijing University of Technology (2021A01).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. S. Wei, J. Wu and Q. Cui, "Deep learning for fine-grained image analysis: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 23, pp. 118–173, 2021.
[2] T. Zhang, X. Zhang and J. Shi, "Depthwise separable convolution neural network for high-speed SAR ship detection," *Remote Sensing*, vol. 21, no. 21, pp. 2483–2492, 2019.
[3] T. Vaiyapuri, S. N. Mohanty, M. Sivaram, I. V. Pustokhina, D. A. Pustokhin *et al.,* "Automatic vehicle license plate recognition using optimal deep learning model," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1881–1897, 2021.
[4] Y. Xia and S. Wan, "A novel sea-land segmentation algorithm based on local binary patterns for ship detection," *Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 3, pp. 237–246, 2014.
[5] S. Y. Fan and F. Luo, "Fractal properties of autoregressive spectrum and its application on weak target detection in sea clutter background," *IET Radar, Sonar & Navigation*, vol. 9, no. 8, pp. 1070–1077, 2015.
[6] Y. LeCun, Y. Bengion and G. Hinton, "Deep learning," *Nature*, vol. 5, no. 21, pp. 436–444, 2015.
[7] T. Y. Lin and S. Maji. "Bilinear cnn models for fine-grained visual recognition," in *Proc. ICCV*, New York, NY, USA, pp. 1449–1457, 2015.
[8] O. Russakovsky, J. Deng and H. Su, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2019.
[9] C. Szegedy, "Going deeper with convolutions," in *Proc. CVPR*, New York, NY, USA, pp. 1–9, 2015.
[10] F. Wang, J. Cheng and W. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
[11] S. Saqib, A. Ditta, M. A. Khan, S. Asad and H. Alquhayz, "Intelligent dynamic gesture recognition using cnn empowered by edit distance," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 2061–2076, 2021.
[12] W. Tan, Y. Wu, P. Wu and B. Chen, "A survey on digital image copy-move forgery localization using passive techniques," *Journal of New Media*, vol. 1, no. 1, pp. 11–25, 2019.
[13] N. Zhang and J. Donahue, "Part-based r-cnns for fine-grained category detection," in *Proc. ECCV*, Springer, Switzerland, pp. 834–849, 2014.
[14] X. S. Wei and C. W. Xie, "Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization," *Pattern Recognition*, vol. 23, no. 4, pp. 704–714, 2018.
[15] Y. Peng, X. He and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1487–1500, 2017.
[16] H. Zheng and J. Fu, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proc. CVPR*, New York, NY, USA, pp. 5012–5021, 2019.
[17] C. Zhu, Y. K. Wang, D. B. Pu, M. Qi, H. Sun *et al.,* "Multi-modality video representation for action recognition," *Journal on Big Data*, vol. 2, no. 3, pp. 95–104, 2020.
[18] C. Yuan, S. Jiao and X. Sun, "Mfffld: A multi-modal feature fusion based fingerprint liveness detection," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 1, no. 1, pp. 1–14, 2021.
[19] C. Szegedy, V. Vanhoucke and S. Ioffe, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, New York, NY, USA, pp. 2818–2826, 2016.
[20] X. R. Zhang, X. Chen, W. Sun, X. Z. He, "Vehicle Re-identification model based on optimized densenet121 with joint loss," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3933–3948, 2021.

[21] W. Liu and Y. Wen, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, New York, NY, USA, pp. 1–7, 2016.
[22] W. Liu, Y. Wen and Z. Yu, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. CVPR*, New York, NY, USA, pp. 212–220, 2017.
[23] H. Wang, Y. Wang and Z. Zhou. "Cosface: Large margin cosine loss for deep face recognition," in *Proc. CVPR*, New York, NY, USA, pp. 5265–5274, 2018.
[24] J. Deng and J. Guo, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, New York, NY, USA, pp. 4690–4699, 2019.
[25] P. Welinder and S. Branson, "Caltech-ucsd birds 200," *California Institute of Technology*, vol. 12, no. 3, pp. 1487–1500, 2010.
[26] J. Krause and M. STARK. "3D object representations for fine-grained categorization," in *Proc. ICCV*, New York, NY, USA, pp. 554–561, 2013.
[27] W. Sun, G. C. Zhang and X. R. Zhang, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy," *Multimedia Tools and Applications*, vol. 80, pp. 30803–30816, 2021.
[28] Y. Gao and O. Beijbom, "Compact bilinear pooling," in *Proc. CVPR*, New York, NY, USA, pp. 317–326, 2016.
[29] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *Proc. CVPR*, New York, NY, USA, pp. 365–374, 2017.