Research on Fine-Grain Model Recognition Based on Branch Feedback Convolution Neural Network

Shijin Li, Ting Shi, Fushou Tao and Rong Jiang

March 19, 2019

Research on Fine-Grain Model Recognition Based on Branch Feedback Convolution Neural Network

Shijin LI
Yunnan University of Finance and Economics
Kunming, china
Sothink1984@126.com

Ting SHI*
Yunnan School of Business, Information and Engineering
Kunming, china
**Corresponding author*
shiting1988@126.com

Fushou Tao
Yunnan University of Finance and Economics
Kunming, china
shuilifang1985@qq.com

Rong Jiang
Yunnan University of Finance and Economics
Kunming, china
765256273@qq.com

***Abstract*—Fine-grained vehicle identification has a wide range of applications in many fields, and the requirements for recognition accuracy are high in various application scenarios. In this paper, a branch fusion convolutional neural network algorithm for fine-grained car recognition is designed. The VGG16 convolutional neural network is merged with AlexNet to form a bifurcated fusion convolutional neural network. On this basis, the multi-branch training idea of GoogleNet is cited to make the network model stabilize and converge during training. The network was trained and tested on the CompCars fine-grained car dataset. The correct rate of the test set Top-1 reached 91.29%, and the model was accurate and effective.**

Keywords—fine-grained vehicle identification; Feedback convolutional neural network; multi-branch training; VGG16;

I. INTRODUCTION

Fine-grained vehicle identification based on computer vision plays a key and broad role in the field of intelligent transportation. It plays an important role in tracking illegal vehicles, traffic flow statistics, and unmanned toll stations. License plates and logos are an important part of the vehicle model that can easily be tampered with or stained and cannot be used as a unique feature of vehicle identification. The model, including many details of the information, is more difficult to disguise. Therefore, the fine-grained identification of the vehicle model provides important information for vehicle identification. Vehicle models are widely used to track illegal vehicles and provide important basis for traffic management judgment and enforcement. Vehicle model identification has important research value and application prospects.

However, car fine-grained identification has great challenges. First of all, there are thousands or even tens of thousands of known types of cars in China and in various countries. The more types of vehicles, the harder it is to achieve precise fine-grained classification. Secondly, many vehicles from the same brand look very similar, and it is almost impossible to accurately classify them only by human eyes or traditional machine vision recognition methods. Finally, whether it is a car image under surveillance or a car image taken by a camera in a natural scene, most of them are affected by natural environments such as smog, snow, and light. These have greatly increased the difficulty of accurate identification of the model.

Deep convolutional neural networks are widely used in computer vision and their performance is also excellent. The biggest advantage is that the local receptive domain is shared with weights. The local receptive field transforms the full connection into a local

connection through a convolution operation because the multi-layer network is capable of extracting high-order statistical properties. The weight sharing shares a convolution kernel in different images or the same image, reducing the number of repeated convolution kernels, which can greatly reduce the number of convolution kernels and it speed up the operation [1].

However, the convolutional neural network depends on the coverage of the training data and the distance between each class. When a single shallow convolutional neural network is used for vehicle type identification, the features extracted by the single convolution layer sometimes cannot effectively identify the fine-grained models. Therefore, the recognition effect is not good; while the deep convolutional neural network can extract more fine-grained features, it is easy to have gradient dispersion during training. However, the deep learning method has a very good effect in image recognition classification. Therefore, designing a suitable convolutional neural network model is of great significance for fine-grained vehicle identification.

In this paper, the VGG16 convolutional neural network is improved to form a two-branch feedback convolutional neural network AV-CNN, and the idea of multi-branch training of GoogleNet is introduced, which effectively solves the gradient dispersion when the network is too deep and the oscillation does not converge during training and problem.

The content of the article is arranged as follows: Section 1 introduces background knowledge of existing methods and methods used; Section 2 details the core methods for achieving fine-grained vehicle identification; Section 3 analyzes data sets and experimental results; Section 4 summarizes look forward to summarizing the existing results, and we will apply this algorithm in other areas in the future.

II. Related Work

Krause et al. proposed a method based on 3D representation to process car images with unconstrained poses and multiple viewpoints, Gu et al [2];. proposed a training geometry classifier 3D car representation method, which is superior to its corresponding classifier. Fine-classified 2D method [3], In [4], the metric learning process of the intra-class variance ternary network is used to improve the classification effect; the author of [5] combines global and local features to propose a coarse to fine CNN model; [6] proposes a Spatial Weighted Summary (SWP) method, which effectively improves the classification ability of DCNN; in [7], two different models are established on the deformable component model (DPM) for partial component identification. Improve the recognition effect; [8] use the strongly supervised DPM method to introduce the semantic part hierarchy into the location part of the subdivision.

As the core of deep learning, convolutional neural networks also have very good performance in the field of vehicle identification. [9] directly applied several DCNN baselines to the car model classification and achieved very good recognition results with theart-ofart; Liu et al. used a common segmentation algorithm in combination with R-CNN [11] to effectively generate component annotations. [10]; Lin et al. use subarray convolutional layer activation as local features and from two successive convolutional layer convolutional feature maps to extract local features [12].

III. VGGNet16 Algorithm Experimental Test

VGGNet is a deep convolutional neural network developed by Oxford University's computer vision portfolio, Google DeepMit. It explored the relationship between the depth of the convolutional neural network and the recognition performance. By repeatedly stacking 3*3 small convolution kernels and 2*2 maximum pooling layers, VGGNet successfully constructed 16-19 deep convolutional neural networks. . In the 2014 ILSVRC competition classification project, he achieved the second place and the first place in the positioning project. At the same time, VGGNet is very scalable, and the generalization of migration to other image data is very good. Whether in academia or industry, it is widely studied and applied. VGGNet16 is a branch of VGG with excellent performance and best generalization. Its structure is shown in Figure 1.

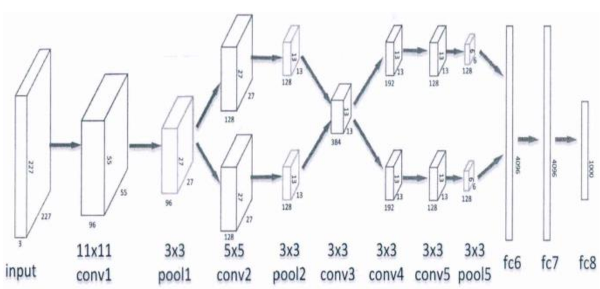


Fig. 1. VGGNet 16 structure diagram

The content of the article is arranged as follows: Section 1 introduces background knowledge of existing methods and methods used; Section 2 details the core methods for achieving fine-grained vehicle identification; Section 3 analyzes data sets and experimental results; Section 4 summarizes look forward to summarizing the existing results, and we will apply this algorithm in other areas in the future.

When VGGNet16 operates on the convolutional layer, the formula is:

$$y_n^p = f(\sum_{i \in T_n} y_m^{p-1} * k_{mn}^p + b_n^p) \qquad (1)$$

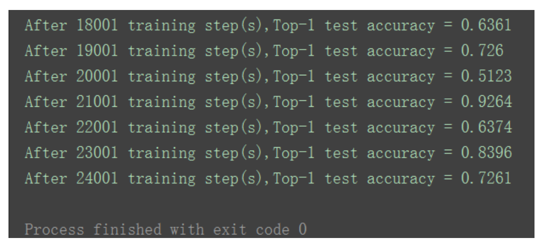
Where y_n^p is the nth channel output of the convolutional layer p; $f()$ is the activation function, the activation function used in AlexNet is the ReLU function; T_n is used to calculate the total input feature map subset; k_{mn}^p represents the convolution kernel; "*" indicates a convolution operation.

When performing a pooling operation, the formula is calculated as [6]:

$$y_n^p = f(\omega_n^p \, pool\left(y_n^{p-1}\right) + b_n^p) \qquad (2)$$

Where, ω_n^p represents the pooling layer weight parameter, and $pool()$ represents the pooling function. By sliding the window method, the maximum pooling operation is performed on the pixel, that is, the maximum value in each pixel block is obtained.

Change the VGGNet 16 network output layer to 280, and the rest will not change. Use CompCars to train the network and test 200 samples per batch. The resulting Top-1 test results are shown in Figure 2.

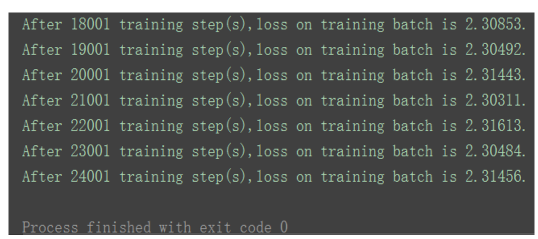


After 18001 training step(s),Top-1 test accuracy = 0.6361
After 19001 training step(s),Top-1 test accuracy = 0.726
After 20001 training step(s),Top-1 test accuracy = 0.5123
After 21001 training step(s),Top-1 test accuracy = 0.9264
After 22001 training step(s),Top-1 test accuracy = 0.6374
After 23001 training step(s),Top-1 test accuracy = 0.8396
After 24001 training step(s),Top-1 test accuracy = 0.7261

Process finished with exit code 0

Fig. 2. VGGNet16 Top-1 test accuracy rate

The loss rate change diagram during training is shown in Figure 3:



After 18001 training step(s),loss on training batch is 2.30853.
After 19001 training step(s),loss on training batch is 2.30492.
After 20001 training step(s),loss on training batch is 2.31443.
After 21001 training step(s),loss on training batch is 2.30311.
After 22001 training step(s),loss on training batch is 2.31613.
After 23001 training step(s),loss on training batch is 2.30484.
After 24001 training step(s),loss on training batch is 2.31456.

Process finished with exit code 0

Fig. 3. VGGNet16 training loss rate

From the experimental results, the test accuracy rate training to 24000 steps is always unstable, and the oscillation range is large. Observing the change of the training loss rate, the loss rate is always above 2.3, although it is stable, it has not fallen below the effective loss. According to the depth of VGGNet16 network, the network has a gradient dispersion phenomenon. At the same time, due to the small amount of data in the CompCars data set, the gradient dispersion is aggravated, so that the training cannot be performed stably, and the loss rate cannot be reduced naturally.

IV. AV-CNN BRANCH FEEDBACK CONVOLUTIONAL NEURAL NETWORK

A. *AV-CNN convolutional neural network without separate training*

It can be seen from the above experiment that VGGNet has a gradient dispersion phenomenon due to the depth of the network, and the training cannot be performed. Combined with the advantages of the VGGNet16 network, at the same time, the single-branch target field of VGGNet16 is limited, and the AV-CNN bifurcated fusion convolutional neural network without separate training is formed. The structure is shown in Figure 4.

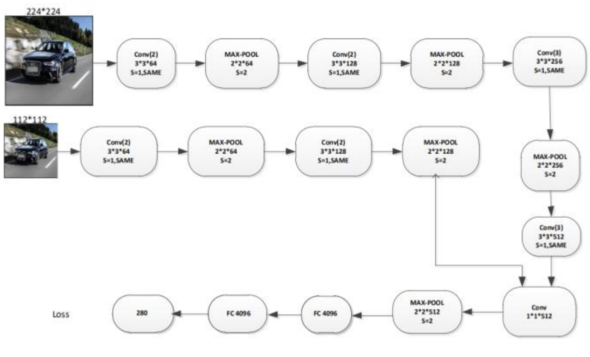
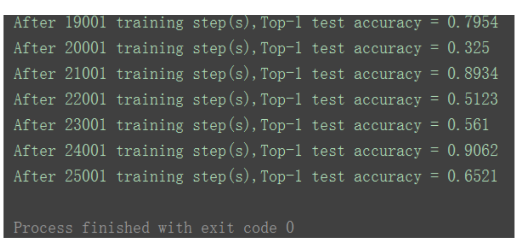


Fig. 4. Double-branch feedback convolutional Neural Network

The convolutional neural network shown in Fig. 4 is divided into two branches for input, one with an input image size of 224*224 and the other with an input image size of 112*112. The network obviously improves the shortcomings of VGGNet16's sensory domain and low feature extraction efficiency. When training an image, it is divided into two images of different sizes for training, which greatly increases the coverage of training data.

CompCars conducted training tests on the network and tested 200 pictures per batch. The resulting Top-1 test results are shown in Figure 5.



```
After 19001 training step(s),Top-1 test accuracy = 0.7954
After 20001 training step(s),Top-1 test accuracy = 0.325
After 21001 training step(s),Top-1 test accuracy = 0.8934
After 22001 training step(s),Top-1 test accuracy = 0.5123
After 23001 training step(s),Top-1 test accuracy = 0.561
After 24001 training step(s),Top-1 test accuracy = 0.9062
After 25001 training step(s),Top-1 test accuracy = 0.6521

Process finished with exit code 0
```

Fig. 5. Double branch feedback neural network test accuracy rate

It can be seen from Fig. 5 that the test results of the network on the CompCars data set are still very low and the correct rate fluctuates significantly, even worse than the original VGGNet16 convolutional neural network results. To this end, the weight changes in the process of training the network are visualized, as shown in FIG. 6.

Fig. 6. Double-branch feedback neural network training weight changing

It can be seen from the weight change graph that when the network is training, the weight is always oscillating and does not converge, so stable model parameters are never obtained. There are two reasons for this:

1. The network has two branches. In the forward propagation, the cost functions of the two branches calculate the corresponding losses respectively, and in the case of backpropagation, the chained derivation mechanism of the training will make the valence function respectively for each The branch is derivation. When one branch is stable, the other branch is still conducting backpropagation, which affects the main path parameters. The change of the main path parameter will adversely affect the stabilized branch, causing it to oscillate again. . Therefore, there is a case where the weight fluctuation cannot converge.

2. The VGGNet network is too deep, and its total parameter quantity is as high as 80 million. Even if two branches are added, the coverage of training data is increased, but the amount of data is still relatively insufficient. Therefore, during the training process, gradient dispersion is likely to occur.

B. *Introducing a separately trained AV-CNN convolutional neural network*

In response to the above problems, combined with GoogleNet [13] to solve the deep network depth of the phenomenon of gradient dispersion phenomenon, and analyze, when each introduction of separate training, will reduce the weight fluctuation instability problem. To this end, an AV-CNN convolutional neural network is formed, as shown in FIG. After experimental verification, the network is stable and converged. The experimental results are shown in Section 4.

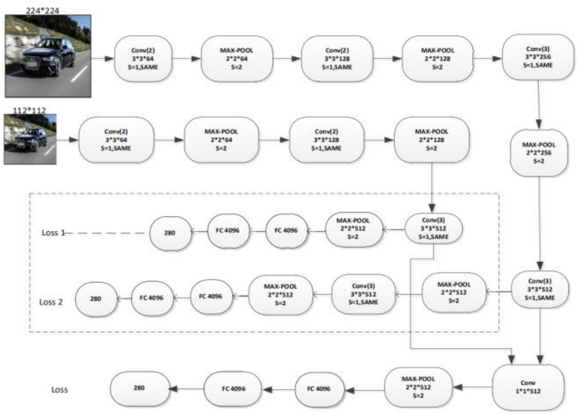


Fig. 7. shows the AV-CNN convolutional neural network after training alone

V. DATA SET PREPROCESSING AND EXPERIMENTAL RESULTS

A. Data Set Preprocessing

All the experiments used above have adopted the de-averaging operation on the data set, which is to subtract the feature mean of all the training set pictures from the features of each picture to be trained, which can effectively reduce the calculation amount and take the data from the original standard. The matrix consisting of vectors in the coordinate system becomes the coordinate system established with the vector mean as the origin. The data set image after the mean operation is shown in Figure 8.

Fig. 8. before and after image de-average comparison

B. Introduction of AV-CNN experimental results after individual training

The training of the AV-CNN after training was introduced by using the CompCars data set. The loss rate curve during the training is shown in Fig. 9.

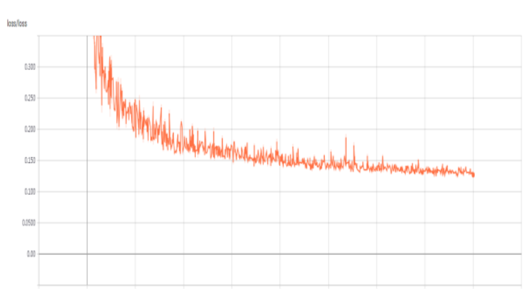


Fig. 9. shows the AV-CNN loss rate curve after training alone

During the training process, the weight change curve is shown in Figure 10.

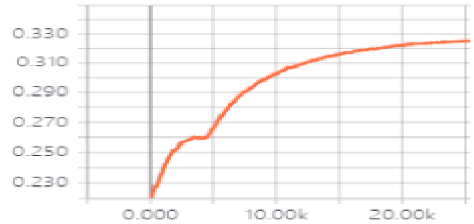


Fig. 10. Training process weight change curve

The results of the Top-1 test accuracy rate are shown in Figure 11.

```
After 19001 training step(s),Top-1 test accuracy = 0.7682
After 20001 training step(s),Top-1 test accuracy = 0.8163
After 21001 training step(s),Top-1 test accuracy = 0.8745
After 22001 training step(s),Top-1 test accuracy = 0.90961
After 23001 training step(s),Top-1 test accuracy = 0.9117
After 24001 training step(s),Top-1 test accuracy = 0.9124
After 25001 training step(s),Top-1 test accuracy = 0.9127

Process finished with exit code 0
```

Fig. 11. Top-1 test accuracy rate results

The change in weight and the change in loss rate during training indicate that the network is stable and the model parameters tend to be stable. The test accuracy rate shows that the accuracy of the test on the CompCars data set is better, and the accuracy of Top-1 reaches 91.27%.

The comparison of the results on the CompCars dataset with the accuracy of other algorithms Top-1 is shown in Table 1. The results of AlexNet, GoogleNet, and ResNet152 are all taken from the original text, and the results of VGGNet16 are from this experiment.

TABLE I. TOP-1 ACCURACY COMPARISON

Model	Top-1 Accuracy
AlexNet[14]	0.848
GoogleNet[15]	0.767
VGGNet16	0.7261
ResNet152[16]	0.907
AV-CNN	0.9127

VI. SUMMARY AND OUTLOOK

This paper introduces a fine-grained vehicle identification method. First, the VGGNet16 is tested on the CompCars dataset to find its shortcomings. A double-branch convolution feedback neural network AV-CNN is established, and its test accuracy rate is 91.27%, which has great significance for the fine-grained identification of automobiles. In addition, this model can be applied in many other fields, such as license plate recognition, text recognition and so on. The next step will be to experiment and model changes in these areas.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No.61763048, No.61263022, No.61303234, Yunnan Provincial Applied Basic Research Program under Grant No.2017FB095, the 18th batch of Yunnan Young and Middle-aged Academic and Technical Leaders Reserve Talent Development Program under Grant No.2015HB038 and the information construction of higher education management under Internet plus model under Grant No.41611217214.

REFERENCES

[1] Wenwen Zeng, Qinglin Meng et al. Handwritten digit recognition based on deep convolutional self-encoding neural network[J]. Journal of Computer Applications, 2019, v.16;No.154(03):35-38.

[2] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3D object representationsfor fine-grained categorization,” in Proc. IEEE Int. Conf. Comput. Vis.Workshops, Dec. 2013, pp. 554–561.

[3] H.-Z. Gu and S.-Y. Lee, “Car model recognition by utilizing symmetric property to overcome severe pose variation,” Mach. Vis. Appl., vol. 24,no. 2, pp. 255–274, 2013.

[4] Y. Bai, F. Gao, et al. "Incorporating Intra-Class Variance to Fine-Grained Visual Recognition." In Multimedia and Expo (ICME), 2017 IEEE International Conference on, pp. 1452-1457. IEEE, 2017.

[5] J. Fang, Y. Zhou, Y. Yu, and S. D. Du, "Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture", IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 7, pp. 1782-1792, 2017.

[6] Q. C. Hu, H. B. Wang, T. Li, and C. H. Shen, "Deep CNNs With Spatially Weighted Pooling for Fine-Grained Car Recognition", IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 11, pp.3147-3156, 2017.

[7] M. Stark et al., "Fine-Grained Categorization for 3D Scene Understanding", Proceedings of the British Machine Vision Conference 2012, 2012.

[8] L. Liao, R. M. Hu, J. Xiao, Q. Wang, J. Xiao, and J. Chen, "Exploiting Effects of Parts in Fine-Grained Categorization of Vehicles", 2015 IEEE International Conference on Image Processing (Icip), pp. 745-749,2015.

[9] L. Yang, P. Luo, C. C. Loy, and X. Tang, “A large-scale car dataset for fine-grained categorization and verification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 3973–3981.

[10] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, “Fine-grained recognition without part annotations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 5546–5555.

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

[12] L. Liu, C. Shen, and A. van den Hengel, “The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 4749–4757.

[13] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[14] SOCHOR J, HEROUT A, RAVEL J. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 3006-3015.