Weather Image Recognition Considering Light Condition Via SENet for Intelligent Traffic System

Yizhou Ding[1,a], Rui Ren[1,b] and Chengxi Luo[1,c]

[1]*School of information.Wuhan University of Technology, Wuhan, China*

a. Jasonmils@whut.edu.cn, b. 275958@whut.edu.cn, c. luochengxi@whut.edu.cn

Keywords: Extreme weather recognition, light condition evaluation, SENet, classification, intelligent traffic system introduction.

Abstract: The comprehensive automatic classification of weather category and environmental light condition has important applications in the field of intelligent Traffic. This paper proposes a weather image classification model based on SENet, which can identify 5 types of weather: sunny, cloudy, rain, fog, and snow, and can evaluate the environmental brightness (Bright, Dark) of each type of weather above. This paper uses a weather image dataset consisting of 8890 weather images, of which 7900 are the training set and 990 are the validation set. A variety of neural networks (AlexNet, GoogLeNet, and BPNN) were used to train and compare the datasets. After comparison, it was found that the accuracy of SENet was as high as 98.48% when the epoch was 200. The accuracy of extreme weather recognition is higher than the other three networks, which is more suitable for extreme weather recognition.

1. Introduction

According to Annika K. Jägerbrand's research on the influence of weather type and brightness on traffic (vehicle speed)[1], there is no significant difference in vehicle speed in daylight, twilight and darkness when the effects of light conditions on vehicle speed are analyzed separately. However, if the weather type is combined with the light conditions, the extreme weather such as rain or snow will cause the speed of vehicle to decrease, and the speed on the road with dark condition is higher than the speed of the road with light. These results indicate that the driver failed to adjust the vehicle speed in the dark to the visibility corresponding to the current light intensity in time, explaining the strong relationship between traffic accidents and dark or weak light conditions [1]. It can be seen that a model that can automatically classify and evaluate weather types and light condition is very important. When different weather types and light condition are judged, different traffic measures can be taken in time to reduce the probability of traffic accidents.

At present, weather recognition still mainly relies on hardware sensors with high energy consumption[2]. However, the installation and maintenance of the sensor consumes a lot of manpower and material resources, and the recognition accuracy of the sensor is greatly affected by the environment, and it is difficult to achieve real-time transmission[2-4]. In recent years, with the continuous development of Intelligent Traffic System, various monitoring devices have been installed on roads, so machine vision-based weather recognition has the characteristics of low resource consumption and higher reliability[2,4,5], which has become the hotspots in the field of intelligent transportation research.

In paper[6], Kang proposed a four-class weather recognition framework using AlexNet and GoogLeNet, which can identify four weather types: fog, rain, snow, and clear weather, and verified it in an open-source weather atlas. Based on GoogLeNet, 92.0% recognition accuracy was obtained, which is higher than 91.1% achieved by AlexNet. In paper[3], Zhu used GoogLeNet to identify extreme weather. It can recognize fog, storm, sunny and snowy days. After training 40,000 times, with fine-tuning, GoogLeNet can reach 94.5 % Training accuracy. However, because the types of weather recognized are limited, and only the types of weather are realized, the application scenarios are limited, and the traffic environment cannot be fully reflected, which affects the efficiency of traffic management. In paper[7-9], a multi-label weather recognition framework was proposed, that is, a weather image has multiple weather tags, which increases the robustness of the system and achieves high recognition accuracy. In paper[10], Dannheim used sensors such as LIDAR and cameras to identify weather types. The system can identify six types of weather: rain, snow, fog, dust, light rain and water vapor. It has high sensitivity and spatial resolution and is suitable for automatic car ABS systems. However, because LIDAR is a hardware sensor, and most of its sensors are concentrated on the outside of the car according to the paper, and only the camera is placed in the car, which is easily affected by the environment. Once the sensor fails, the weather recognition accuracy will drop sharply.

In summary, the existing weather identification methods have the following problems:

1) At present, most weather recognition methods can only identify about 4 types of weather types, and there are fewer recognition types. Only the weather types are considered, and the combination of environmental light conditions is not considered.

2) The installation and maintenance of the sensor will consume a lot of manpower and material resources, and its accuracy may be affected by the environment, and once damaged, the accuracy of weather recognition will drop sharply;

In order to solve the problems above, this paper proposes an image classification model that can identify 5 types of weather: sunny, overcast, rain, fog, and snow, and can evaluate the environmental light condition (Bright, Dark) of various types of weather. In recent years, convolutional neural networks (CNNs) are the state-of-the-art in challenging recognition tasks such as target detection, face recognition, and retrieval. CNNs can automatically extract effective features from big data without manual selection[3]. Therefore, this paper studies the model of applying SENet to weather image recognition, and compares and analyzes it with three other models (AlexNet[11], GoogLeNet[12,13], and BPNN[14,15]). The experiment found that three types of deep convolutional network models (AlexNet, GoogLeNet, SENet) and the shallow network model BPNN all have a certain effect on classifying 5 types of weather: sunny, overcast, rain, fog, and snow, and judge the environmental light condition (Bright, Dark). However, because SENet has better spatial resolution, the characteristics of important channels are strengthened, and the characteristics of non-important channels are weakened. Therefore, when identifying extreme weather, higher recognition accuracy can be obtained by SENet, which is suitable for distinguishing extreme weather types. The code and weather image datasets implemented in this paper is available on the GitHub link below: https://github.com/Jasonmils/Weather-Image-Recognition-Considering-Light-Condition-Via-CNNs

2. Applied CNN Framework

The layout of a convolutional neural network (CNN) is close to the biological neural network. Weight sharing reduces the complexity of the network, especially in image classification, and can achieve its best performance. This paper uses CNNs for weather image classification. It extracts local features through continuous convolution, then samples from the pooling layer, converts high-dimensional features into low-dimensional representative features, and finally obtains feature vectors through fully connected layers. This feature vector can be the input of classification or retrieval operation. At present, there are many CNN structures, but different CNNs have different classification effects on image sets. SENet[16] proposed by Momenta Hu Jie's team has the following advantages:

1) The SE Block is universal and can match all CNNs and embed them in other CNNs;

2) Ordinary convolutional networks usually aggregate information on local areas and feature dimensions on feature maps, and use a series of convolution operations to make the network obtain the global information, and SENet can distinguish the importance of channels to improve the precision of important categories through the association between channels.

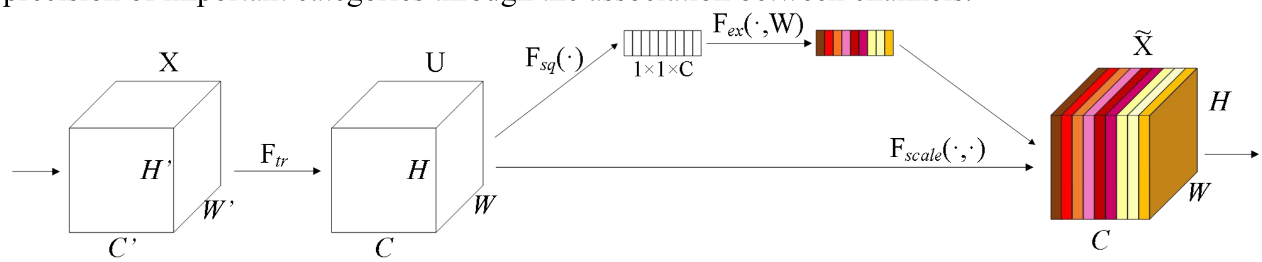

Figure 1: SE Block diagram.

This paper uses SENet embedded Inception v3 as the main framework. For CNN, its core calculation is a convolution operator, which learns a new feature map from the input feature map through a convolution kernel. In essence, convolution is a feature fusion of a local area, which includes spatial (H and W dimensions) and inter-channel feature fusion. A model of the dependency relationship between feature channels is obtained through squeeze and excitation operations. The diagram of SENet is shown in Figure 1.

F_{tr} is a standard convolution operator: $F_{tr}: X \to U$, $X \in R^{H'\times W'\times C'}, U \in R^{H\times W\times C}, V = [v_1, v_2, v_3, \cdots\cdots v_c]$, where v_c represents the c-th convolution kernel, then the expression that outputs $U = [u_1, u_2, u_3, \cdots\cdots u_c]$:

$$u_c = v_c * X = \sum_{S=1}^{C_0} V_C^S * x^s \qquad (1)$$

Where $*$ denotes convolution, and v_C^S a 2-D spatial kernel representing a single channel of v_c that acts on the corresponding channel of X. It inputs the spatial features on a channel, but because the convolution results of each channel are summed, the channel feature relationship is mixed with the spatial relationship learned by the convolution kernel. The SE Block is to extract this kind of confusion, so that the model directly learns the channel feature relationship.

Since convolution only operates in a local space, it is difficult to obtain enough information to extract the relationship between channels. Therefore, in order to solve the problem of channel dependency utilization, the global spatial information needs to be squeezed into the channel descriptor and the global average pooling is used. Global average pooling generates channel statistics[16]. The channel statistic z is generated by shrinking U in the spatial dimension H × W, where the c-th element of z is shown in (2).

$$z_c = F_{sq}(u_c) = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} u_c(i,j) \tag{2}$$

The squeeze operation gets the global description characteristics. In order to make full use of the data obtained in the squeeze operation, the channel dependency is fully captured by excitation. This operation needs to meet two criteria:

1)it must be flexible and able to learn the non-linear relationship between various channels;

2) learning relationship is not mutually exclusive. Hence, here we apply the gating mechanism in the form of sigmoid:

$$s = F_{ex}(z,W) = \sigma(g(z,W)) = \sigma(W_2\delta(W_1 z)) \tag{3}$$

Where $W_1 \in R^{\frac{C}{r}\times C}, W_2 \in R^{C\times\frac{C}{r}}$, δ refers to the ReLU function. Two fully-connected layers are added on the basis of non-linearity so as to reduce the complexity of the model and improve the generalization ability. W_1 and W_2 are the parameters of the two fully-connected layers. Using the fully-connected layer can fuse the information of each channel[17]. We use a bottleneck structure containing two fully-connected layers. The first layer plays the role of dimensionality reduction and is activated by the ReLU function. The last layer restores the original dimensions. The final output of the SE block $X = [x_1, x_2, x_3, \cdots\cdots x_c]$ is obtained by adjusting the weight of U, namely:

$$x_c = F_{scale}(u_c, s_c) = s_c u_c \tag{4}$$

The flexibility of the SE block means that it can be applied to conversions other than standard convolutions[12]. Since the application scenario of this paper is intelligent traffic based on weather recognition, which requires the network to have high real-time and accuracy, Inception v3 was selected as the embedded framework of SENet. Inception v3 [18] is an improvement over Inception v2. The most important improvement is the ability to solve the n × n convolution integral into two one-dimensional convolutions (1 × n and n × 1), which can speed up the calculation, and make the network more real-time in weather recognition, can increase the nonlinearity of the framework, and improve the accuracy of the network. So, this paper embeds the SE block into the Inception v3 block to form the SE-Inception v3 block. The schematic diagram is shown in Figure 2.

SE-Inception block is divided into 9 layers. The first layer is the Input layer. The second layer applies an Inception block and is pre-trained on ImageNet. The third layer is the Squeeze layer, which globally average pools the output of the Inception block. The fourth to sixth layers are Excitation layers. The fourth and fifth layers form a bottleneck structure that includes two fully connected layers. The first fully connected layer reduces the dimension 2048 output by the Squeeze layer, and the reduction ratio is 16[12]. The second fully connected layer restores the original dimensions. The purpose of using the bottleneck structure is to increase the nonlinearity of the network, which can better fit the complex correlation between channels. The sixth layer rescales the input size to 1 × 1 × 2048. The seventh layer multiplies the output of the Inception block and the output of the Excitation layer to adjust weight. The eighth layer is global average pooling again.

The last layer is a fully connected layer, which is identified using the SoftMax function for classification.

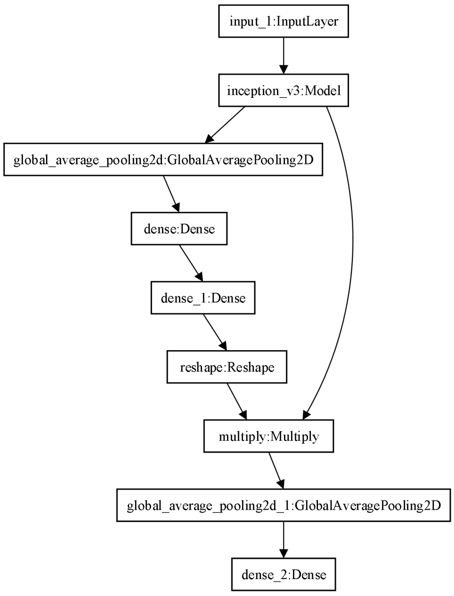

Figure 2: SE-Inception v3 block.

3. The Constructed Dataset

This paper constructs a weather image dataset containing 8890 images, of which 7,900 are in the training set and 990 in the validation set. Part of the image data of sunny, cloudy, foggy and snowy days is from paper[8-9], and the image data of rainy days is mainly from paper[19]. Due to the poor quality of some images in the original dataset constructed in other paper, it is not suitable for weather recognition. Therefore, the collected images dataset contains only 3025 images. Hence, the dataset collected by myself was used to supplement the original dataset to 8890 images, and some samples were expanded using image processing technology to balance the number of various pictures. Finally, considering the influence of light condition, this paper divides the weather image dataset into 10 categories: Cloudy_bright, Cloudy_dark, Foggy_bright, Foggy_dark, Rainy_bright, Rainy_dark, Snowy_bright, Snowy_dark, Sunny_bright, Sunny_dark. The image dataset contains all kinds of complex scenes, such as urban traffic pavement, buildings, traffic, ports, etc., so that the dataset has a certain randomness. The system has a strong ability to recognize test images in different situations. Since weather images are collected from multiple angles, identifying weather images in WeatherDataset is extremely challenging. Some image samples are shown in Figure 3 (a)~ (j).

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3: Weather image dataset(a)Cloudy_bright (b)Cloudy_dark (c)Foggy_bright (d)Foggy_dark (e)Rainy_bright (f)Rainy_dark (g)Snowy_bright (h)Snowy_dark (i)Sunny_bright (j)Sunny_dark.

All the images in the dataset are resized and augmented before entering training process to improve robustness. The image adjustment uses the cv2.resize function to adjust the images in the dataset to the size of 75 * 75. Then transform the picture to get a stronger generalization ability, which can be better applied to the application scenario (weather recognition). The data augment method adopted in this paper is image flip, which uses cv2.flip function to achieve the corresponding flip processing of three flipMode (1, 0, -1) image processing, namely horizontal flip, vertical flip, horizontal vertical flip. After processing, the number of weather image dataset increased to 4 times. The schematic diagram of image preprocessing is shown in Figure 4.

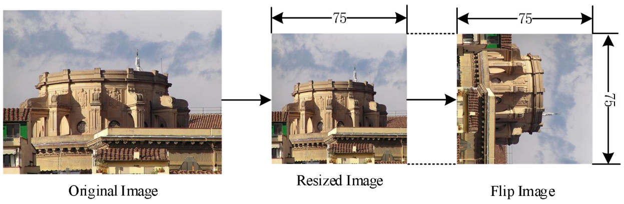

Figure 4: Image collection preprocessing.

4. Experiment and Result

In order to achieve the requirements of this paper, we used Anaconda to call TensorFlow-Keras to build the SENet-Inception v3 framework, and used two deep convolutional neural networks, AlexNet and GoogLeNet&Inception v3, and shallow convolutional neural network BPNN to compare and evaluate performance of the SENet-Inception v3 framework. As is shown in Fig.5, after the resize and data augment, the images in WeatherDataset are entered into four kinds of frames for training, and finally the same validation set is used to test the four trained frameworks.

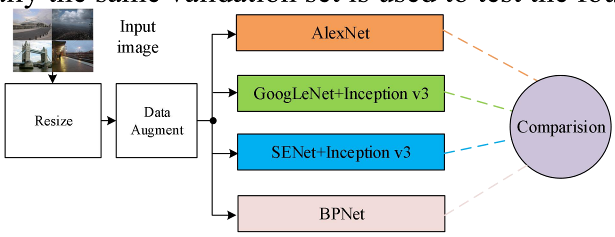

Figure 5: The Process of Experiment.

To evaluate the framework implemented in this paper and test the effectiveness of the framework constructed, our experiment was implemented on a cloud server within the PyCharm Community software of version 2019, Python software of version 3.7.4 and Anaconda software of version 3.0 on a personal computer equipped with INTEL® CORE™ i7-8700k processor, 16 GB memory, and NVIDIA GEFORCE 1080TI*4 GPU. Then we used MATLAB® 2018 to compare the four kinds of frameworks. The evaluation index used in this paper is the accuracy, the F1 value of the accuracy and the recall, and the recognition precision of recognizing extreme weather. The specific experimental process is as follows:

4.1. Accuracy & F1 Test

During the training process, the batch size is set to 128 and the epoch is set to 200. For the optimization parameters, set the learning rate to 0.002, beta_1 to 0.9, and beta_2 to 0.999[20], the hyperparameter reduction ratio r of SENet-Inception v3 is set to 16, and the four networks are trained and tested with the same set parameters. The relationship between accuracy, loss curve and epoch is shown in Figure 6 and Figure 7. When epoch reached 200, the validation accuracy of each network is shown in Table 1. After comparing the experimental results, it was found that under the premise of the same batch size and the same learning rate, during the 200 epochs, the loss curve of SENet's training process was more stable, and the recognition accuracy of its validation set reached 98.43%, which means this framework performs better than other three. Although the loss curve of the GoogLeNet-Inception v3 framework decreases rapidly during the initial iteration, and its subsequent performance is relatively stable, its accuracy is lower than that of SENet. While the epoch of AlexNet and BPNN reach 100, the fluctuations began to appear from time to time, indicating that the model is not stable enough and the validation accuracy is relatively lower than the first two frames. In contrast, SENet tends to be stable after 140 epochs, and the recognition accuracy of the validation set is basically stable above 94.3%, which shows that under the conditions of batch size=128, epoch=200, learning rate= 0.002, beta_1= 0.9, and beta_2=0.999, it is found through comparative experiments that the SENet framework can better fulfill the requirements proposed in this paper under the same conditions. Compared with the shallow network BPNN, the deep network can obtain a higher recognition accuracy, and the SENet&Inception v3 framework can obtain a higher validation accuracy than other frameworks in training, and its average F1 value is higher than the other three, which proves the comprehensive characteristics of the recognition accuracy and recall of the SENet&Inception v3 framework.

Table 1: Test Result when epoch=200.

Frame	Type	Accuracy	F1 score
GoogLeNet	Deep	0.958537	0.95
AlexNet	Deep	0.949546	0.92
SENet	Deep	0.984864	0.98
BPNN	Shallow	0.938446	0.95

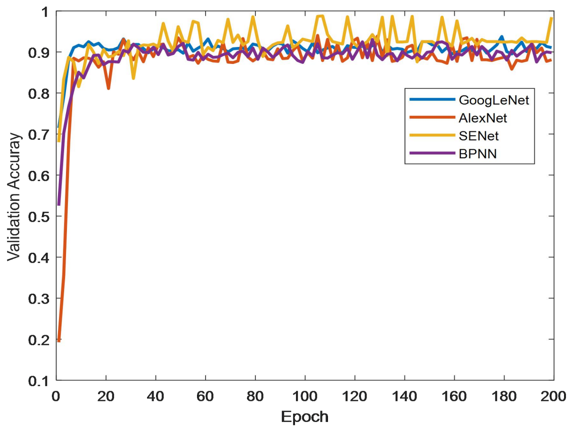

Figure 6: Validation accuracy.

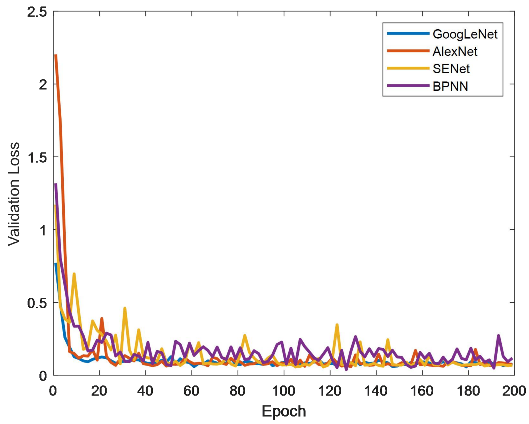

Figure 7: Validation loss.

4.2. SENet-Inceptionv V3 Hyperparameter R Test

During the training process, the batch size is set to 128 and the number of trainings is set to 200. For the optimization parameters, set the learning rate to 0.002, beta_1 to 0.9, beta_2 to 0.999, and test the recognition accuracy while the hyperparameter attenuation ratio r of SENet is set to 16, 32, 64 and 128 within epoch set to 200. The recognition accuracy of the SENet-Inception v3 is shown in Figure 8.

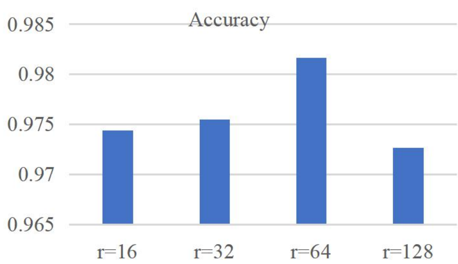

Figure 8: Test results of SENet with hyperparameter r when epoch =200.

It can be seen from the figure that when the reduction ratio *r* is increased, the validation accuracy is improved, and it is on the rise. A smaller scale significantly increases the size of the model's parameters, which increases the complexity of the framework. The increase in complexity does not improve performance monotonously, and the increase in r reduces the complexity of SENet, reduces the amount of calculation, and reduces the computational burden of SENet, so the accuracy rate increases. When the ratio r increases to 128, the validation accuracy has decreased, but compared with r set to 16, it saves a lot of calculation costs. Therefore, we can adjust different ratios r to weigh the accuracy and calculation burden.

4.3. Comparison of Recognition Rates for Extreme Weather

In this paper, we describe the darker weather types: Cloudy_Dark, Foggy_dark, Rainy_dark, Snowy_dark as the extreme weather types that have a greater impact on traffic conditions. Under the condition of epoch=200, the extreme weather recognition precision of each network is calculated through the confusion matrix of four networks. The confusion matrix tested by SENet drawn by Matpolid is shown in Figure 9. The confusion matrix of each frame is used to calculate the recognition precision of each weather and light condition category as shown in Table 2.

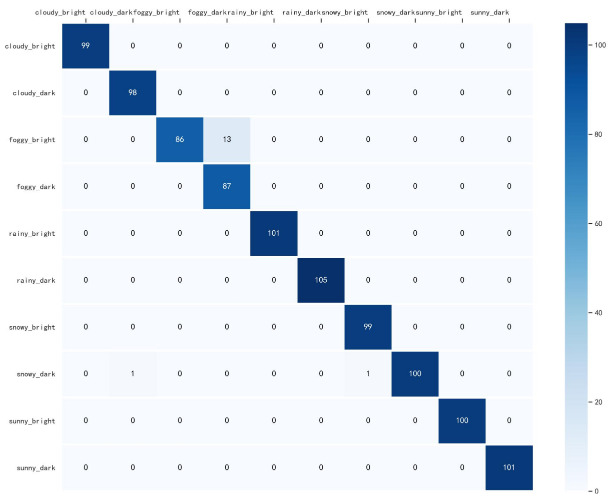

Figure 9: Confusion matrix of SENet.

Table 2: Precision of four framework.

WeatherType	Cloudy		Foggy		Rainy		Snowy		Sunny	
LightCondition	Bright	Dark	Bright	Dark	Bright	Dark	Bright	Dark	Bright	Dark
AlexNet	100%	98.96%	85.10%	92.70%	100%	84.00%	100%	99.00%	100%	99.01%
GoogLeNet+ Inception	100%	100%	93.25%	96.91%	100%	83.33%	100%	100%	100%	100%
SENet+Inception	100%	100%	93.33%	97.91%	100%	100%	100%	100%	100%	100%
BPNN	98%	95.3%	82.9%	90.8%	100%	83.11%	98.32%	95%	100%	94.2%

It can be concluded from Table 2 that SENet has more advantages in identifying extreme weather. Unlike SENet, AlexNet and GoogLeNet do not obtain an association between Channels, and only aggregate information on the local area and feature dimensions on the feature map. Although GoogLeNet deepens and broadens the framework of convolutional neural networks based on AlexNet, the essence is to use a series of convolution operations to make the network feel the global information, which cannot reflect the different importance between Channels. SENet strengthens the characteristics of important channels and weakens the characteristics of non-important channels. Therefore, it can obtain higher recognition accuracy when recognizing extreme weather.

5. Conclusions

This paper applies SEnet to the Recognition of extreme weather, and can determine its light conditions. Using SENet&Inception v3 to build a framework that meets the requirements, and compared with the three networks of AlexNet, GoogLeNet, and BPNN, it is concluded that the recognition performance of the SENet&Inception v3 framework can meet the needs of practical applications, and has great application value in areas such as autonomous driving, intelligent transportation, and climate assistance. In the next work, the image dataset constructed in this paper will be extended to more light conditions, such as: Daylight, Darkness, Twilight, etc. In terms of weather categories, subdividing categories such as Slightly foggy, moderately foggy, severely foggy to refine the weather conditions is more conducive to fine-tuned control in the field of intelligent traffic.

Acknowledgments

This work was supported by National innovation and entrepreneurship training program for Undergraduate of Wuhan University of Technology (S201910497057).

References

[1] Jägerbrand, Annika K. , and J. Sjöbergh . "Effects of weather conditions, light conditions, and road light on vehicle speed." SpringerPlus 5.1(2016):505.

[2] Kang, Li-Wei & Chou, Ke-Lin & Fu, Ru-Hong. (2018). Deep Learning-Based Weather Image Recognition. 384-387. 10.1109/IS3C.2018.00103.

[3] Zhu, Ziqi , et al. "Extreme Weather Recognition Using Convolutional Neural Networks." 2016 IEEE International Symposium on Multimedia (ISM) IEEE, 2016.

[4] C.Lu, D.Lin, J.Jia, and C.K.Tang .2016. Two-Class Weather Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence PP, 99(2016) ,1–1. DOI: https://doi.org/10.1109/TPAMI.2016.2640295

[5] M.Elhoseiny, S.Huang, and A.Elgammal. 2015. Weather classification with deep convolutional neural networks. In ImageProcessing (ICIP), 2015 IEEE International Conference on. 3349–3353.

[6] Li X , Wang Z , Lu X . A Multi-Task Framework for Weather Recognition[C]// ACM, 2017.

[7] Wang, Jiang , et al. "CNN-RNN: A Unified Framework for Multi-label Image Classification." (2016)

[8] Li, Xuelong , Z. Wang , and X. Lu . "A Multi-Task Framework for Weather Recognition." ACM, 2017.

[9] Zhao, Bin & Liu, Wei & Lu, Xiaoqiang & Wang, Zhigang. (2019). A CNN-RNN Architecture for Multi-Label Weather Recognition.

[10] Dannheim, Clemens & Icking, Christian & Mader, Markus & Sallis, Philip. (2015). Weather Detection in Vehicles by Means of Camera and LIDAR Systems. 186-191. 10.1109/CICSyN.2014.47.

[11] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[12] Ullah K R , Xiaosong Z , Rajesh K . Analysis of ResNet and GoogLeNet models for malware detection[J]. Journal of Computer Virology and Hacking Techniques, 2018.16Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, 7132-7141.

[13] C. Szegedy W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[14] Zhu, Fu Zhen , et al. "Research on Image Super-Resolution Reconstruction Based on BPNN and RBFNN." Applied Mechanics and Materials 20-23(2010):445-451.

[15] Sheu, Bing J. , and J. Choi . Back-Propagation Neural Networks. Neural Information Processing and VLSI. Springer US, 1995.

[16] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, 7132-7141.

[17] Y Bian, J Shen, X Xiong, Y Li, W He and P Li. Crowd Counting via Enhanced Feature Channel Convolutional Neural Network. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 2019, pp. 824-831.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 2818-2826.

[19] Fu, Xueyang , et al. "Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal." IEEE Transactions on Image Processing 26.6(2017):2944-2956.

[20] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.