A Traffic Sign Recognition Model with Only 140 KB

Luo Dawei
Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China, 100101
1361722864@qq.com

Fang Jianjun
College of Urban Rail Transit and Logistics, Beijing Union University, Beijing, China.100101
jianjun@buu.edu.cn

Yao Dengfeng
Beijing Key Laboratory of Information Service Engineering, Bejing Union University, Bejing, China,100101
tjtdengfeng@buu.edu.cn

ABSTRACT
To design a sign recognition model with low computational complexity and Low parameter quantity, we uses Group Convolution to compress the parameters, and designs extreme block to solve the problem that the number of input channels of Group Convolution must be equal to the number of output channels and that the feature can not be extracted across channels. In this paper, the number of convolution kernels is set according to the number of classifications. Finally, the original 30 MB CifarNet is compressed into a 140 KB classification model. And we tested it on the BelgiumTS Dataset. The experimental test results show that after the model size is compressed to the original 1/220, top1 is not reduced, but it is increased by 87.31%, and top5 is increased by 0.5%. Experiments prove that the compression strategy is effective. And the experiment also explored the relationship between the number of convolution kernels and the number of classifications.

CCS Concepts
• **Computing methodologies → Artificial intelligence → Computer vision → Computer vision problems → Object recognition.**

Keywords
Group Convolution; feature extraction; channel adjustment; image classification; model compression.

1. INTRODUCTION
Traffic sign recognition is a relatively basic category in the field of automatic driving[1], and automatic driving requires that traffic signs be recognized quickly and accurately [2]. Therefore, a traffic sign recognition model with low computational complexity and small computational complexity is necessary [3].

Traffic sign recognition in the early stage mainly used traditional methods[4,5,6], because they are fast. For example, De l E A, et al. [7] use the color threshold method to recognize the traffic sign. Jeraj G [8] uses color segmentation and locates key points of traffic signs to identify. Maldonado-Bascon S, et al. [9] used support vector machines (SVM) to combine image color and shape feature extraction for recognition. There are many other methods, but most of them are to extract the color and shape of the sign to complete the recognition. Actual road conditions are often affected by shooting angles, lighting conditions, and occlusion problems. Traditional methods cannot cope with this problem, so the Classification algorithm based on convolutional neural networks (CNN) is becoming popular. Á ArcosGarcía, et al. [10] designed a deep neural network for traffic signs by using convolution layer and space transformer network. Kwangyong L, et al. [11] proposed a real-time traffic sign detection and recognition method based on general graphics processing unit (GPGPU) for illumination effects. Shi-hao Yin, et al. [12] designed a new TSR system design method based on deep convolutional neural network. The traffic sign recognize methods based on convolutional neural network has been well explored, and the accuracy of most models has reached more than 90% [13,14,15]. However, in order to be able to adapt to various devices and environments, we expect the model to be as lightweight as possible while maintaining accuracy. In this paper, a traffic sign recognition model with high accuracy and low computational complexity is designed by combining Group Convolution [16] and ordinary convolution.

There are three contributions:

a) The block is designed by ordinary convolution and grouping convolution, which greatly reduces model parameters and computation.

b) Design experiments to explore the relationship between the number of convolution kernels in convolution neural network and the number of categories in classification tasks, which greatly reduces the redundancy of convolution kernels.

c) A traffic sign recognition model with Top1 reaching 91.47% and model size less than 140 KB is designed.

2. OUR METHOD
We selected CifarNet from the darknet projects as the baseline，because of its simple structure and high classification accuracy. Group Convolution can effectively reduce the model parameters, so we use Group Convolution instead of ordinary convolution. In addition, we have designed several structures to gradually explore the relationship between the number of convolution kernels and the number of classifications, and to find the most suitable model.

2.1 Group Convolution
Group Convolution was first proposed in AlexNet to separate networks. In this paper, Group Convolution is used to group input feature maps, and then each group is convoluted separately to reduce the parameter budget. And our Group Convolution has two characteristics,

a) The number of input features = the number of groupings = the number of convolution kernels = the number of output features;

b) The size of convolution kernels is 3×3.

RSVT '19, October 16–18, 2019, Wuhan, China

ACM ISBN 978-1-4503-6242-9/19/03…$15.00
DOI: https://doi.org/10.1145/3366715.3366723

The structure of Group Convolution is shown in the Fig 1. Each ordinary convolution kernel needs feature extracted from all feature maps. But our Group Convolution is different. Each convolution kernel only needs to extract features from its corresponding feature map, which makes the size of the feature extraction convolution kernels only 1/C of the ordinary convolution, where C represents the number of input feature maps. After convolution, the input feature maps are connected by concatenate operation. In this way, the number of input feature maps is equal to that of output feature maps.

However, there are two drawbacks in this structure. one is, the features extracted by convolution kernel are only extracted on their respective feature maps. Group Convolution can not obtain information on different feature maps, that is, it can not extract features across channels. the other is Group Convolution can not extract more or fewer features because it requires input features to be etc. Output characteristics. To solve this problem, we designed extreme block, which will be described in detail below.

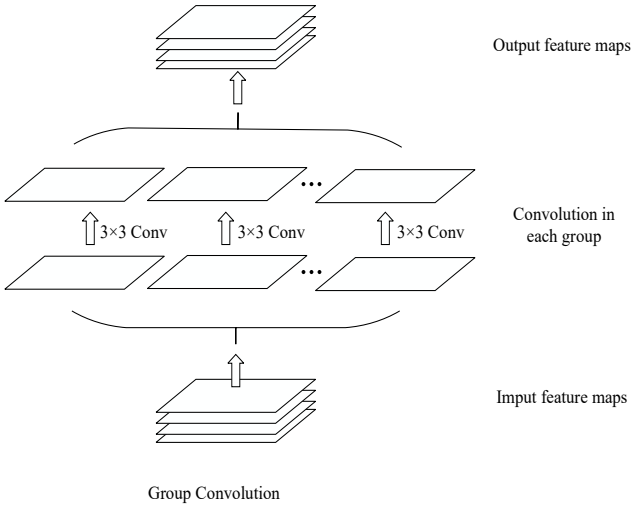


Figure 1. Group Convolution

In addition, To minimize the number of model parameters, we use Group Convolution with stride = 2 to replace maxpool for downsampling. Another advantage of this method is that it can extract features to some extent while downsampling.

2.2 Extreme Block

As shown in Figure.2. Because Group Convolution can't extract features across channels and can't change the number of output features, we designed a block, we named it extreme block, because it is designed to pursue the extreme lightweight model.

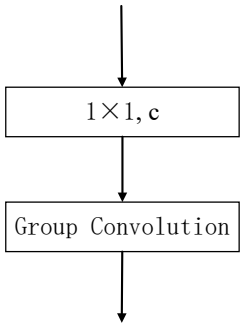


Figure 2. extreme block

Extreme block is mainly composed of a 1×1 convolution and a 3×3 grouping convolution. 1×1 convolution is used to adjust the number of channels of feature maps and obtain cross-channel information, while increasing the non-linearity of feature extraction. 3 × 3 grouping convolution is used for feature extraction.

When we need to extract different feature numbers, we can transform the channel number by convolution of 1×1 in extreme block, and then transfer it to Group Convolution. When we do not need to change the number of extracted features, 1×1 convolution can also be used for cross-channel information transmission, avoiding Group Convolution extracting a single feature.

2.3 Model's Structure

We chose CifarNet from the darknet project as the baseline, because of its small model and high accuracy. Its structure is shown in the Table 1. Firstly, a crop layer is used to transform the input image into a 28×28×3 feature map. Then, three stages are used to extract features. Each stage contains three convolutions of 3×3. The number of features extracted by three stages is 128, 256 and 512 in turn. Finally, the feature maps are compressed to 62 dimensions by a 3×3 convolution, and a 62-dimensional real number is obtained by avgpool, which is input to a softmax for classification.

In the first step of compression, we replace the convolutional layer of each stage in CifarNet with an extreme block, each block containing a 1×1 convolution and a 3×3 Group Convolution. Then replace maxpool with the Group Convolution of stride=2 to reduce the parameter operation. We named it CifarNet-Extreme1.

In the second step of compression, we switched the number of features extracted from the three stages in CifarNet-Extreme1 from 128, 256, 512 to 32, 64, 128. We analyze that the final output of this classification model is 62 types of targets, if each category requires one or more features. Then the model should not have too many convolution kernels beyond the number of classifications. Therefore, we first reduce the number of convolutional kernels by 50%. This step is also used to confirm whether lowering the number of convolution kernels will significantly affect the accuracy of the model, and if so, to indicate that the model is very bloated and that the compression strategy will fail. We named it CifarNet-Extreme2.

In the third step of compression, we switch the number of features extracted from 32, 64, 128 to 31, 62, 62, from 32, 64, and 62 in CifarNet-Extreme2. We found that almost all CNN-based classification models need to convert the extracted features into a set of real numbers before the final output layer, whose length is equal to the number of categories. Therefore, we wonder whether the number of convolutional kernels in an efficient, extreme-oriented model can be equal to the number of classified. So, an extremely light model was designed to be tested, we named it CifarNet-Extreme3.

In the fourth step of compression, we switched the number of features extracted from 31, 62, 62 to 31, 31, 31, and we also designed a more extreme model to reduce the number of convolution kernels as more as possible. We named it CifarNet-Extreme4. The model is also used to explore whether the maximum convolutional number can be lower than the number of categories.

Table 1. structure of models

CifarNet	CifarNet-Extreme1	CifarNet-Extreme2	CifarNet-Extreme3	CifarNet-Extreme4
Crop Layer: 64 x 64 -> 28 x 28 x 3				
Conv3-128 Conv3-128 Conv3-128	Block-128×3	Block-32×3	Block-31×3	Block-31×3
maxpool	3×3,128,stride=2,C=128	3×3,32,stride=2,C=32	3×3,31,stride=2,C=31	3×3,31,stride=2,C=31
Conv3-256 Conv3-256 Conv3-256	Block-256×3	Block-64×3	Block-62×3	Block-31×3
maxpool	3×3,256,stride=2,C=256	3×3,64,stride=2,C=64	3×3,62,stride=2,C=62	3×3,31,stride=2,C=31
Conv3-512 Conv3-512 Conv3-512	Block-512×3	Block-128×3	Block-62×3	Block-31×3
Conv3-62				
avgpool				
softmax				

3. EXPERIMENTS

In order to get a lightweight traffic sign recognition model, which can be used in various scenarios or devices, we designed a 140 KB model based on cCifarNet and tested it on Belgium TS Dataset [17].

3.1 Dataset

We used the classification part of Belgium TS Dataset for testing. The data set contains 62 categories of Belgian tags, a total of 7095 pictures and labels, including 4575 training sets and 2520 test sets. Some samples are shown in Fig 3.

Figure 3. Belgium TS Dataset

3.2 Training

The training of these models is as shown in the Fig 4. CifarNet's loss curve declined rapidly in the first 3000 iterations and then steadily declined until 35000 iterations stabilized at 0.1903. Its top5 curve reached 98% in 5000 iterations, and remained stable until 98%. This shows that the training of CifarNet is good, the model converges eventually, and the training is effective, which can be used for testing.

The loss curve of the compressed model converges well. The loss of CifarNet-Extreme1 is stable at 0.1428, that of CifarNet-Extreme2 is stable at 0.2601. CifarNet-Extreme1's loss is stable at 0.3291 and that of CifarNet-Extreme1 is stable at 0.3854.

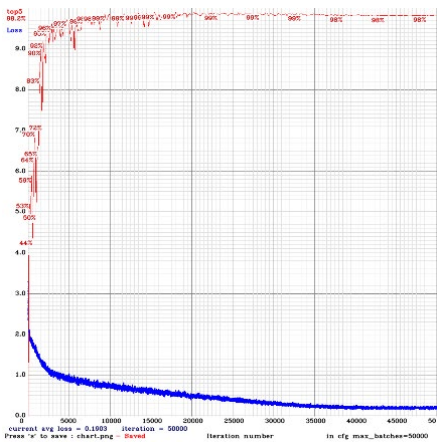
The compressed model, whose top5 curve is much more cluttered in the early oscillation than the original CifarNet. Among them, CifarNet-Extreme1 had fewer shocks, and its top5 curve reached 98% in its 3000 iterations, but then it once dropped into 72%. there is no sign of convergence untill 17000 iterations and the model eventually converges to 98%. the early shocks of CifarNet-Extreme2's top5 curve were more intense than others, which reached 98% in iterations 10,000 times, and the shocks continued to 22,000, but it eventually converges to 98% like CifarNet-Extreme1. The early shocks of CifarNet-Extreme3 is increased, but its top5 curve reached 99% in 14,000 iterations and began to stabilize at 20,000 times, eventually converging to 99%. The CifarNet-Extreme4's early shock was the worst, with its top5 curve reaching 98% in 14,000 iterations and, surprisingly, its top5 curve eventually converge at 99%.

In terms of loss, it seems that CifarNet-Extreme1 is the best performer. But CifarNet-Extreme3 is the best performer on the top5 curve.

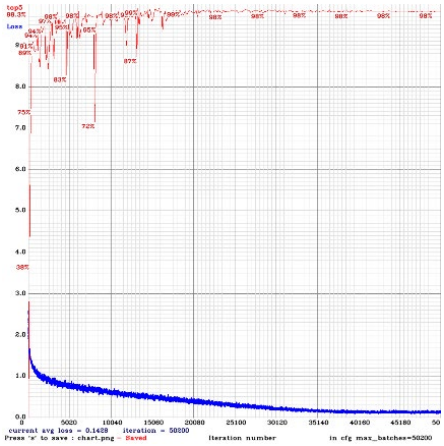
3.3 Results

The accuracy analysis of these models is shown in Table 2, and we use top1 and top5 as the accuracy evaluation indicators in this paper. top1 refers to the largest of the probability vectors in the model output as the classification result, and if the classification of the largest probability in the classification result is correct, the classification is correct. Otherwise, the classification is incorrect. top5 refers to the top five with the largest probability vector, which is classified correctly as long as the correct probability appears.

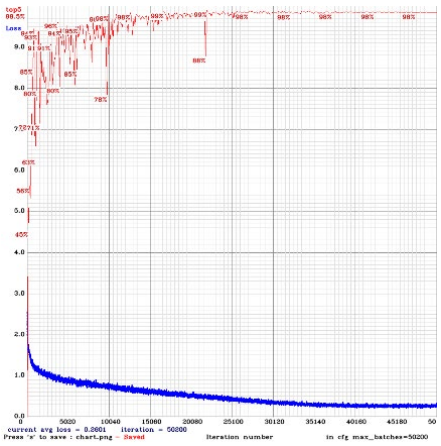
CifarNet's top1 is 90.60% and top5 is 98.21%. After the first compression, CifarNet-Extreme1's top1 is 90.28%, top5 is 98.25%, top1 decreased by 0.32%, and top5 increased by 0.04%. There is not much change in accuracy, but this paper aims to compress the model, and it is a good phenomenon that there is no significant decrease in accuracy.



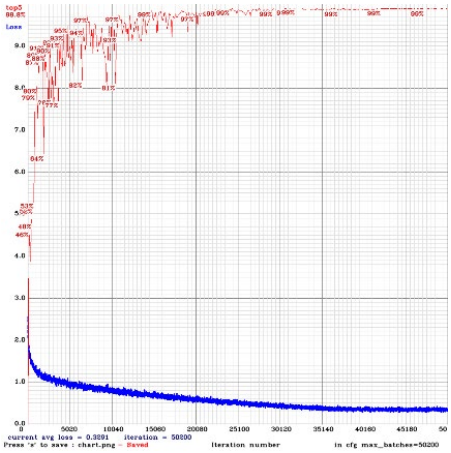
(a) trianing of CifarNet



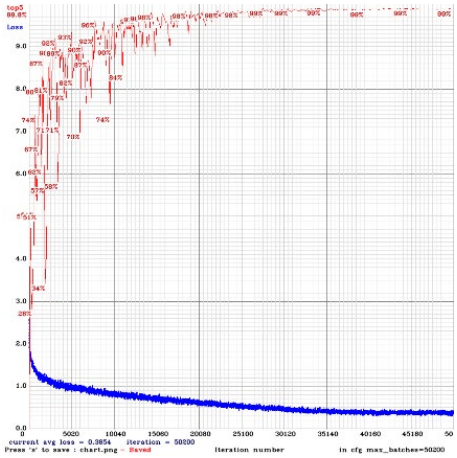
(b) trianing of CifarNet-Extreme1



(c) trianing of CifarNet-Extreme2



(d) trianing of CifarNet-Extreme3



(e) trianing of CifarNet-Extreme4

Figure 4. training rendering

After the second step of compression, CifarNet-Extreme2's top1 is 90.68%, top5 is 98.49%, top1 is increased by 0.08%, and top5 is increased by 0.28%. Although the accuracy is not much improved, it is very significant that both top1 and top2 are improved.

Table 2. ACC of these models

model	Top1(%)	Top5(%)
CifarNet	90.595	98.214
CifarNet-Extreme1	90.278	98.254
CifarNet-Extreme2	90.675	98.492
CifarNet-Extreme3	91.468	98.770
CifarNet-Extreme4	88.373	98.810

After the third step of compression, CifarNet-Extreme3's top1 is 91.47% and top5 is 98.77%. This is a very surprising result, top1 increased by 0.87% and top5 increased by 0.6%.

After the fourth step of compression, CifarNet-Extreme4's top1 is 88.37% and top5 is 98.81%. This compression is not good, top1 dropped by 2.22%, although top5 increased by 0.6%.

Let's look at their parameter size analysis, it is shown in Table 3. CifarNet's BFLOPS is 1.627 and its model size is 29.4MB. Compared to CifarNet, CifarNet-Extreme1 is compressed by about 90%, its BFLOPS is 0.194, and its model size is 3.56MB.

CifarNet-Extreme2 is another large-scale compression. Its BFLOPS is 0.015, and its model size is 289KB, which is only 1% of CifarNet. CifarNet-Extreme3 has a BFLOPS of 0.011 and a model size of 141KB, which is only half of CifarNet-Extreme2 and one-hundredth of CifarNet. CifarNet-Extreme4 is a very extreme network with a BFLOPS of 0.005 and a model size of 100KB.

We draw an ACC-Model Size diagram with the model size as the x-axis and the accuracy as the y-axis, as shown in the Fig 5. CifarNet-Extreme3 is closest to the upper left corner in the Fig 5,, which means that it has the best performs in these models. Its model size is only 140KB, which is 1/200 of the original model, and its top1 is the highest, reaching 91.47%, which is 0.87% more than the original one.

Table 3. Model size of these models

model	BFLOPS	Model size (KB)
CifarNet	1.627	30132
CifarNet-Extreme1	0.194	3652
CifarNet-Extreme2	0.015	289
CifarNet-Extreme3	0.011	141
CifarNet-Extreme4	0.005	100

3.4 Others

We also tried to add Spatial Pyramid Pooling (SPP) to improve their accuracy, but failed. We think that global and local features are helpful in the performance of image classification, especially sign recognition, but the experiment cruelly tells us no. We analyzed perhaps the spatial scale information extracted by SPP when using the different size maxpool spawned some kind of bad interference, which destroyed the characteristic information.

3.4.1 Adding Spatial Pyramid Pooling

We also tried to add Spatial Pyramid Pooling (SPP) to improve their accuracy, but failed. We think that global and local features are helpful in the performance of image classification, especially sign recognition, but the experiment cruelly tells us no. We analyzed perhaps the spatial scale information extracted by SPP when using the different size maxpool spawned some kind of bad interference, which destroyed the characteristic information.

3.4.2 Deleting some 1×1 convolutions

In addition, we also try to delete some 1×1 convolutional layers without channel number adjustment. However, after deletion, the accuracy is greatly reduced. We analyze that these 1×1 convolutional layers do not adjust the channel number, but they increased cross-channel information extraction to improve network performance. And they increased depth and nonlinearity of feature extraction, too. They still have important influences and cannot be deleted directly.

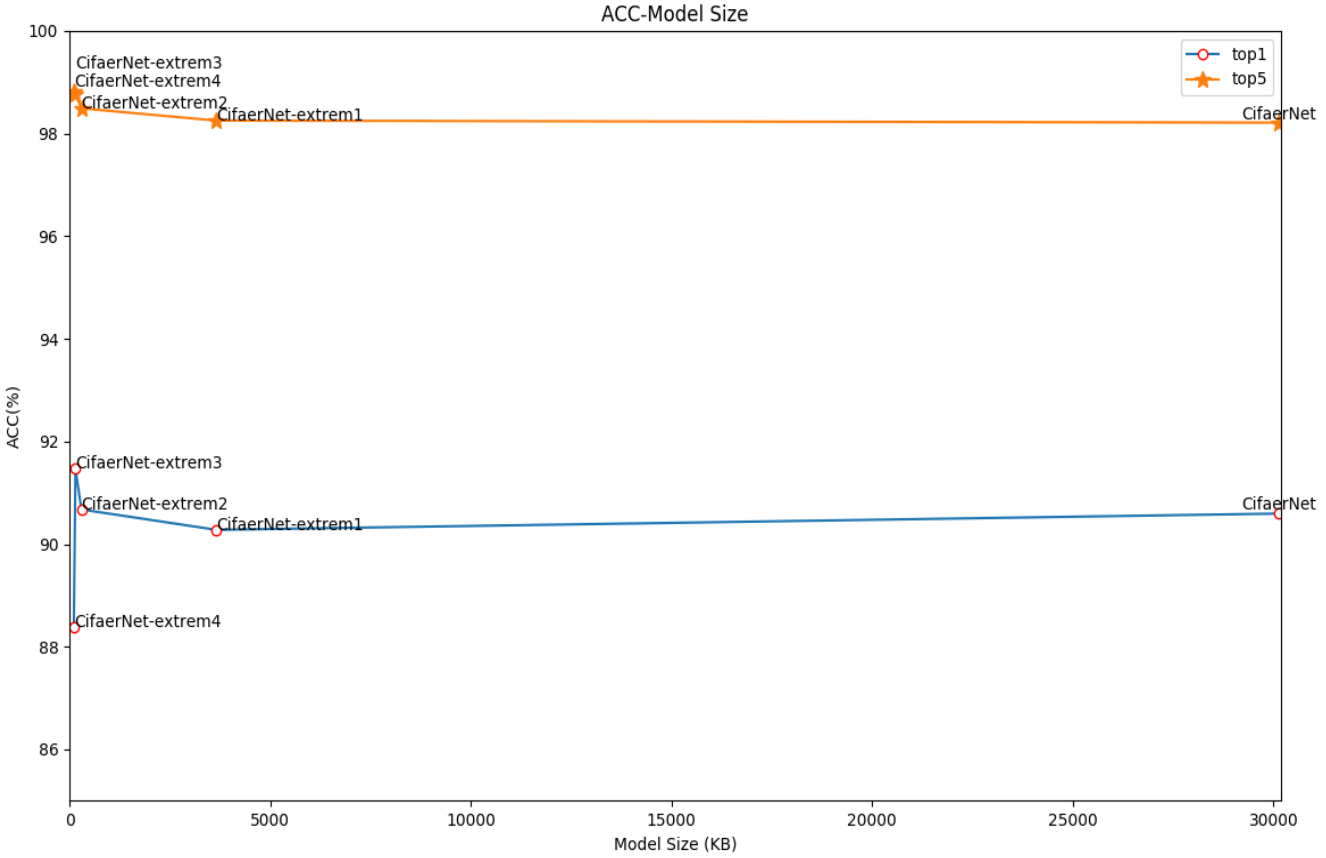


Figure.5 ACC-Model Size diagram

4. CONCLUSION

In this paper, a lightweight feature extraction module block is designed by ordinary convolution and Group Convolution, and the convolutional layer in CifarNet is replaced by a convolutional layer. Finally, a high-precision and extremely light traffic sign recognition model is designed. In addition, the design experiment on the BelgiumTS Dataset explores the relationship between the number of convolution kernels and the number of categories in the classification task. Experiments show that there is a lot of redundancy in the convolution nucleus in CifarNet, and the number of convolution kernels does not need to exceed the number of categories. But the experiment is only tested on CifarNet and BelgiumTS Dataset, and we will further explore the validity of the theory in the ImageNet dataset and other classification models in the further.

5. ACKNOWLEDGMENTS

Thank the authors of Belgium for publishing their dataset so that we can test our method on it; thank the authors of Darknet for making it easy for us to do our experiment.

6. REFERENCES

[1] Escalera A D L, Armingol J M A, Mata M. Traffic sign recognition and analysis for intelligent vehicles[J]. Image Vis Comput, 2003, 21(3):247-258.

[2] Luo H, Yi Y, Bei T, et al. Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, PP(99):1-12.

[3] Aghdam H H, Heravi E J. Traffic Sign Detection and Recognition[M]// Guide to Convolutional Neural Networks. 2017.

[4] Zaklouta F, Stanciulescu B. Real-time traffic sign recognition in three stages[J]. Robotics & Autonomous Systems, 2014, 62(1):16-24.

[5] Stallkamp J, Schlipsing M, Salmen J, et al. Man vs. Computer: benchmarking machine learning algorithms for traffic sign recognition.[J]. Neural Netw, 2012, 32(2):323-332.

[6] Fleyeh H, Davami E. Eigen-based traffic sign recognition[J]. Iet Intelligent Transport Systems, 2011, 5(3):190-0.

[7] De l E A , Moreno L E , Salichs M A , et al. Road traffic sign detection and classification[J]. IEEE Transactions on Industrial Electronics, 1997, 44(6):0-859.

[8] Jeraj G . Traffic sign recognition system[J]. Journal of Physics Condensed Matter An Institute of Physics Journal, 2011, 23(34):345403.

[9] Maldonado-Bascon S , Lafuente-Arroyo S , Gil-Jimenez P , et al. Road-Sign Detection and Recognition Based on Support Vector Machines[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(2):264-278.

[10] Á ArcosGarcía, Álvarezgarcía J A, Soriamorillo L M. Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods.[J]. Neural Netw, 2018, 99(12):158-165.

[11] Kwangyong L , Yongwon H , Yeongwoo C , et al. Real-time traffic sign recognition based on a general purpose GPU and deep-learning[J]. PLOS ONE, 2017, 12(3):e0173317-.

[12] Shi-hao Yin, Ji-cai Deng, Da-wei Zhang, et al. Traffic sign recognition based on deep convolutional neural network[J]. Optoelectronics Letters, 2017, 13(6):476-480.

[13] Jain A, Mishra A, Shukla A, et al. A Novel Genetically Optimized Convolutional Neural Network for Traffic Sign Recognition: A New Benchmark on Belgium and Chinese Traffic Sign Datasets[J]. Neural Processing Letters, 2019:1-25.

[14] Rosario G, Sonderman T, Zhu X. Deep Transfer Learning for Traffic Sign Recognition[C]// 2018:178-185.

[15] Yin Shihao, Deng Jicai, Zhang Dawei, et al. Traffic sign recognition based on deep convolutional neural network[J]. Optoelectronics Letters, 2017, 13(6):476-480.

[16] Xie S, Girshick R B, Dollar P, et al. Aggregated Residual Transformations for Deep Neural Networks[C]. computer vision and pattern recognition, 2017: 5987-5995.

[17] Mathias M, Timofte R, Benenson R, et al. Traffic sign recognition — How far are we from the solution?[C]. international joint conference on neural network, 2013: 1-8.