Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Traffic sign detection and recognition based on CNN-ELM

To cite this article: Wenju Li *et al* 2021 *J. Phys.: Conf. Ser.* **1848** 012106

View the article online for updates and enhancements.

This content was downloaded from IP address 181.214.165.38 on 13/04/2021 at 14:18

Traffic sign detection and recognition based on CNN-ELM

Wenju LI[1*], Xinyuan NA[1], Pan SU[1] and Qing ZHANG[1]

[1]Collage of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, 201418, China

*Corresponding author e-mail:wjli@sit.edu.cn

Abstract. Aiming at the problem of low traffic sign recognition rate and slow speed, a traffic sign recognition algorithm combining CNN and Extreme Learning Machine is proposed. First, the ResNet50 network is used to extract image features, and then the Region Proposal Network (RPN) is used to generate proposals from the extracted image feature maps. Finally, the extreme learning machine is used to classify the generated proposals, and the fully connected layer is used for regression prediction. The experiment shows that compared with the Faster R-CNN model, the CNN+ELM improves the recognition accuracy on the TT-100K dataset 7.7% and reduces the training time per epoch by 32 seconds.

1. Introduction

The detection and recognition of traffic signs is the key to the development of intelligent driving technology. The complex and changeable actual scenes have brought great challenges to the detection and recognition of traffic signs. In the process of traffic sign recognition, it mainly includes two stages: detection and classification. Traditional traffic sign recognition methods need to use artificial feature to extract regions of interest, and the accuracy is low. In 2013, Liang et al. [4] proposed a template matching method based on the shape characteristics of traffic signs. In 2016, Y. Xu et al. [5] studied the method of using ELM as a classifier for detecting traffic signs. The ELM is a single hidden layer feedforward neural network with a total of 3 layers of network structure. The algorithm has good generalization performance under the premise of extremely fast learning speed [6]. In 2018, D. Yasmina et al. [7] proposed a road traffic sign recognition method based on deep learning, using an improved LeNet-5 network to extract the deep representation of traffic signs for recognition. At present, deep learning methods have become one of the important methods used in object detection, especially small object detection and recognition.

Aiming at the problem of low accuracy and slow speed of traffic sign detection and recognition, a method combining CNN and ELM is proposed. This article is based on the Faster R-CNN network structure to improve and replace the softmax classifier with the improved ELM.

2. Related work

2.1. Object Detection

In recent years, the object detection model based on deep learning has been greatly improved in recognition accuracy and speed. The one-stage method of YOLO series [8, 9, 10, 11] directly predicts the coordinates and category of the object, this type of model treats object detection as a regression task [8]. In addition, there are two-stage methods in the R-CNN series [12, 13, 14, 15], which treat object

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

detection as a classification task, and first use proposal algorithms, such as Selective Search, RPN, detect the proposals where the object may exist, and then use the multi-class classifier to classify the proposals.

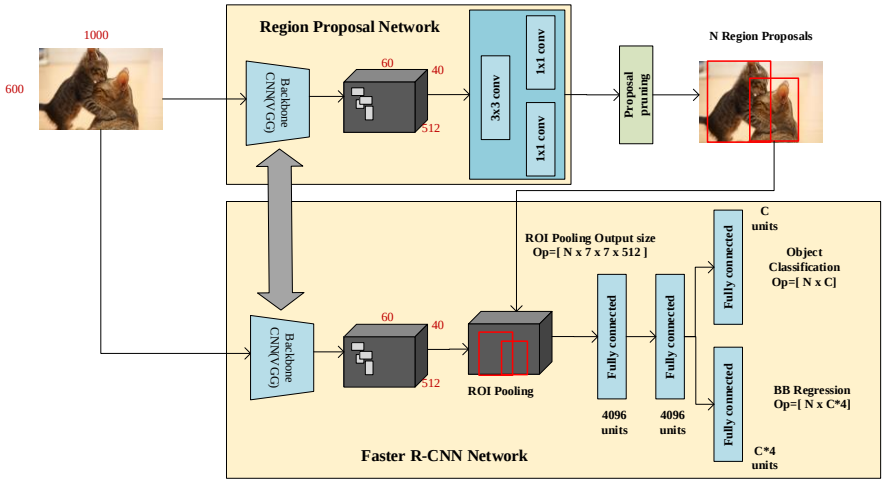


Figure 1. Faster R-CNN network structure

Faster-R-CNN is a classic two-stage object detection algorithm and it's consists of two parts (Figure 1), the first part is the RPN, which is used to generate proposals from the picture. The second part is the Fast R-CNN classifier, which extracts features from the feature map according to the proposals, and performs multi-categories classification and positioning regression on the proposals. Figure 1 is the network structure of Faster R-CNN, using VGG16 or ZF-net network as the feature extractor, then RPN uses a fully convolutional network to generate proposals, and then uses ROI Pooling and fully connected layers to classify and locate objects in the proposals [15]. In 2020, Fan Q, Zhou W et al. [16] proposed an object detection framework based on Few-Shot learning combined with multi-scale texture relationship and attention mechanism for the object detection problem of small datasets. Wu Y, Chen Y, etc. [17] found that the fully connected layer is more suitable for classification, and the convolutional layer is more suitable for object locating.

The method proposed in this paper is improved on the Faster R-CNN model. In the detection part, ResNet50 is used as the feature extractor, and 5 feature maps are extracted from the backbone to form a feature pyramid. After obtaining the proposals, ELM is used as the final classifier to help the detection and classification model converge quickly, and improve the model's recognition accuracy of traffic signs.

2.2. Extreme learning machine

The softmax classifier is a commonly used for neural networks in classification tasks. ELM is a neural network classifier proposed by G. Huang [6] in 2006, and it is implemented using a three-layer fully connected layer. As shown in Figure 2, assuming that the trainset has only N=1 sample, first use the input layer and the hidden layer to extract the M features of the sample to obtain a feature matrix $F_{N\times M}$. Then take the Moore-Penrose generalized inverse of this N×M matrix, and then take the matrix multiplication of the obtained M×N inverse matrix and the category label $L_{N\times C}$ of the training sample, to obtain the weight matrix $weight_{C\times M}$ of the output layer by transpose the matrix $\beta_{M\times C}$. During training, the network weights of the input layer and the middle layer are randomly initialized, then fixed, and sample features are extracted by them. The weight of the output layer is calculated using the extracted sample features and sample category labels, using the Moor-Penrose generalized inverse method. For any complex matrix $A_{m\times n}$, the Moore-Penrose generalized inverse refers to the generalized inverse matrix $G_{n\times m}$ that simultaneously satisfies the following four conditions:

$$AGA = A \quad (1)$$

$$GAG = G \quad (2)$$

$$(GA)^T = GA \quad (3)$$

$$(AG)^T = AG \quad (4)$$

Among them, the inverse matrix that satisfies the condition (1) is called the minus inverse A^-, satisfies (1), (2) is called the reflexive minus inverse A_r^-, satisfies (1), (3) is called the minimum norm generalized inverse A_m^-, satisfying (1), (4) is called the least square generalized inverse A_l^-. Moore-Penrose generalized inverseA^+, also known as plus inverse, or pseudo-inverse, must be minus inverse, reflexive minus inverse, minimum norm generalized inverse and least square generalized inverse at the same time.

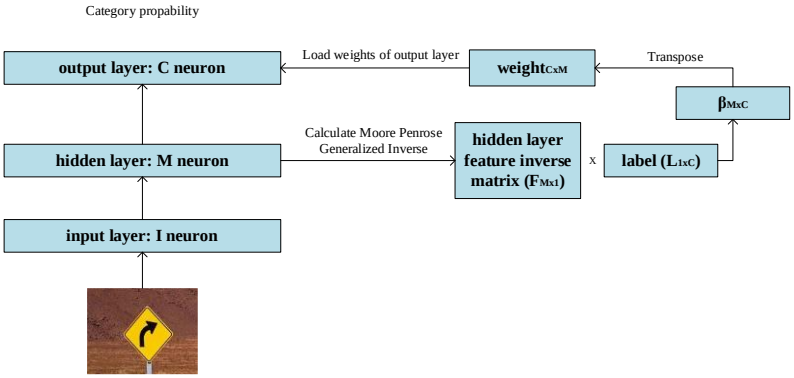


Figure 2. Schematic diagram of the network structure of the extreme learning machine

However, because the ELM algorithm needs to save the features of all samples before updating the output weights, when the training set is large at times, it consumes a lot of memory, so current networks are only used for classification tasks of small-scale datasets. This paper proposes a phased weight update method for extreme learning machines to adapt to the training of classification network models in the case of large-scale datasets.

3. Traffic sign detection and recognition based on CNN-ELM

3.1. Improvements of Faster-R-CNN

In the model proposed in this paper, ResNet50 is used as a feature extraction network, and five feature maps of different scales are extracted from it to form a feature pyramid, and then the RPN generate proposals on the feature maps of 5 different scales, and each scale contains 3 aspect ratios. The RPN network consists of two parallel fully convolutional networks, one is responsible for predicting the coordinates of the candidate area, and the other is responsible for predicting the category (object or background) of the candidate area. RPN can take a feature map of any size as input, and then generate k (default 3) candidate boxes and corresponding object scores at each position on the feature map.

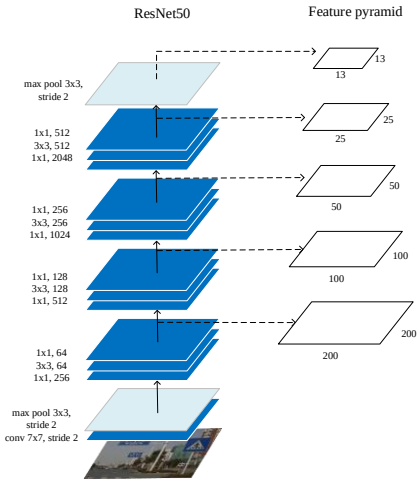
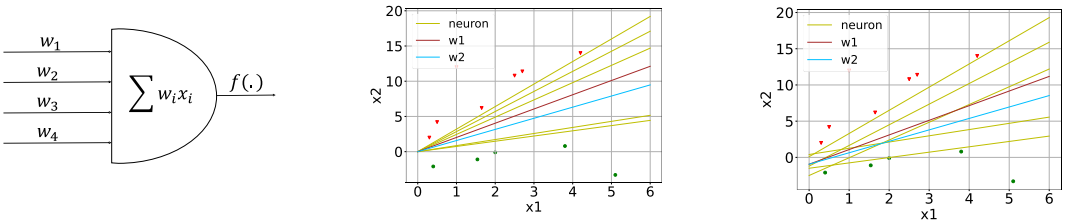


Figure 3. The backbone network and feature pyramid of Faster R-CNN

3.2. ELM's improved classification model

In artificial neural networks, the synapses of biological neurons are simply simulated using linear functions such as $y = wx$, while the entire neuron is simplified using $y = \sum w_i x_i + b$, as shown in Figure 4(a). Consider a linear neuron with a single input and single output. The neuron parameter model at this time is a straight line. In Figure 4(b), there are 5 neurons (yellow straight lines, all biases are 0), and their weights are respectively 3.9, 3.55, 2.45, 0.86, 0.74, the mean is 2.3. In order to get the best classifier, use the mean of the weights as the new classifier, as shown in the brown line w1 in Figure 4(b). However, this curve is more biased towards the left data point, which is not the expected result.



(a) Neuron model (b) Unbiased linear classifier (c) Linear classifier with bias

Figure 4. Neuron structure diagram and neuron-based classifier

In order to avoid obtaining the above-mentioned sub-optimal classifier, in Figure 4(b), a better classifier is to average the weight of each neuron in polar coordinates, and the resulting straight line w2 is the expected classifier. The angle of a straight line in polar coordinates is exactly the arctangent of the slope. In Figure 4(c), when these biases exist, the average value of the biases can be directly calculated. Based on this, the following method is obtained:

$$\theta_{avg} = \frac{1}{N}\sum_{i=1}^{N} atan\,(w_i) \quad (5)$$

$$w_{new} = tan\,(\theta_{avg}) \quad (6)$$

$$b_{new} = \frac{1}{N}\sum_{i=1}^{N} b_i \quad (7)$$

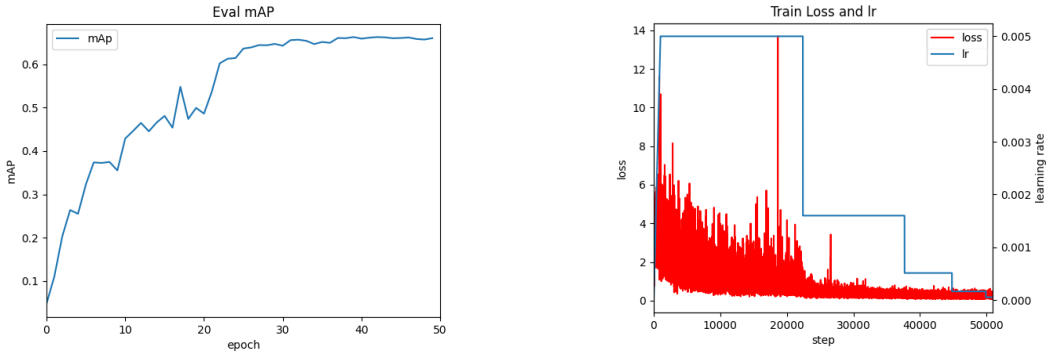
w_{new} is the weight of the new classifier. In 4(b) and 4(c), it appears as the slope of the blue line w2, and b_{new} is the bias of the new classifier, that is, the line w2 in Figure 4(c) bias.

In order to adapt the network to the training method of CNN, the network weights of the ELM output layer are calculated and updated many times. And the input layer of the ELM is removed, leaving only the middle layer and the output layer. During network training, batch size is set to 6, and the weight of the ELM output layer is calculated every 6 iterations, and together with the previous n-1 weights.

4. Experimental results and analysis

4.1. Experimental program

According to the characteristics of small objects in the TT-100K traffic sign data set, the parameters of the Faster R-CNN network are adjusted. First, in order to adapt to the detection of small objects, ResNet50 is used as the feature extraction network, and from the 10th, 22nd, 40th, and 49th layers, the convolutional layer extracts feature maps of different scales to form a 5-layer feature pyramid. Then, the size of the anchors was changed from 32, 64, 128, 256, 512 to 16, 32, 64, 128, 256, and each scale corresponds to each layer of the feature pyramid. In this way, at each position of each layer of the feature pyramid, 3 anchor boxes are generated according to the aspect ratios of 0.5, 1.0, and 2.0.



(a) Training accuracy curve (b) Change curve of learning rate and training loss

Figure 5. Faster R-CNN + ELM training curve.

During training, the batch size of the network is 6. In the weight calculation part of the ELM, the network accumulates the output features of the ELM hidden layer of 20 consecutive network iterations, and then calculates and updates the weight of the ELM output layer every 20 iterations, and record the angle value of the current weight in polar coordinates. When calculating the weight of the ELM network, θ_{avg} is calculated according to the following formula:

$$\theta_{avg} = \frac{iters \times \theta_{old} + \theta_{new}}{iters + 1} \tag{8}$$

Among them, θ_{old}represents the angle value calculated last time, θ_{new} represents the angle value obtained by the current calculation, and *iters* represents the current number of iterations. θ_{old} represents the average value of the ELM network weight angle value during the previous *iters* of the current epoch, and θ_{avg} represents the average value of the network weight angle calculated according to the *iters*.

This experiment uses HP desktop computer, i7 9700K CPU, NVIDIA RTX2080 GPU, 16G memory. The dataset is selected from TT100K. A total of 60 traffic sign categories are selected, a total of 6108 training set pictures, 3072 verification set pictures, batch size is 6, the initial learning rate is 0.005, the learning rate reduce factor is 0.32, and the optimizer uses SGD, the learning rate reduce adopts ReduceLROnPlateau. The experimental results are shown in Figure 5.

From the Average Precision comparison in Table 1, it can be seen that, compared to Faster R-CNN, the combination of Faster R-CNN and ELM further improves the model's detection accuracy for medium and large objects, but the detection accuracy for small objects is slightly reduced. At the same time, the addition of ELM has significantly improved the model's recall rate and training speed, as shown in Table 2, which shows that ELM has also improved the model's detection accuracy.

Table 1. Average Precision comparison

	AP50	**AP75**	**Small**	**Medium**	**Large**
Faster R-CNN[14]	0.583	0.509	**0.283**	0.501	0.605
Faster R-CNN + ELM	**0.660**	**0.539**	0.253	**0.567**	**0.693**

Table 2. Recall, training speed, test speed comparison

	Small	**Medium**	**Large**	**Train speed (*iters/speed*)**	**Test speed (*iters/speed*)**
Faster R-CNN[14]	0.404	0.654	0.752	12.04	**16.96**
Faster R-CNN + ELM	**0.443**	**0.716**	**0.796**	**12.85**	16.91

4.2. Experimental results and analysis

It can be seen from the Table1 that adding an ELM classifier can improve training time, and the accuracy of AP50 and AP75 is also higher. Since the weight of the output layer of the ELM is directly obtained by calculation based on the existing sample features and its category labels, rather than being gradually adjusted through learning, in the entire training process, no matter where the network is trained, ELM can always maintain the accuracy at a high level. Because of this feature of ELM, the model proposed in this article does not need to consider the accuracy of classification too much during training, but only needs to focus on the object locating stage, thus reducing the computational complexity of model training.

5. Conclusion

This paper proposes a solution combining Faster R-CNN and ELM for the detection and recognition of small object traffic signs in real scenes. Experiments have proved that combining ELM and CNN can improve the accuracy of model recognition, reduce training time. However, the addition of the ELM module is not friendly to the detection of small objects. This may be because small objects contain less texture information, and the feature vectors obtained when extracting features are quite different from objects of other sizes. In the next step, we will further research on the detection and recognition of small objects.

Acknowledgments

This work had supported by the Natural Science Foundation of Shanghai of China (19ZR1455300).

References

[1] M. Liang, M. Yuan, X. Hu, J. Li and H. Liu, "Traffic sign detection by ROI extraction and histogram features-based recognition," The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, 2013, pp. 1-8, doi: 10.1109/IJCNN.2013.6706810.

[2] Y. Xu, Q. Wang, Z. Wei and S. Ma, "Traffic sign recognition based on weighted ELM and AdaBoost," in Electronics Letters, vol. 52, no. 24, pp. 1988-1990, 24 11 2016, doi: 10.1049/el.2016.2299.

[3] G. Huang, Q. Zhu and Siew Z C K. Extreme learning machine: Theory and applications [J]. Neurocomputing, 2006.

[4] D. Yasmina, R. Karima and A. Ouahiba, "Traffic signs recognition with deep learning," 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, 2018, pp. 1-5, doi: 10.1109/ICASS.2018.8652024.

[5] Redmon J , Divvala S , Girshick R , et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. IEEE, 2016.

[6] Redmon J , Farhadi A . YOLO9000: Better, Faster, Stronger[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2017:6517-6525.

[7] Redmon J , Farhadi A . YOLOv3: An Incremental Improvement[J]. arXiv e-prints, 2018.

[8] Bochkovskiy A , Wang C Y , Liao H Y M . YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. 2020.

[9] Girshick R , Donahue J , Darrell T , et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. 2013.

[10] He K , Zhang X , Ren S , et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904-16.
[11] Girshick R . Fast R-CNN[J]. Computer ence, 2015.
[12] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
[13] Fan Q , Zhuo W , Tang C K , et al. Few-Shot Object Detection With Attention-RPN and Multi-Relation Detector[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
[14] Wu Y , Chen Y , Yuan L , et al. Rethinking Classification and Localization for Object Detection[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.