 CMC, vol.60, no.1, pp.147-161, 2019

# Traffic Sign Recognition Method Integrating Multi-Layer Features and Kernel Extreme Learning Machine Classifier

**Wei Sun[1, 3, *], Hongji Du[1], Shoubai Nie[2, 3] and Xiaozheng He[4]**

**Abstract:** Traffic sign recognition (TSR), as a critical task to automated driving and driver assistance systems, is challenging due to the color fading, motion blur, and occlusion. Traditional methods based on convolutional neural network (CNN) only use an end-layer feature as the input to TSR that requires massive data for network training. The computation-intensive network training process results in an inaccurate or delayed classification. Thereby, the current state-of-the-art methods have limited applications. This paper proposes a new TSR method integrating multi-layer feature and kernel extreme learning machine (ELM) classifier. The proposed method applies CNN to extract the multi-layer features of traffic signs, which can present sufficient details and semantically abstract information of multi-layer feature maps. The extraction of multi-scale features of traffic signs is effective against object scale variation by applying a new multi-scale pooling operation. Further, the extracted features are combined into a multi-scale multi-attribute vector, which can enhance the feature presentation ability for TSR. To efficiently handle nonlinear sampling problems in TSR, the kernel ELM classifier is adopted for efficient TSR. The kernel ELM has a more powerful function approximation capability, which can achieve an optimal and generalized solution for multiclass TSR. Experimental results demonstrate that the proposed method can improve the recognition accuracy, efficiency, and adaptivity to complex travel environments in TSR.

**Keywords:** Traffic sign recognition, multi-layer features, multi-scale pooling, kernel extreme learning machine.

## 1 Introduction

Traffic sign recognition (TSR) has been applied in transportation practice to provide guidance information to drivers or vehicles to ensure the traffic safety and mobility. With the advances in artificial intelligence and machine learning, TSR methods are integrated

[1] School of Information and Control, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

[2] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

[3] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing, 210044, China.

[4] Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, New York, 12180, USA.

[*] Corresponding Author: Wei Sun. Email: sunw0125@163.com.

into driver assistance systems and intelligent driverless vehicles as an indispensable component [Yang, Long, Sangaiah et al. (2018)]. Though many TSR algorithms are available, their limited computational efficiency prevents them from real-time applications due to the complex travel environment including various lighting and weather conditions.

In a TSR method, feature extraction and traffic sign classification are two critical steps. In the feature extraction step, traditional methods adopt hand-crafted feature descriptors, including Histogram of Orientated Gradient (HOG), Scale-Invariant Feature Transform (SIFT) and Local Binary Pattern (LBP). For example, Wang et al. [Wang, Ren, Wu et al. (2013)] extracted the HOG features of five classes traffic signs and obtained better classification results. However, the generalization ability of the method is not strong enough for practical applications because it requires prior knowledge for extracting object characteristics. Zhu et al. [Zhu, Wang, Yao et al. (2013)] combined HOG, SIFT and LBP features to characterize the traffic sign, however, the dimension of the combined features is too high to conduct a real-time recognition.

At present, convolutional neural networks (CNN) based feature extraction methods are popular because of the efficiency of deep leaning. Compared to traditional hand-crafted features, CNN can automatically learn more targeted features based on the classification type. Furthermore, CNN works well by simulating perceptual processing of human visual cortex, thus can learn discriminative features more robustly. Jin et al. [Jin, Fu, and Zhang (2014)] proposed a CNN-based TSR method and suggested a hinge loss stochastic gradient descent method to train CNNs, which offered a faster and more stable convergence. Zeng et al. [Zeng, Xu, Fang et al. (2015)] proposed a CNN-based TSR method, in which only end-layer features are used in the classifier to recognize the traffic signs. Their proposed method achieved higher recognition rate than the methods based on hand-crafted features. A further study in Zeiler et al. [Zeiler and Fergus (2014)] found that the feature maps generated in different layers in CNNs correspond to different features of the object. For example, the features in shallow layers can express the object details such as texture and edge. However, the features in deep layers express the semantically abstract information. If we combine multiple features in different layers from CNNs, a higher accuracy of TSR would be achievable.

For object recognition, support vector machine (SVM) and neural networks have good performance in classifier design [Maldonado, Lafuente, Gil et al. (2007)]. In general, SVMs have a low recognition accuracy when high-dimension features are used. Ciresan et al. [Cireşan, Meier, Masci et al. (2012)] designed a classifier using a deep network for TSR and gained a high recognition accuracy (99.46%). However, the adopted gradient descent method must adjust many parameters in the training process, leading to long training time. In addition to the SVM and neural network classifiers, extreme learning machine (ELM) is another promising classifier. It adjusts less number of parameters in the training process. Huang et al. [Huang, Yu, Gu et al. (2017)] demonstrated that the ELM classifier can provide more accurate recognition results promptly for TSR. Other studies [Aziz, Mohamed and Youssef (2018); Zeng, Xu, Shen et al. (2017)] also demonstrated the kernel ELM has a strong generalization ability and more powerful function approximation capability.

To improve the efficiency of TSR, this paper proposes a new method based on multi-layer features and kernel extreme learning machine (MLF-KELM). The main contributions of our study are as follows. First, we extract multi-layer features using the CNN, including shallow and deep layer features that can express the local details and semantic information of traffic signs. Such multi-attribute features can improve representation ability for classifying traffic signs. Second, a multi-scale pooling operation is proposed to extract multi-scale features of traffic signs, leading to the improved robustness against scale variation. Finally, unlike traditional classifiers that only used a single end-layer feature, we propose to combine the extracted multi-attribute features in the KELM-based classifier for an accurate and robust classification. The proposed KELM-based classifier has fast convergence speed conveying a low computation complexity to practical applications.

The rest of this paper is organized as follows. Section 2 presents our proposed MLF-KELM architecture, which contains feature extraction via CNN (Section 2.1), multi-layer features fusion (Section 2.2), and kernel ELM-based classifier (Section 2.3). Section 3 shows the experimental results and performance analysis. Section 4 provides concluding remarks and future research directions.

## 2 MLF-KELM architecture

The study [Zeiler and Fergus (2014)] showed the feature maps in different layers corresponding to different feature attributes of the recognition object. If we can use the CNN multi-layer feature attributes, the recognition accuracy can be improved. Also, we concern the training time of TSR method. The ELM algorithm requires fewer parameters to be adjusted in training, leading to reduced training time [Huang, Zhu and Siew (2006)]. Furthermore, the kernel ELM has a powerful function approximation ability, which handles nonlinear problems efficiently. Therefore, we construct an MLF-KELM model that integrates the CNN multi-layer features with kernel ELM classifier for better recognition efficiency and accuracy.

Fig. 1 describes the proposed MLF-KELM architecture, which includes three successive steps: (1) Feature extraction via CNN, (2) Multi-layer features fusion, and (3) Kernel ELM-based classifier. The following three subsections describe the detailed steps, respectively.

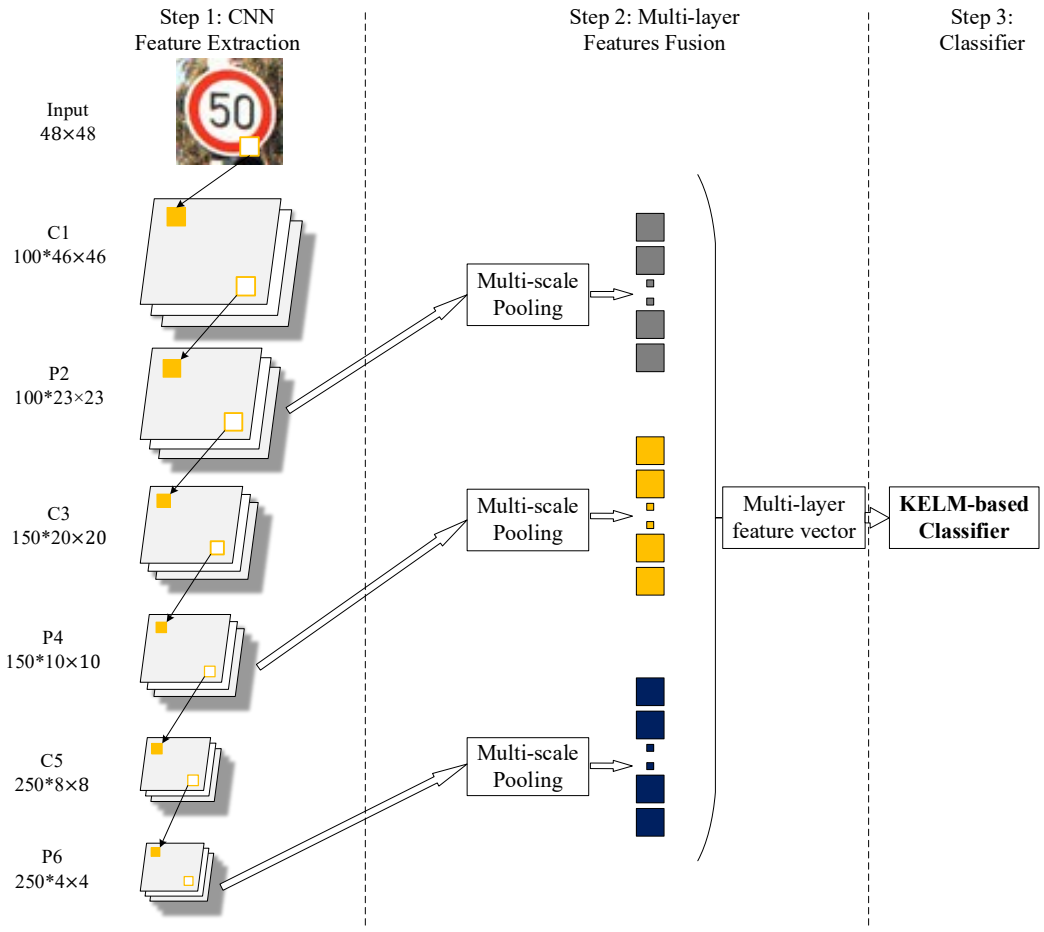


**Figure 1:** The proposed MLF-KELM architecture

### *2.1 Feature extraction via CNN*

CNN is a hierarchical learning model that can automatically extract deep features in images through alternating convolutional and max-pooling layers. The CNN structure constructed in this study consists of three convolutional layers, three max-pooling layers, and one fully-connected layer. As summarized in Tab. 1, notations C1, C3, and C5 represent convolutional layers with 100 maps, 150 maps and 250 maps defined by different sizes convolution kernels. In our architecture, convolution kernels and bias are randomly produced, and the activation function is the hyperbolic tangent function:

$$f(x) = \tanh(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x}) \tag{1}$$

The P2, P4, and P6 layers following each convolution layer are max-pooling layers, which reduce the computational complexity in training while remaining invariance. We refer to Cireşan et al. [Cireşan, Meier, Masci et al. (2011)] to build up the CNN

architecture, which achieved highly competitive performance. Tab. 1 shows the whole parameters setting of the CNN architecture.

**Table 1:** CNN architecture parameters

| | Number of maps and neurons | Kernel size | Stride |
|---|---|---|---|
| Input | 1 map of 48×48 neurons | — | — |
| C1 | 100 maps of 46×46 neurons | 3×3 | 1 |
| P2 | 100 maps of 23×23 neurons | 2×2 | 2 |
| C3 | 150 maps of 20×20 neurons | 4×4 | 1 |
| P4 | 150 maps of 10×10 neurons | 2×2 | 2 |
| C5 | 250 maps of 8×8 neurons | 3×3 | 1 |
| P6 | 250 maps of 4×4 neurons | 2×2 | 2 |
| F7 | 43 neurons | — | — |

This study regards the CNN structure as a feature extractor. We propose to combine the extracted features as an input to the classifier in the next section.

### *2.2 Multi-layer features fusion*

Following convolution and max-pooling operations, the traditional CNN maps the end-layer feature to a one-dimension vector, which is an input of the Softmax classifier. The back propagation algorithm is used in the end to train the network. This network only considers the information of the end-layer feature map. This section proposes a network framework with multi-feature expression. We use multi-layer feature maps to form a feature vector with multi-attributes to express traffic signs.

Li et al. [Li, Jiang, Pang et al. (2017)] extracted the last three layers features maps of hepatocellular carcinoma nuclei with CNN and obtained the ideal identification characteristics. They proved that the multiple-layer features maps contain the different identifying attributes and characteristic information of the object. Inspired by this observation, we extract three layers (P2, P4, and P6) features to express the multiple attributes of traffic signs. First, the P2, P4, and P6 layers' feature maps are extracted in the feedforward training. Then, the multi-scale features are extracted using multi-scale pooling. Last, we combine the extracted features to form a multi-scale multi-attribute feature vector of traffic signs.

Multi-scale pooling is an improved method based on spatial pyramid pooling. The spatial pyramid has a multi-scale hierarchical structure, which retains the spatial information in the local space blocks combining [He, Zhang, Ren et al. (2014)]. These multi-scale features following multi-scale pooling can adapt to the scale change of the object. Furthermore, the multi-scale pooling can enhance the invariance of the extracted features with CNN, thus improving the accuracy and robustness of object recognition [Gong, Wang, Guo et al. (2014)]. Fig. 2 illustrates the process of multi-scale pooling in detail. We adopt multiple sampling sizes and sampling strides for multi-scale pooling. Regardless of the size of the feature maps, each feature map outputs three feature

matrices of different scales after multi-scale pooling, i.e. $1\times1\times m$, $2\times2\times m$ and $3\times3\times m$, where $m$ represents the number of feature maps. Then, these three feature matrices in each feature map are concatenated into a column vector, i.e. $(14\times m)\times1$. In particular, the P2-layer, P4-layer, and P6-layer can form three column vectors respectively after multi-scale pooling, i.e. $1400\times1$, $2100\times1$ and $3500\times1$. Finally, we combine these three column vectors into a multi-scale multi-attribute traffic sign feature vector.

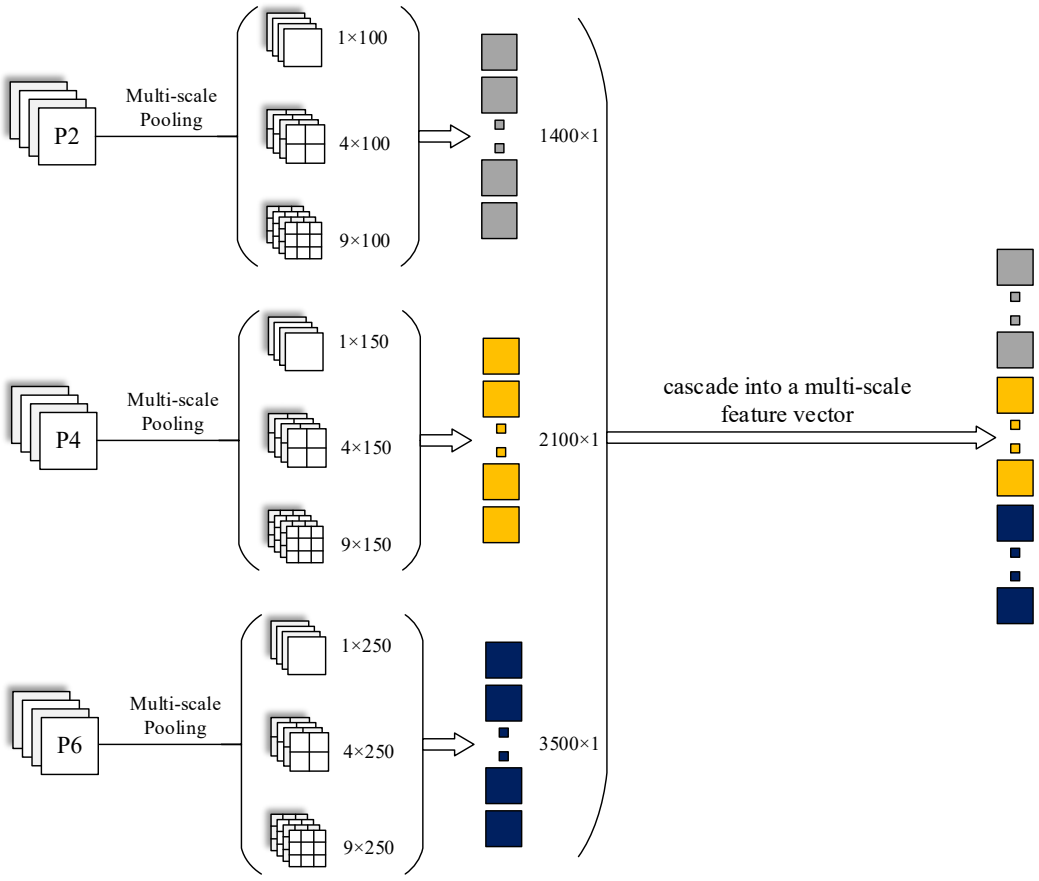


**Figure 2:** Multi-scale pooling

Multi-scale pooling is an improved method based on spatial pyramid pooling. The spatial pyramid has a multi-scale hierarchical structure, which retains the spatial information in the local space blocks combining [He, Zhang, Ren et al. (2014)]. These multi-scale features following multi-scale pooling can adapt to the scale change of the detection object. Furthermore, the multi-scale pooling can enhance the invariance of the extracted features with CNN, thus improving the accuracy and robustness of object recognition [Gong, Wang, Guo et al. (2014)]. Fig. 2 illustrates the process of multi-scale pooling in detail. We adopt multiple sampling sizes and sampling strides for multi-scale pooling. Regardless of the size of the feature maps, each feature map outputs three feature matrices of different scales after multi-scale pooling, i.e., $1\times1\times m$, $2\times2\times m$ and $3\times3\times m$, where $m$ represents the number

of feature maps. Then, these three feature matrices in each feature map are concatenated into a column vector, i.e., $(14 \times m) \times 1$. Specifically, the P2-layer, P4-layer and P6-layer can form three column vectors respectively after multi-scale pooling, i.e., $1400 \times 1$, $2100 \times 1$ and $3500 \times 1$. Finally, we combine these three column vectors into a multi-scale multi-attribute traffic sign feature vector.

### *2.3 Kernel ELM-based classifier*

After combining the multi-layer features, the dimension of the features is higher than the single-layer features. Therefore, it is necessary to find a classifier with computational simplicity, short training time, and high computational efficiency. ELM is a promising supervised algorithm based on the single hidden layer feedforward neural network that has two advantages. First, the weights between the input and hidden layers and biases in the hidden layers are randomly assigned. Compared to conventional learning techniques that adjust many parameters in training, the ELM algorithm can reduce the computational complexity, and training time is reduced significantly. The second advantage of ELM is its strong generalization ability. Furthermore, because the strong mapping ability of the kernel function, the ELM with the kernel function has a more powerful function approximation capability, which handles nonlinear problems efficiently. Due to these advantages, we apply the kernel ELM-based classifier to solve the multiclass recognition problems in TSR.

#### *2.3.1 Structure of ELM-based classifier*

The structure of the ELM-based classifier is shown in Fig. 3.

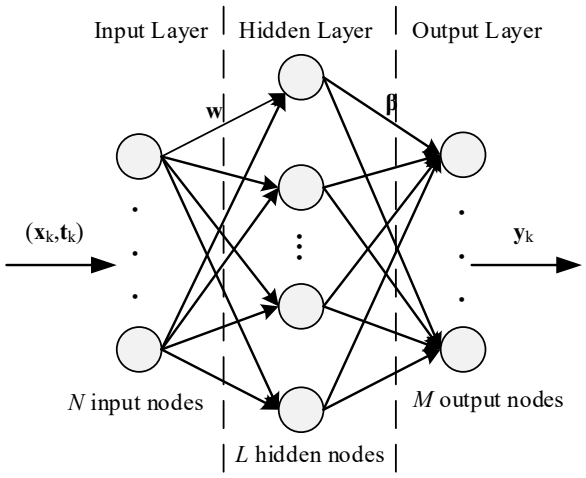


**Figure 3:** ELM structure

From Fig. 3, in the input layer of ELM, the traffic sign multi-layer features vector $\mathbf{x}$ is inputted. The dimension of $\mathbf{x}$ is denoted by $P$. In the hidden layer of ELM, the number of hidden nodes is denoted by $L$. The output of the $i$th ($i = 1, 2, \cdots L$) hidden node is represented by

$$g(\mathbf{x};\mathbf{w}_i,b_i)=g(\mathbf{x}\bullet\mathbf{w}_i+b_i) \tag{2}$$

where $g$ is a nonlinear piecewise continuous activation function. Our method in this study applies the sigmoid function to represent $g$. Here, $\mathbf{w}_i$ is the random weight vector that connects the $i$th hidden node with the input vector, and $b_i$ is the random bias of the $i$th hidden node. For a given multi-layer feature vector $\mathbf{x}$, its mapped feature vector can be expressed as

$$\mathbf{h}(\mathbf{x})=[g(\mathbf{x};\mathbf{w}_1,b_1),\ldots,g(\mathbf{x};\mathbf{w}_L,b_L)] \tag{3}$$

In the output layer of ELM, the number of output nodes is denoted by $M$. In this paper, each output node corresponds to a traffic sign class. The function below calculates the value of the $j$th ( $j=1,2\ldots,M$ ) output node,

$$f_j(\mathbf{x})=\sum_{i=1}^{L}\beta_{i,j}\times g(\mathbf{x};\mathbf{w}_i,b_i) \tag{4}$$

where $\beta_{i,j}$ is the output weight between the $i$th hidden node and the $j$th output node. Thus, the output vector in the output layer is expressed as

$$\mathbf{f}(\mathbf{x})=[f_1(\mathbf{x}),\ldots,f_M(\mathbf{x})]=\mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{5}$$

where

$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1\\ \vdots\\ \boldsymbol{\beta}_L\end{bmatrix}=\begin{bmatrix}\beta_{1,1} & \cdots & \beta_{1,M}\\ \vdots & \ddots & \vdots\\ \beta_{L,1} & \cdots & \beta_{L,M}\end{bmatrix} \tag{6}$$

In the recognition process, for a traffic sign multi-layer feature vector $\mathbf{x}$ of the test sample, its class label of $\mathbf{x}$ can be determined as

$$\text{label}(\mathbf{x})=\arg_{j=1,\ldots,M}\max f_j(\mathbf{x}) \tag{7}$$

*2.3.2 Training the ELM-based classifier*

The training process of ELM focuses on the three parameters: the input weights $\mathbf{w}$, biases $b$, and the output weights $\boldsymbol{\beta}$. Because the input weights and biases are randomly assigned, we only need to train the output weights $\boldsymbol{\beta}$.

For the given $N$ training samples $(\mathbf{x}_k,\mathbf{t}_k)$, $\mathbf{x}_k$ is the traffic sign feature vector and $\mathbf{t}_k$ is the binary traffic sign class label for $\mathbf{x}_k$, $k=1,...,N$. Taking the training samples into (4) yields a linear representation:

$$\mathbf{H}\boldsymbol{\beta}=\mathbf{Y} \tag{8}$$

where $\mathbf{Y}$ is the output vector,

$$\mathbf{Y}=\begin{bmatrix}\mathbf{y}_1\\ \vdots\\ \mathbf{y}_N\end{bmatrix}=\begin{bmatrix}y_{1,1} & \cdots & y_{1,M}\\ \vdots & \ddots & \vdots\\ y_{N,1} & \cdots & y_{N,M}\end{bmatrix} \tag{9}$$

and $\mathbf{H}$ is the output vector in the hidden layer:

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}(\mathbf{x}_1)\\ \vdots \\ \mathbf{h}(\mathbf{x}_N)\end{bmatrix}=\begin{bmatrix}g(\mathbf{x}_1;\mathbf{w}_1,b_1) & \cdots & g(\mathbf{x}_1;\mathbf{w}_L,b_L)\\ \vdots & \ddots & \vdots \\ g(\mathbf{x}_N;\mathbf{w}_1,b_1) & \cdots & g(\mathbf{x}_N;\mathbf{w}_L,b_L)\end{bmatrix} \tag{10}$$

The training process aims to minimize the training error $\|\mathbf{T}-\mathbf{H}\boldsymbol{\beta}\|^2$ and the Euclidian norm of output weight $\|\boldsymbol{\beta}\|$. The resulting training process can be represented as a constrained optimization problem:

$$\begin{aligned}&\min:\psi(\boldsymbol{\beta},\boldsymbol{\xi})=\frac{1}{2}\|\boldsymbol{\beta}\|^2+\frac{1}{2}C\|\boldsymbol{\xi}\|^2\\ &\text{st}:\ \mathbf{H}\boldsymbol{\beta}=\mathbf{T}-\boldsymbol{\xi}\end{aligned} \tag{11}$$

where $C$ is the regularization coefficient, $\boldsymbol{\xi}$ denotes the tolerance parameter, which can enhance the generalization ability of the model.

This paper applies Lagrange multiplier method [Huang, Zhou, Ding et al. (2012)] to solve the constrained optimization problem:

$$\boldsymbol{\beta}=\mathbf{H}^T(\mathbf{I}/C+\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{12}$$

*2.3.3 Kernel ELM-based classifier*

In fact, as the form of the feature mapping function $\mathbf{h}(\mathbf{x})$ is unknown, the kernel technique can be introduced into the ELM based on Mercer's condition [Huang, Ding, and Zhou (2010)]. Namely, we can replace $\mathbf{H}\mathbf{H}^T$ with a kernel function. In this paper, we propose to use the RBF function as the kernel $K(\mathbf{x}_i,\mathbf{x}_j)$.

$$\mathbf{H}\mathbf{H}^T(i,j)=K(\mathbf{x}_i,\mathbf{x}_j) \tag{13}$$

$$\mathbf{H}\mathbf{H}^T=\Omega_{\text{ELM}}=\begin{bmatrix}K(\mathbf{x}_1,\mathbf{x}_1) & \cdots & K(\mathbf{x}_1,\mathbf{x}_N)\\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_N,\mathbf{x}_1) & \cdots & K(\mathbf{x}_N,\mathbf{x}_N)\end{bmatrix} \tag{14}$$

Substituting Eq. (14) into Eq. (12) yields the output vector $\mathbf{f}(\mathbf{x})$ of the kernel ELM:

$$\mathbf{f}(\mathbf{x})=\mathbf{h}(\mathbf{x})\boldsymbol{\beta}=\mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C}+\mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}=\begin{bmatrix}K(\mathbf{x},\mathbf{x}_1)\\ \vdots \\ K(\mathbf{x},\mathbf{x}_k)\end{bmatrix}\left(\frac{\mathbf{I}}{C}+\Omega_{\text{ELM}}\right)^{-1}\mathbf{T} \tag{15}$$

The kernel ELM improves the function approximation capability, which can handle nonlinear problems efficiently. Therefore, the kernel ELM classifier can achieve an optimal and generalized solution for multiclass TSR.

## 3 Experiments and analysis

### *3.1 Experimental setup*

For experimentation purpose, we normalize the size of input images because the CNN only accepts images with uniform size. The normalization algorithms include the nearest interpolation, bilinear interpolation, and cubic convolution interpolation. Considering the normalized image quality and the computational complexity during interpolation, we scale the image to the fixed size of $48 \times 48$ pixels using bilinear interpolation.

To reduce the computational complexity, our experiments only utilize the gray scale images calculated through the following formula,

$$Gray = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{16}$$

where $R$, $G$ and $B$ represent red, green and blue channels of each traffic sign image respectively. Note that the proposed method can handle color images with additional computation time and training time.

We verify the proposed algorithm using the German Traffic Sign Recognition Benchmark (GTSRB) [Stallkamp, Schlipsing, Salmen et al. (2011)] as the experiment dataset that contains 43 classes of traffic signs. The GTSRB dataset contains 51,839 images in total captured from the real world travel environment. The dataset has been divided into 39,209 training images and 12,630 test images. The images sizes are not same, varying from 15×15 pixels to 250×250 pixels. All experiments are conducted on a PC with 3 GHz i7 CPU 16 GB RAM and a NVIDIA GTX1060 graphics card with 6 GB ram. The CNN training process terminates once the default training epoch (e.g. 50 iterations) is reached. Initial weights of the CNN are drawn from a uniform random distribution in the range $[-0.01, 0.01]$.

We randomly select an image of the traffic sign and input it into the trained network to illustrate the multi-layer features. The feature maps of each network layer are shown in Fig. 4. We find that different feature maps have different characteristics in terms of textures, corners, and edges. Moreover, the features in deep layers are more abstract through alternating convolution and pooling. The hierarchical and distinguishable features are similar to the human visual system.

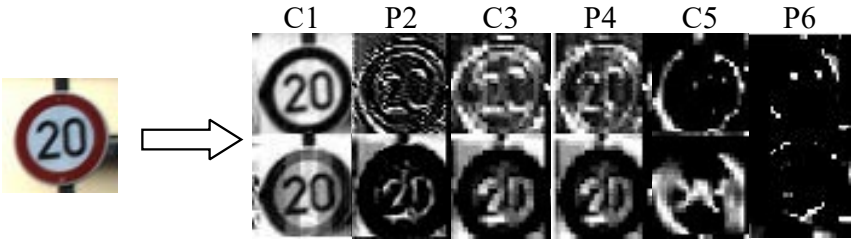


**Figure 4:** Feature map visualization

### *3.2 Comparison of multi-layer feature and single-layer feature*

To verify the efficacy of multi-layer features for TSR, we conduct the experiment to compare the average accuracy of multi-layer feature (MLF) against single-layer feature (SLF) based on the ELM-based classifier. The experiment adopts the end-layer feature of CNN as the SLF.

The number of hidden layer nodes (i.e., $L$) influences the accuracy of classification significantly. Fig. 5 shows the average recognition accuracy of different numbers of hidden nodes. The accuracy of the multi-layer feature is higher than that of the single-layer feature, about two percent on average.

Fig. 5 also illustrates that the average recognition accuracy increases with the number of hidden nodes $L$. After $L$>8000, the increase of accuracy slows down. To balance recognition accuracy and computational complexity, we choose the number of hidden nodes $L$=8000 in all experiments. As shown in Fig. 5, the average accuracy of the single-layer feature is 95.43%, and the average accuracy of the multi-layer feature is 97.54%.

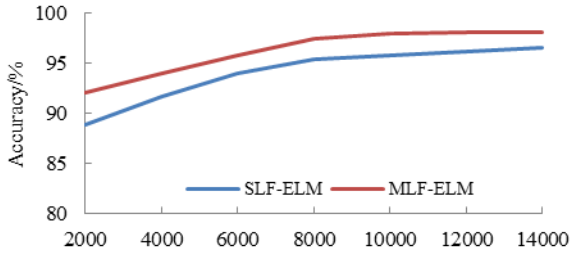


**Figure 5:** Recognition accuracy of different numbers of hidden nodes $L$

Since the single-layer features do not adequately represent the comprehensiveness and multi-attribute characteristics of the traffic sign, the recognition accuracy is not high enough for real-time applications. The multi-layer features make full use of the diversity of features and improve the recognition accuracy significantly.

We use traffic sign images captured in complex travel environments to verify the robustness of the proposed method. Fig. 6 shows examples of traffic sign images captured under three categorized conditions: light deficiency, partly occluded, and motion blur. For each category, we use 100 images as a testing subset. Tab. 2 shows the average recognition accuracy of multi-layer feature and single-layer feature using images captured under the complex travel environments. The accuracy of the multi-layer feature is still higher than that of the single-layer feature under the complex travel environments, mainly because of the comprehensiveness of the multi-layer feature. Compared to the experiment results in Fig. 5, the average accuracy reduces, especially in the case of partial occlusion. Nevertheless, our method can obtain a good performance. The recognition accuracy under most conditions is above 95%. It demonstrates that our proposed method is robust to various travel environmental conditions.

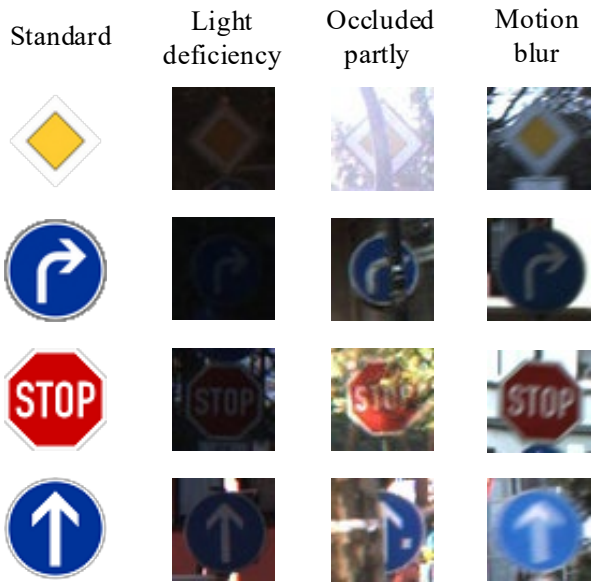


**Figure 6:** Examples of traffic sign images captured under complex travel environments

**Table 2:** Recognition accuracy under the complex travel environments

| Method | Accuracy | | |
|---|---|---|---|
| | Light deficiency | Partly occluded | Motion blur |
| SLF-ELM | 94.43% | 93.08% | 93.91% |
| **MLF-ELM** | **96.68%** | **95.24%** | **96.01%** |

### *3.3 Comparison of classifiers*

We verify the advantage of kernel ELM-based classifier in our proposed method by conducting experiments using kernel ELM-based classifier, ELM-based classifier, Softmax classifier, and SVM classifier, with the same multi-layer features. The comparison applies three evaluation metrics: the average recognition accuracy, training time, and recognition time. Tab. 3 summarizes the comparison results.

**Table 3:** Results of different classifier

| Classifier | Recognition accuracy | Training time | Recognition time |
|---|---|---|---|
| Softmax | 96.53% | 44.8 min | 16.5 ms/frame |
| SVM | 96.87% | 32.9 min | 39.2 ms/frame |
| ELM-based | 97.54% | 6.4 min | 7.68 ms/frame |
| **Kernel ELM-based** | **97.93%** | **3.8 min** | **5.83 ms/frame** |

Tab. 3 highlights that the average accuracy of kernel ELM-based classifier is superior to all other three classifiers with an accuracy of 97.93%. As the kernel ELM classifier has a

powerful function approximation capability, its training time is much less than other classifiers. Furthermore, the kernel ELM-based classifier also has advantages in recognition time with an average value of 5.83 ms per frame. The short recognition time can satisfy the computational requirement of practical applications.

### *3.4 Verification in other dataset*

This study verifies the transferability and the generalization ability of the model trained by GTSRB by using a new dataset to an experiment. The datasets from Laboratory for Intelligent & Safe Automobiles (LISA) [Mogelmose, Trivedi and Moeslund (2012)] and Belgium Traffic Sign Classification Benchmark (BTSCB) [Mathias, Timofte, Benenson et al. (2013)] are combined as a new dataset. Since LISA contains 47 traffic sign classes, and BTSCB contains 62 traffic sign classes, and our proposed model only has 43 output nodes. Therefore, we only choose 43 traffic sign classes from LISA and BTSCB to be the new testing dataset. In total, there are 200 images (100 images in each dataset) in the new dataset. The experimental results show that the average accuracy is 96.41%, and the average recognition time of each image is 9.15 ms, which demonstrate a consistent performance and the transferability of the proposed method.

## 4 Conclusions

This paper has proposed a novel MLF-KELM architecture for TSR. In the feature extraction step, we use the CNN to extract deep features of images and multi-layer features to express traffic signs. In classification step, we adopt the kernel ELM-based classifier to categorize traffic signs. To our knowledge, this is the first time when an MLF-KELM classifier is used for TSR. TSR experiments on the GTSRB dataset show that the proposed method exhibits promising performance and a strong generalization in a much shorter period of training.

Future research will focus on two aspects. First, we plan to embed our proposed method in a more general system that first localizes traffic signs in realistic scenes and then classifies them. Second, we will apply the deep reinforcement learning with visual attention to TSR to find key areas of traffic signs, which can improve the recognition accuracy efficiently.

**Acknowledgement:** This work is supported in part by the National Nature Science Foundation of China (No. 61304205, 61502240), Natural Science Foundation of Jiangsu Province (BK20141002), and Innovation and Entrepreneurship Training Project of College Students (No. 201710300051, 201710300050).

## References

**Aziz, S.; Mohamed, E. L.; Youssef, F.** (2018): Traffic sign recognition based on multi-feature fusion and elm classifier. *Procedia Computer Science*, vol. 127, pp. 146-153.

**Cireşan, D.; Meier, U.; Masci, J.; Schmidhuber, J.** (2011): A committee of neural networks for traffic sign classification. *International Joint Conference on Neural Networks*, pp. 1918-1921.

**Cireşan, D.; Meier, U.; Masci, J.; Schmidhuber, J.** (2012): Multi-column deep neural network for traffic sign classification. *Neural Networks*, vol. 32, pp. 333-338.

**Gong, Y.; Wang, L.; Guo, R.; Lazebnik, S.** (2014): Multi-scale orderless pooling of deep convolutional activation features. *European Conference on Computer Vision*, pp. 392-407.

**He, K.; Zhang, X.; Ren, S.; Sun, J.** (2014): Spatial pyramid pooling in deep convolutional networks for visual recognition. *European Conference on Computer Vision*, pp. 346-361.

**Huang, G.; Ding, X.; Zhou, H.** (2010): Optimization method based extreme learning machine for classification. *Neurocomputing*, vol. 74, no. 1-3, pp. 155-163.

**Huang, G.; Zhou, X.; Ding, X.; Zhang, R.** (2012): Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 513-529.

**Huang, G.; Zhu, Q.; Siew, C.** (2006): Extreme learning machine: Theory and applications. *Neurocomputing*, vol. 70, no. 1, pp. 489-501.

**Huang, Z.; Yu, Y.; Gu, J.; Liu, H.** (2017): An efficient method for traffic sign recognition based on extreme learning machine. *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 920-933.

**Jin, J.; Fu, K.; Zhang, C.** (2014): Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1991-2000.

**Li, S.; Jiang, H.; Pang, W.** (2017): Joint multiple fully connected convolutional neural network with extreme learning machine for hepatocellular carcinoma nuclei grading. *Computers in Biology and Medicine*, vol. 84, pp. 156-167.

**Maldonado, B. S.; Lafuente, A. S.; Gil, J. P.; Gomez, M. H.; Lopez, F. F.** (2007): Road-sign detection and recognition based on support vector machines. *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 264-278.

**Mathias, M.; Timofte, R.; Benenson, R.; Gool, L. V.** (2013): Traffic sign recognition-How far are we from the solution? *International Joint Conference on Neural Networks*, pp. 1-8.

**Mogelmose, A.; Trivedi, M. M.; Moeslund, T. B.** (2012): Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484-1497.

**Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C.** (2011): The German traffic sign recognition benchmark: A multi-class classification competition. *International Joint Conference on Neural Networks*, pp. 1453-1460.

**Wang, G.; Ren, G.; Wu, Z.; Zhao, Z.; Jiang, L.** (2013): A hierarchical method for traffic sign classification with support vector machines. *International Joint Conference on Neural Networks*, pp. 1-6.

**Yang, T.; Long, X.; Sangaiah, A. K.; Zheng, Z.; Tong, C.** (2018): Deep detection network for real-life traffic sign in vehicular networks. *Computer Networks*, vol. 136, pp. 95-104.

**Zeiler, M.; Fergus, R.** (2014): Visualizing and understanding convolutional networks. *13th European Conference on Computer Vision*, pp. 818-833.

**Zeng, Y.; Xu, X.; Fang, Y.; Zhao, K.** (2015): Traffic sign recognition using deep convolutional networks and extreme learning machine. *5th International Conference on Intelligent Science and Big Data Engineering*, pp. 272-280.

**Zeng, Y.; Xu, X.; Shen, D.; Fang, Y.; Xiao, Z.** (2017): Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1647-1653.

**Zhu, Y.; Wang, X.; Yao, C.; Bai, X.** (2013): Traffic sign classification using two-layer image representation. *20th IEEE International Conference on Image Processing*, pp. 3755-3759.