Portable Convolution Neural Networks for Traffic Sign Recognition in Intelligent Transportation Systems

Junhao Zhou*, Hong-Ning Dai*, and Hao Wang†
*Macau University of Science and Technology, Macau SAR
junhao_zhou@qq.com; hndai@ieee.org
†Norwegian University of Science and Technology, Gjøvik, Norway
hawa@ntnu.no

***Abstract*—Deep convolutional neural networks (CNN) have the strength in traffic-sign classification in terms of high accuracy. However, CNN models usually contains multiple layers with a large number of parameters consequently leading to a large model size. The bulky model size of CNN models prevents them from the wide deployment in mobile and portable devices in Intelligent Transportation Systems. In this paper, we design and develop a portable convolutional neural network (namely portable CNN) structure used for traffic-sign classification. This portable CNN model contains a stacked convolutional structure consisting of factorization and compression modules. We conducted extensive experiments to evaluate the performance of the proposed Portable CNN model. Experimental results show that our model has the advantages of smaller model size while maintaining high classification accuracy, compared with conventional CNN models.**

***Keywords*—Convolutional neural networks; Portable; Factorization; Model Compression; Intelligent Transportation Systems**

I. Introduction

Traffic-sign recognition plays an important role in developing Intelligent Transportation Systems (ITS) [10]. Traditional methods for traffic sign recognition are mainly based on machine learning algorithms, including support-vector-machine (SVM) classifiers with LIPID (local image permutation interval descriptor) [16] and sparse reprentations [9]. Moreover, it is shown in [6] and [12] that Multilayer perceptrons (MLP) perform high accuracy and achieve low false positive rates when identifying the characters in speed-limit signs in [1].

Recently, deep convolutional neural network (CNN) models show the advantages in learning complicated and hierarchical features of massive image data [8]. For example, the work of [4] proposes a Multi-column deep neural network (MCDNN) structure, which has superior performance than other machine learning models in German Traffic Sign Recognition Benchmark (GTSRB) [11]. Meanwhile, other deep CNN models such as VGG [14], GoogleNet [15], ResNet [5] also demonstrate the outstanding performance in image classification.

However, deep CNN models usually contains multiple layers with a large number of parameters. As a result, CNN models typically have a large model size. Moreover, they also require using strong processing devices (e.g., Graphics Processing Units) to train the models. In addition, the large model size of CNN models also results in the huge communication overhead in distributed CNN model-training [7]. Therefore, these drawbacks hinder the wide deployment of CNN models in mobile and portable devices in ITS, e.g., Portable Navigation Devices (PND) or Roadside Units (RSU). It is necessary to design a lightweight CNN model while maintaining high accuracy in traffic-sign classification.

In this paper, we design and develop a lightweight CNN model for traffic-sign classification. In particular, our model consists of a stacked convolutional structure which consists of factorization and compression layers. Our model has the merits including the smaller model size than conventional CNN models while maintaining high accuracy in traffic-sign classification. For example, the proposed Portable CNN model has model size of 5.8 MB, which are much smaller than other conventional CNN models. Meanwhile, the accuracy of the proposed model is 98.62% outperforming other models.

The main research contributions of this paper can be summarized as follows.

- We put forth a stacked convolutional structure consisting of factorization and compression modules. In particular, portable CNN model is mainly composed of three paths and a compression layer; three paths mainly refer to the simple convolution path, the factorization path and the short-cut path. Then, we concatenate the outputs from these paths to the compression layer. This design can significantly reduce the complexity of CNN model while maintaining high predication accuracy.
- We conduct extensive experiments based on a realistic traffic-sign dataset. We evaluate the performance of the proposed portable CNN model with other representative CNN models including MCDNN [4] model, VGG-16 [14] and AlexNet [8]. Our model outperforms the conventional models in terms of higher classification accuracy and smaller model size. In addition, we also conduct extensive experiments to investigate the impact of parameters on the proposed portable CNN model.

The remainder of this paper is organized as follows. Section II describes our model structure. Experimental results are presented in Section III. We conclude the paper in Section IV.

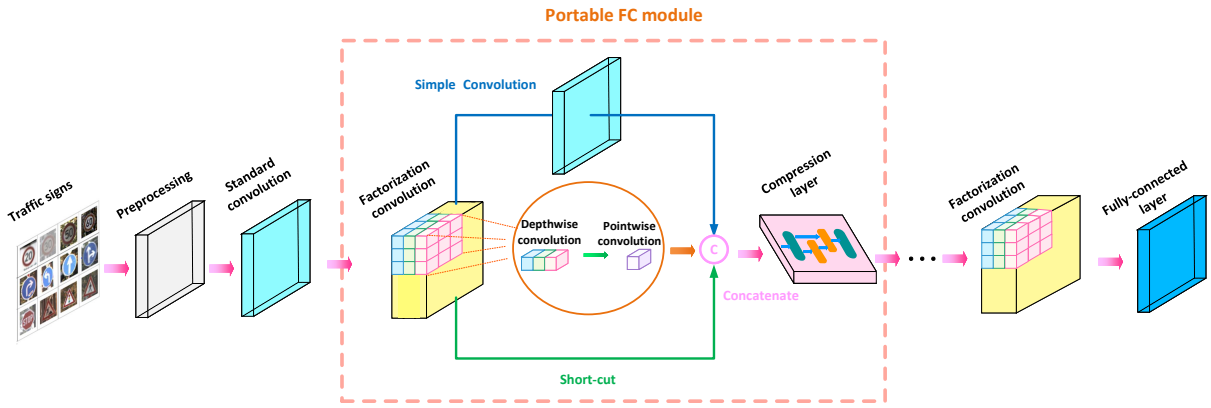


Fig. 1: Portable CNN Model consists of factorization convolution layers and compression layers.

II. Portable Convolutional Neural Netwworks

A. Overview of architecture

In this paper, we present a portable CNN model, which consists a major component: portable factorization and compression module (namely portable FCM) as shown in Fig. 1. The portable FCM consists of three paths concatenated with a compression layer where three paths mainly refer to the simple convolution path, the factorization path and the short-cut path.

We then briefly describe the working procedure of the portable CNN model.

1) **Preprocessing.** The recent study [2] shows that the class-imbalance problem in input data sets is detrimental to CNN models. Meanwhile, the traffic-sign data sets such as German Traffic Sign Recognition Benchmark (GTSRB) [11] often contain blur, distorted and blemished images, consequently affecting the performance of CNN models. Therefore, we adopt data oversampling and augmentation methods [3] to solve the class-imbalance problem and noisy data.
2) **Standard convolution.** We choose a standard convolution structure to process the traffic-sign images. The standard convolution structure consists of several convolutional layers, pooling layers and a fully-connected layer. The convolutional input is an $m \times m \times r$ image, where m denotes the height (and the width) of image and r denotes the number of channels (e.g., $r = 3$ in RGB model because of red, green and blue channels. Meanwhile, we choose b filters, each of which has a size of $n \times n \times q$ in the convolutional layer, where n is typically smaller than the dimension of the input image and q is equal to the channels.
3) **portable FCM.** We present a new building block called portable FCM. It is a key component in our portable CNN model. This network module is mainly composed of three paths concatenated with a compression layer, where the three paths refer to the simple convolution path, the factorization path and the short-cut path. The simple convolution path is similar to a common convolution layer in a typical CNN while the factorization consists of a depth-wise convolution and a point-wise convolution consequently reducing the complexity of models. The short-cut path can help to mitigate the gradient-loss problem [5]. Then, we concatenate the outputs from these paths with a compression layer.
4) **Factorization convolutional layer.** In this layer, the conventional convolution is decomposed into the depthwise convolution and the pointwise convolution. Moreover, we also optimize the convolution stride to reduce the computing cost.
5) **Fully-connected layer.** We next employ a fully-connected layer which consists a number of neurons to extract the main features of traffic signs. The calculation procedure is similar to that in the standard convolution layer. In particular, we denote the number of neurons by β, which is tuneable in our experiments.

B. Portable FCM

In this section, we introduce the portable FCM which is a core building block in our portable structure. We define the portable FCM as follows. A portable FCM is mainly comprised of three paths concatenated with a compression layer among which the simple convolution path can improve the performance of the model via standard convolution; the factorization path can reduce the computing cost and portable the model by factorization convolution; the short-cut path can make the features skipping some designated layers directly like a highway, consequently improving the effect of information transferring. Finally, the model concatenates the output of these three paths to the compression layer. Fig. 2 illustrates the working mechanism of the portable FCM. We next describe the technical details of these components as follows.

1) Simple convolution path: Simple convolution path is a channel connecting the input data to compression layer via a standard convolution. The simple convolutional filter size is 3 $\times$ 3 in our model. This path can improve the performance and the stability for our model.

2) Factorization path: Factorization path is a channel in order to build the portable model via factorization convolution. Unlike the standard convolutional layer, a factorization convolutional layer is decomposed into the depthwise convolution and the pointwise convolution via a factorization convolution in this layer. The depthwise convolution essentially factorizes the standard convolution into M depthwise convolutional filters, each with a size of $D_K \cdot D_K$. The pointwise convolution

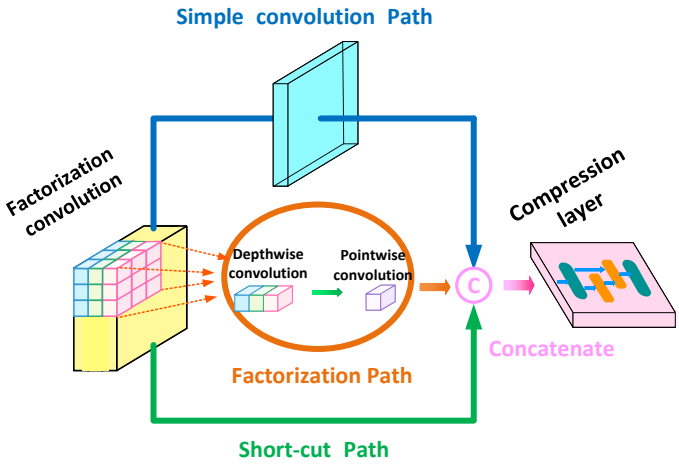


Fig. 2: Portable FCM consists of three paths concatenated with a compression layer

combines two outputs of depthwise convolution through N pointwise convolutional filters, each with a size of 1×1. This path can reduce the computing cost for our model. We calculate the computational cost of the factorization convolution and evaluate the cost reduction in contrast to the standard convolution operation.

3) Short-cut path: Short-cut path is equivalent to a green channel using a short-cut connection. The short-cut connection which can skip the designated training layers by mapping low-level feature directly to high-level feature; this manner is similar to ResNet [5]. Meanwhile, it can achieve deeper network and higher performance via solving the gradient loss problems.

4) Concatenate to compression layer: After obtaining three output data from three paths, we concatenate and compress them using compression layer. Compression layer employs a Concatenated Rectified Linear Unit (CReLU) proposed in [13] to design a compression convolution filter. Compared with conventional activation functions such as Rectified Linear Units (ReLU), CReLU can decrease the extra and unnecessary computational cost. Therefore, the compression layer can significantly overcome the drawbacks of ReLU with a simple but effective modification.

In particular, a negation operation and a concatenation operation are conducted before invoking the ReLU activation function in contrast to conventional ReLU activation function. Moreover, CReLU can also help to reduce the redundant filters. Specifically, we can compress the CNN model via halving the number of the filters by using CReLU compression layer.

III. Experiment

In this section, we conduct the experiments to evaluate the performance of the proposed portable CNN model. We first describe basic experimental settings in Section III-A. We then evaluate the performance of the proposed portable CNN model by comparing with conventional CNN models in Section III-B. Moreover, we also evaluate the impacts of parameters on the performance of the proposed portable CNN model in Section III-C.

Fig. 3: Examples from GTSRB Dataset

A. Experimental Settings

1) Dataset description.: We conduct our experiments on German Traffic Sign Recognition Benchmark (GTSRB) dataset [11], which has been widely used in evaluating classification algorithms in traffic sign recognition. GTSRB dataset contains more than 50,000 traffic sign images, which have been categorized into 40 classes. We select three major categories: *Speed-limit signs*, *Direction signs* and *Attention signs*. Fig. 3 shows some selected examples from each of the datasets. The number of traffic signs in each category is different from each other (i.e., the class-imbalance problem). Therefore, we first preprocess the dataset via the aforementioned oversampling and data augmentation. To simplify our discussion, we name the dataset containing Speed-limit signs as GTSRB-1, the dataset containing Direction signs as GTSRB-2, the dataset containing Attention signs as GTSRB-3 and the dataset containing all the three categories of traffic signs as GTSRB-T (GTSRB Total).

2) Comparison algorithms.: We evaluate the performance of the proposed portable CNN model with other conventional CNN models as described as follows.

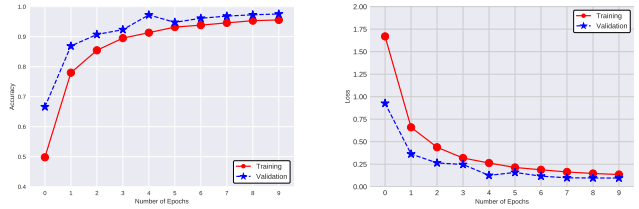
MCDNN [4] is a multi-layer CNN model used for GTSRB dataset and performed excellent (won the final phrase in the German traffic sign recognition benchmark with even better accuracy than human recognition in 2011). This model consists of 6 layers (i.e., 2 convolutional layers, 2 pooling layers and 2 fully-connected layers) .

AlexNet was proposed and developed by Krizhevsky, Sutskever and Hinton [8]. It consists of totally 8 layers: 5 convolutional layers and 3 fully-connected layers. The activation function is ReLU.

VGG-16 was proposed and developed by Simonyan and Zisserman [14]. This model significantly increases the number of layers in CNN architectures to 16 layers (the 19-layer version is named as VGG-19). It consists of 13 convolutional layers and 3 fully-connected layers.

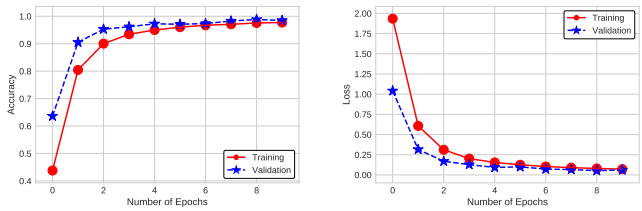
Factorization-Net is a CNN model with a single factorization convolutional layer. It can be regarded as a special case of our proposed portable CNN model without compression layer.

3) Performance metrics.: We conduct the experiments by considering two performance metrics: *classification accuracy* and *model size*. In particular, the classification accuracy is defined as the ratio of the number of correct classifications to the total number of classifications. To evaluate the model size, we mainly consider the total number of parameters of the



(a) Accuracy (b) Loss

Fig. 4: From left to right: Accuracy rates and loss rates of portable FCM were shown. Left: The accuracy rate of training set is 96.65% and validation set accuracy is 97.26% after running 10 times iterations on GTSRB-T dataset. Right: Loss of training set is 0.135 and that of validation set is 0.096 after 10 epochs.



(a) Accuracy (b) Loss

Fig. 5: From left to right: Accuracy rates and loss rates of Factorization-Net running on GTSRB-T are shown after 10 times iterations. Left: The accuracy rate of training set is 97.71% and validation set accuracy is 98.46% after 10 times iterations. The result is just little reduction with the accuracy of the baseline model [4]. Right: Loss rates mean the result of the model convergence. For Factorization-Net, the loss of training set is 0.0717 and that of validation set is 0.0603.

trained models and the file size of the trained models (in terms of MB).

B. Experimental results

Table I presents the performance comparison of our proposed portable CNN model with other conventional CNN models. It is worth noting that the experiments are conducted on four datasets: GTSRB-1, GTSRB-2, GTSRB-3 and GTSRB-T. In the experiments, we choose the number of the factorization convolutional layers to be $\alpha = 4$ and the number of of neurons in the fully connected layer to be $\beta = 256$. Factorization-Net has the same number of the factorization convolutional layers as our model.

Accuracy. It is shown in Table I that portable CNN model outperforms other existing models in all the four datasets (GTSRB-1, GTSRB-2, GTSRB-3, GTSRB-T). For example, the accuracy of portable CNN model in GTSRB-T is 98.62%, the highest accuracy among all the models even though M-CDNN and VGG-16 also achieve the close accuracy values. The performance improvement of the proposed portable CNN model may attribute to the excellent features of portable CNN model such as reducing the unnecessary and redundant parameters, which may affect the accurate classification on traffic signs.

Model size. Table I also gives the comparison on the model size between the proposed portable CNN model and other conventional models. It is shown in Table I that portable CNN model has much smaller model size than other models. For example, portable CNN model has the file size of 5.8 MB with 713,259 parameters, which is about $5.2\times$ smaller than AlexNet and $20.5\times$ smaller than VGG-16 with comparable classification score. portable CNN model has even smaller model size than MCDNN model, which has shallower structure than AlexNet and VGG-16.

C. Impacts of parameters

We then investigate the impacts of various parameters on the performance of portable CNN model.

TABLE II: Comparison with computational cost

Convolution type	Computational cost
Standard convolution	**2,359,296**
Factorization convolution	**335,872**

1) Effect of Portable FCM: We evaluate the impact of portable FCM on the entire structure. Table I shows that the portable FCM can reduce the model size while maintaining the high accuracy. For example, the model size can be reduced to 4.4 MB, even smaller than that of Factorization-Net model size (6.1 MB) while it maintains almost the highest accuracy like MCDNN model. Furthermore, Fig. 4 shows the accuracy rates and the loss rates.

2) Effect of factorization convolution: In CNN, computational cost is an important factor that influences the efficiency of the CNN structure. Next, We compare with the computational cost between a standard convolution in M-CDNN and a factorization convolution in Factorization-Net. For Factorization-Net (Table I), we factorize the standard convolution filter by factorization convolution which consists of depthwise convolution and pointwise convolution.

Compared with the computational cost of standard convolutions, it is $7\times$ cost reduction for factorization convolution. In addition, we can see the performance in Table I that after factorization convolution processing, Factorization-Net model size is 6.1 MB. Compared with the model size of conventional CNN structure, such as MCDNN [4] (19.7 MB), our factorization experiment reduced model size by $3.23\times$. In Figure 5, it describes a processing of Factorization-Net training. It shows that the perfomance is just little reduction comparing with MCDNN.

3) Effect of Portable CNN: We then investigate the impact of portable CNN. We evaluate three structures of Factorization-Net, portable FCM and portable CNN. The experiments were also conducted on data GTSRB-T only.

Table III presents the results. It is shown in Table III that portable FCM can reduce the redundant parameters while

TABLE I: Performance comparison with other conventional CNN models.

Models	Model Size	No. of Parameters	Accuracy (GTSRB-1)	Accuracy (GTSRB-2)	Accuracy (GTSRB-3)	Accuracy (GTSRB-T)
MCDNN	19.7 MB	2,466,507	97.95%	97.79%	97.21%	98.50%
AlexNet	30.2 MB	3,889,835	95.02%	96.60%	95.53%	96.31%
VGG-16	118.8 MB	15,291,499	96.52%	97.29%	97.70%	98.60%
Factorization-Net	6.1 MB	754,373	96.75%	95.61%	94.95%	97.71%
portable CNN	**5.8 MB**	**713,259**	**97.09%**	**97.33%**	**97.32%**	**98.62%**

TABLE III: Evaluation with structures

Model	Accuracy (GTSRB-T)	Model Size	No. of Parameters
Factorization-Net	97.71%	6.1 MB	754,373
portable FCM	96.65%	4.4 MB	540,843
portable CNN	98.62%	5.8 MB	713,259

maintaining high classification accuracy. Furthermore, our proposed portable CNN model outperforms other models in terms of highest accuracy after combining factorization convolutional layers. This result implies that the proposed CNN is highly portable and it may be used in mobile scenarios.

4) Effect of Number of Neurons in Fully-Connected Layer : We also investigate the impact of the number of neurons in the fully-connected layer. Similarly, we conduct the experiments on dataset GTSRB-T only. In particular, we denote the number of neurons in the fully-connected layer by β. We vary the value of β from 64 to 256. Meanwhile, we also compare the performance with other conventional models when other parameters are fixed.

It is shown in Table IV that the proposed portable CNN outperforms other conventional models in terms of highest accuracy when the number of neurons in the fully-connected layer is varied from 64 to 256.

Next we evaluate the number of neurons in the fully-connected layer, β. We can see in Table IV, after we investigate the classification results with GTSRB dataset, the results shown that when $\beta = 256$, the experimental accuracy is the highest.

TABLE IV: Evaluation with number of neurons in the fully-connected layer

β	AlexNet	VGG-16	MCDNN	Factorization-Net	Portable CNN
$\beta = 64$	78.29%	89.34%	96.17%	82.66%	96.66%
$\beta = 128$	93.51%	96.82%	97.85%	96.54%	97.60%
$\beta = 256$	96.31%	98.60%	98.50%	97.11%	98.62%

IV. Conclusion

In this paper, we put forth a portable convolutional neural network used in traffic-sign classification. In particular, this model contains a stacked structure, in which several portable FCMs alternate with factorization convolutional layers. Our model has the merits including the small model size while maintaining high classification accuracy. For example, the proposed portable CNN model has model size of 5.8 MB, which are much smaller than other conventional CNN models. Meanwhile, the accuracy of the proposed model also outperforms other models. This is mainly because the optimized design on convolution layers and compression layers, consequently removing the redundant parameters. In the future, we will further evaluate the performance of the proposed CNN models by deploying the model in mobile embedded platforms, which have inferior computing capability to PC platforms.

References

[1] Alexandre Bargeton, Fabien Moutarde, Fawzi Nashashibi, and Benazouz Bradai. Improving pan-european speed-limit signs recognition with a new global number segmentation before digit recognition. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 349–354. IEEE, 2008.

[2] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249 – 259, 2018.

[3] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

[4] D Cireşan, U Meier, J Masci, and J Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Benjamin Hoferlin and Klaus Zimmermann. Towards reliable traffic sign recognition. In *Intelligent Vehicles Symposium*, pages 324–329, 2009.

[7] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(2):2012, 2012.

[9] Ke Lu, Zhengming Ding, and Sam Ge. Sparse-representation-based graph embedding for traffic sign recognition. *Traffic*, 31:32, 2012.

[10] Hengliang Luo, Yi Yang, Bei Tong, Fuchao Wu, and Bin Fan. Traffic sign recognition using a multi-task convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1100 – 1111, 2018.

[11] Angela F. Danil De Namor, Mohammad Shehab, Rasha Khalife, and Ismail Abbas. The german traffic sign recognition benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks*, pages 1453–1460, 2011.

[12] Yok-Yen Nguwi and Abbas Z Kouzani. Detection and classification of road signs in natural environments. *Neural computing and applications*, 17(3):265–289, 2008.

[13] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, ICML'16, pages 2217–2225, 2016.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *The 3rd International Conference on Learning Representations (ICLR)*, 2015.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. pages 1–9, 2014.

[16] Tian Tian, Ishwar Sethi, and Nilesh Patel. Traffic sign recognition using a novel permutation-based local image feature. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 947–954. IEEE, 2014.