Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3041117, 14 pages
https://doi.org/10.1155/2022/3041117

Research Article

Evaluation of Vision Transformers for Traffic Sign Classification

Yuping Zheng[1] **and Weiwei Jiang**[2]

[1]*College of Electrical, Energy and Power Engineering, Yangzhou University, Yangzhou 225009, China*
[2]*Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Weiwei Jiang; jwwthu@163.com

Received 25 December 2021; Revised 15 April 2022; Accepted 12 May 2022; Published 4 June 2022

Academic Editor: Yan Huang

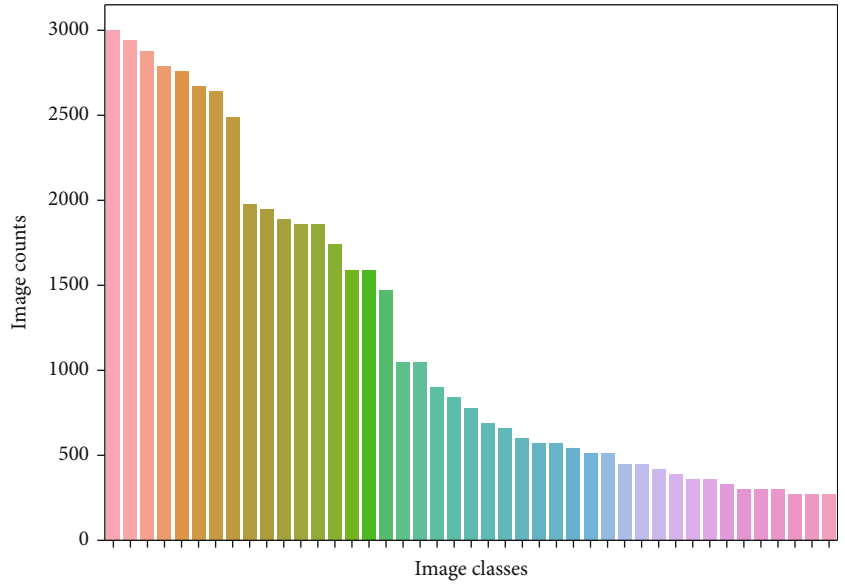
Traffic sign recognition is one of the most important tasks in autonomous driving. Camera-based computer vision techniques have been proposed for this task, and various convolutional neural network structures are used and validated with multiple open datasets. Recently, novel Transformer-based models have been proposed for various computer vision tasks and have achieved state-of-the-art performance, outperforming convolutional neural networks in several tasks. In this study, our goal is to investigate whether the success of Vision Transformers can be replicated within the traffic sign recognition area. Based on existing resources, we first extract and contribute three open traffic sign classification datasets. Based on these datasets, we experiment with seven convolutional neural networks and five Vision Transformers. We find that Transformers are not as competitive as convolutional neural networks for the traffic sign classification task. Specifically, there are performance gaps of up to 12.81%, 2.01%, and 4.37% existing for the German, Indian, and Chinese traffic sign datasets, respectively. Furthermore, we propose some suggestions to improve the performance of Transformers.

1. Introduction

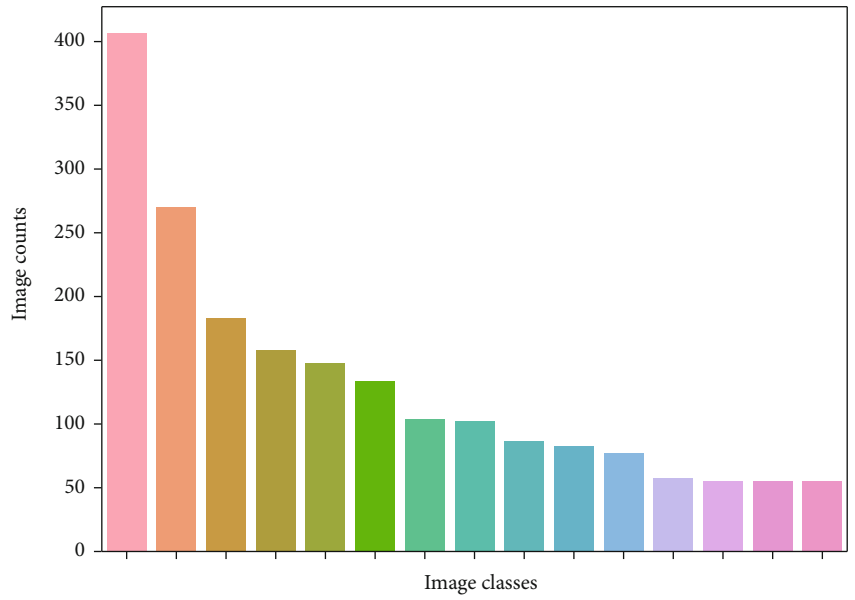
In recent years, both academia and industry have paid increasing attention to autonomous driving, which can be divided into two categories. One category is unmanned driving. More emphasis is placed on the autonomous driving of the car to achieve a comfortable driving experience or save labor costs. The representatives include unmanned vehicles from Internet giants, such as Baidu, Google, and Apple. Another category is the advanced driver assistance system (ADAS), represented by the advanced driver assistance system known as Autopilot 2.0 introduced by Tesla. Implementing automatic driving can improve rode safety and traffic efficiency. For people who die in traffic accidents every year, using on-board sensors to detect related risks can effectively reduce these accidents. Besides, it can save people's time and energy in learning to drive and human resources. For example, the demand for logistics distribution in China is increasing, in which automatic driving will make the entire process more efficient. Assisted driving is the premise of unmanned driving. Among the automatic driving, these functions that ADAS can realize using sensors include adaptive cruising, lane departure warning, traffic sign recognition, and so on.

On the hardware side, both radar and cameras are used in autonomous driving, for dealing with pedestrian detection, road condition detection, vehicle detection, traffic sign recognition, and more relevant problems. Different devices have their own advantages and disadvantages. Laser radar is mainly used to build a three-dimensional model of the surrounding environment in real time. Millimeter-wave radar is mainly used for adaptive cruising and automatic emergency braking. Ultrasonic radar is used for reversing reminders and automatic parking [1]. Cameras are widely used for lane departure warning, forward collision warning, traffic sign recognition, panoramic parking, and driver attention monitoring. Compared to laser radar, millimeter-wave radar, and ultrasonic radar, cameras are more suitable for pedestrian detection and traffic sign recognition, with a lower cost [2].

Among the different tasks in autonomous driving, traffic sign classification plays an important role in regulating traffic and driving safely and supports the development of

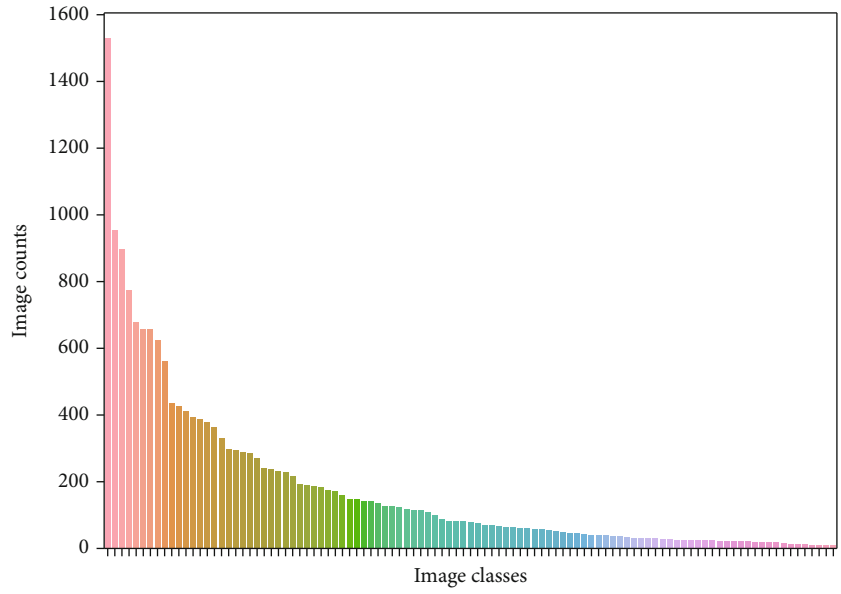


(a)



(b)

Figure 1: Continued.



(c)

Figure 1: Data distribution of the three traffic sign datasets: (a) the German traffic sign dataset; (b) the Indian traffic sign dataset; (c) the Chinese traffic sign dataset.

advanced driver assistance systems. Traditional computer vision methods are used to detect and classify traffic signs, but these solutions require much work to collect large-volume images manually, which is very costly and time-consuming [3]. Methods based on shape or color are also widely used, but they have a common weakness in several respects, such as illumination change, occlusion, scale change, rotation, and translation. Even though these problems can also be solved by machine learning, a large annotated data database is required. In recent years, deep learning has been proposed for traffic sign recognition and has achieved state-of-the-art performance. Research in the field of deep learning has led to the emergence of many models, such as VGG16 [4], ResNet [5], MobileNet [6], and Transformer [7]. As one of the important factors contributing to the success of deep learning, some open image datasets have been used in previous studies, e.g., the German Traffic Sign Recognition Dataset (GTSRB) [8].

In this study, we focus on the methods of traffic sign classification. As far as the authors know, no previous studies have applied different variants of Vision Transformers in this specific task. The Transformer structure is based on the attention mechanism, which was initially proposed to solve the sequence-to-sequence issue in the natural language processing area. Unlike traditional recurrent neural networks (RNNs), e.g., GRU or LSTM, and convolutional neural networks (CNNs), Transformer is an attention-only structure without using recursion or convolution operations and improving the parallel efficiency without damaging the performance [7]. Introduced into the computer vision field, Vision Transformers are proposed as novel solutions to vision-related issues. Various Vision Transformer structures are under development, including those we would use for this study. In summary, we use five variants of Vision Transformers, namely, the Vision Transformer model (ViT) [9], RealFormer [10], Sinkhorn Transformer [11], Nyströmformer [12], and Transformer in Transformer (TNT) [13]. As baselines, we also use seven convolutional neural networks, namely, VGG16, ResNet, DenseNet [14], MobileNet, SqueezeNet [15], ShuffleNet [16], and MnasNet [17]. To accomplish the evaluation and comparison, we collect and build three traffic sign datasets, which are then shared with the research community.

Table 1: Data distribution of the traffic sign datasets.

Dataset	Classes	Data balance
German	43	0.9252
Indian	15	0.9247
Chinese	103	0.8612

The questions we want to investigate in this study are as follows:

Whether Vision Transformers are effective for traffic sign classification?

Whether Vision Transformers are as effective as convolutional neural networks in this specific task?

Based on our experimental results, we find that Vision Transformers are more effective on smaller datasets. With increasing data size, their performance degrades considerably. Additionally, Vision Transformers are not as competitive as convolutional neural networks for the traffic sign classification task for now. Specifically, there are performance gaps of up to 12.81%, 2.01%, and 4.37% for the German, Indian, and Chinese traffic sign datasets between the best Transformer and the best CNN, respectively. CNNs

Figure 2: Image samples of the three traffic sign datasets: (a) the most popular five classes for the German traffic sign dataset; (b) images from the five classes with the least number in the German traffic sign dataset; (c) the most popular five classes for the Indian traffic sign dataset; (d) images from the five classes with the least number in the Indian traffic sign dataset; (e) the most popular five classes for the Chinese traffic sign dataset; (f) images from the five classes with the least number in the Chinese traffic sign dataset.

tend to converge faster than Vision Transformers and we use a longer epoch for Vision Transformers in this study. Among different Vision Transformer structures, RealFormer and Transformer are more effective variants of ViT, while Sinkhorn Transformer and Nyströmformer are less attractive for traffic sign classification.

Our contributions in this study are summarized as follows:

(1) We contribute three datasets for traffic sign classification with the same data format derived from open resources for comparing both convolutional neural networks and Vision Transformers. The datasets are publicly available (http://github.com/jwwthu/DL4Traffic/tree/main/Traffic_Sign_Datasets).

(2) According to the experiments on these three datasets, we demonstrate that Vision Transformers are not as competitive as convolutional neural networks for the traffic sign classification task for now limited to our experimental models

(3) We present a detailed comparison as a reference for future work in this paper as well as possible future research directions to improve the performance of Vision Transformers

The following parts of this paper are organized as follows. Related works are reviewed in Section 2. The datasets we contribute are described in Section 3. The models we investigated are discussed in Section 3. The experiments are analyzed in Section 4. Conclusions are drawn in Section 5.

2. Related Work

In this section, we give a brief overview of convolutional neural networks for image classification and the current progress in solving traffic sign recognition issues.

2.1. Convolutional Neural Networks. Convolutional neural networks have been widely used in the computer vision area as well as other fields including the traffic domain [18, 19].

Table 2: Data size of the traffic sign datasets.

Dataset	Training set	Validation set	Testing set
German	31,367	7,842	12,630
Indian	1,264	316	396
Chinese	11,627	2,907	3,634

In this study, we use seven different convolutional neural networks, covering both the normal and lightweight structures which are with small size of parameters. Lightweight CNNs are particularly designed for low-resource devices and suitable for deployment in cameras or smartphones. The CNNs we use include VGG16 [4], ResNet [5], DenseNet [14], MobileNet [6], SqueezeNet [15], ShuffleNet [16], and MnasNet [17]. As one of the pioneers, VGG16 [4] was proposed in 2014, which is composed of several convolutional layers and pooling layers and achieves a second top performance in the ImageNet competition, demonstrating the pros of using more convolutional layers. ResNet [5] was proposed in 2016, which introduces the residual structure and greatly improves the training efficiency of much deeper neural networks at that time. DenseNet [14] was proposed in 2017 and proposes a new network structure, e.g., a dense block. In the dense block, the input of each convolution layer is the concatenation of feature map outputs by all convolution layers before the block.

MobileNet [6] was proposed in 2017 and is a representative lightweight network with depth-wise separable convolution. The network width and input resolution of MobileNet can be arbitrarily changed by setting two hyperparameters, namely, the width multiplier and resolution multiplier. SqueezeNet [15] is another lightweight network with the fire module, in which both the squeeze layer and expand layer are used to reduce model parameters.

ShuffleNet [16] was proposed in 2017, which combines group convolution and depth-wise separable convolution. It proposes pointwise group convolution to avoid a large number of $1 \times 11 \times 1$ convolution computations. Channel

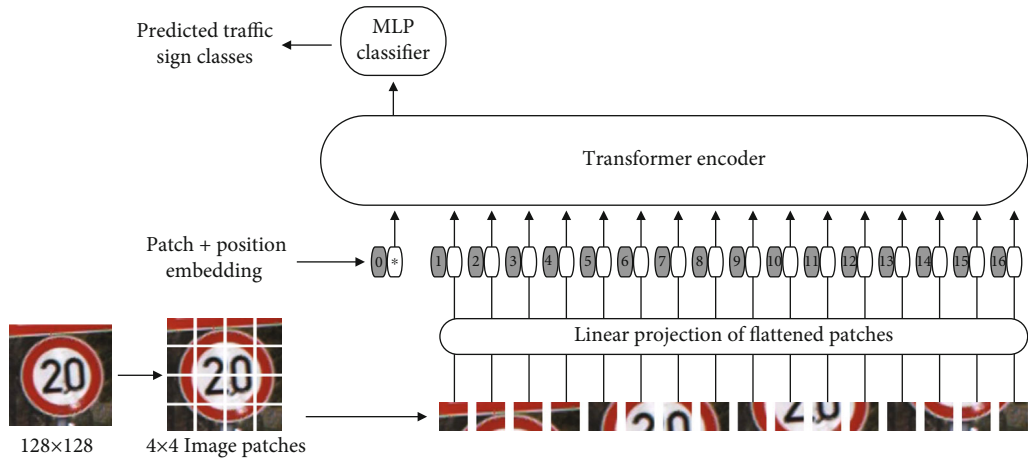


Figure 3: The Vision Transformer structure used in this study.

shuffle is used to alleviate the side effects brought by group convolution, and the network structure is designed based on the redundant block, so that the computation of the network can be reduced and the high performance can still be maintained. MnasNet [17] was put forward in 2019. The decomposed hierarchical search space maintains the network layer diversity, while the search space is still very simple, which is better than the current manually designed networks and automatically searched networks in both accuracy and time consumption.

2.2. Traffic Sign Recognition. Traffic sign recognition problems are widely seen in the literature. A comprehensive survey on traffic sign detection, tracking, and classification and the investigated details of algorithms, methods, and their specifications on detection, tracking, and classification can be found in [20].

Most of the previous work is aimed at improving the detection and classification performance with deep learning, especially convolutional neural networks. An improved traffic sign detection and recognition model is proposed for intelligent vehicles [21], which is based on the classical LeNet-5 model. The new model uses the Gabor kernel as the initial convolutional kernel and adds batch normalization processing after the pooling layer. An accuracy of 99.75% is achieved in the evaluation. To overcome the impact of scale and rotation, a new method is presented in [22], which only needs one reference image to recognize the traffic sign. Using the captured image pair, a virtual image is generated and further used. Experiments show that the proposed method significantly improves recognition performance with a minimum number of reference images and reaches 93.1% accuracy, which enjoys a 4.9% improvement over traditional methods. A weighted multi-CNN trained with a novel methodology in [23] finally achieved a state-of-the-art recognition accuracy of 99.59% when tested on the German traffic sign recognition benchmark dataset. Additionally, the proposed classifier is less complex than the existing classifiers.

Some of the existing studies focus on the lightweight models that are suitable for deployment on resource-limited devices. For resource-limited electronic devices, small, lightweight, accurate deep convolutional neural networks (ConvNets) are investigated in [24] for fast traffic sign recognition. Using uniform macroarchitecture and depthwise separable convolution, two models named qNet and sqNet are found to be more efficient with less parameters and computations. Two lightweight networks are used as the teacher network and student network in [25], which can obtain a higher recognition precision with fewer trainable parameters and are suitable for deployment on mobile embedded devices. A lightweight SPPN-CNN traffic sign recognition method is proposed in [26], which fuses techniques including image normalization, affine transformation, contrast limited adaptive histogram equalization (CLAHE), spatial pyramid pooling (SPP), and batch normalization (BN). The model achieves 98. 04% classification accuracy in GTSRB and a recognition rate of 3,000 fps in low GPU allocation.

Datasets are necessary for relevant studies. There are many studies with a focus on contributing new datasets. A dataset is built in [27] for traffic sign detection, which contains 10,500 images that are captured from Chinese roads under real environmental conditions from 73 traffic sign classes. Based on the new dataset, an application is further proposed for embedded implementation by using the deep learning technique based on convolutional neural networks. The proposed application achieves high performance in both accuracy and speed. In addition, the application was developed for embedded implementation thanks to the lightweight size of the deep learning model used for traffic sign detection.

Apart from Vision Transformers, multiple convolutional neural networks have also been evaluated and compared in previous studies. For example, an experimental comparison of eight traffic sign detectors based on deep neural networks is conducted in [28]. Different metrics are used, e.g., accuracy, speed, memory consumption, number of floating point operations, and number of learnable parameters that can be used in CNN. These works reveal valuable insights that help practitioners choose and deploy an appropriate traffic sign detector in actual systems.

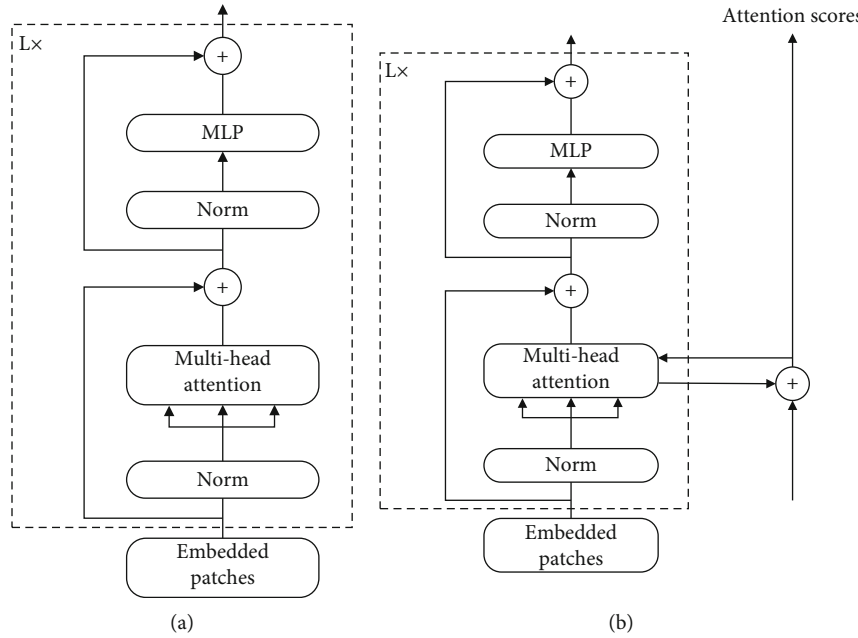


Figure 4: Transformer encoder structures: (a) ViT [9] and (b) RealFormer [10].

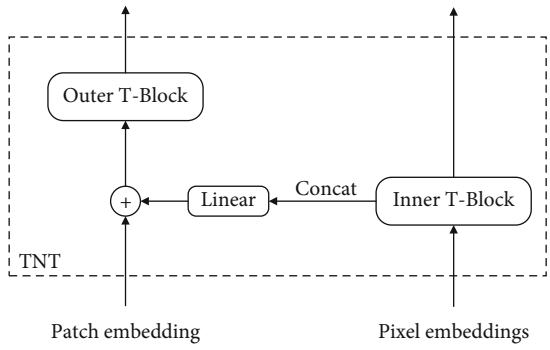


Figure 5: The Transformer in Transformer block [13].

Table 3: Hyperparameters for ViT and its variants.

Hyperparameter	Value
Number of classes	Refer to Table 1
Image patch size	32
Output dimension of the encoder	1024
Number of Transformer blocks	6
Number of heads in multihead attention layer	16
Dimension of the MLP layer	2048
Dropout rate	0.1
Embedding dropout rate	0.1

Table 4: Hyperparameters for TNT.

Hyperparameter	Value
Number of classes	Refer to Table 1
Image patch size	32
Image pixel size	4
Dimension of patch token	256
Dimension of pixel token	12
Number of TNT blocks	4
Attention dropout rate	0.1
Feedforward dropout rate	0.1

3. Method

3.1. Datasets. In this study, we contribute three different traffic sign datasets extracted from previous studies, namely, the German traffic sign, the Indian traffic sign, and the Chinese traffic sign dataset. Since our research focuses on traffic sign classification, we leverage the existing traffic sign detection datasets with bounding boxes and labels to build these traffic sign classification benchmarks. Specifically, the German traffic sign dataset comes from the famous German Traffic Sign Recognition Dataset (GTSRB) [8]. The Indian traffic sign dataset comes from an open Indian Cautionary Traffic sign dataset (http://ieee-dataport.org/documents/indian-cautionary-traffic-sign-data-set). The Chinese traffic sign dataset comes from the CSUST Chinese Traffic Sign Detection Benchmark [25]. For all the traffic sign images extracted from the original images using bounding boxes, we unify their resolution by resizing all images to 128×128 pixels. For the German and Indian traffic signs, we use the original classes as the labels. For the CSUST Chinese Traffic Sign Detection Benchmark, only three labels are provided, i.e., mandatory, prohibitory, and warning, which are not enough for traffic sign classification. We randomly assigned the images to the two of us. Mr. Jiang marks the result, and I cross-check to prevent personal bias. We choose only the classes with at least 10 images. The same requirement for the minimum number of images within a single class is also applied for the German and Indian datasets. Finally, we retain 43 classes for the German traffic sign dataset, 15

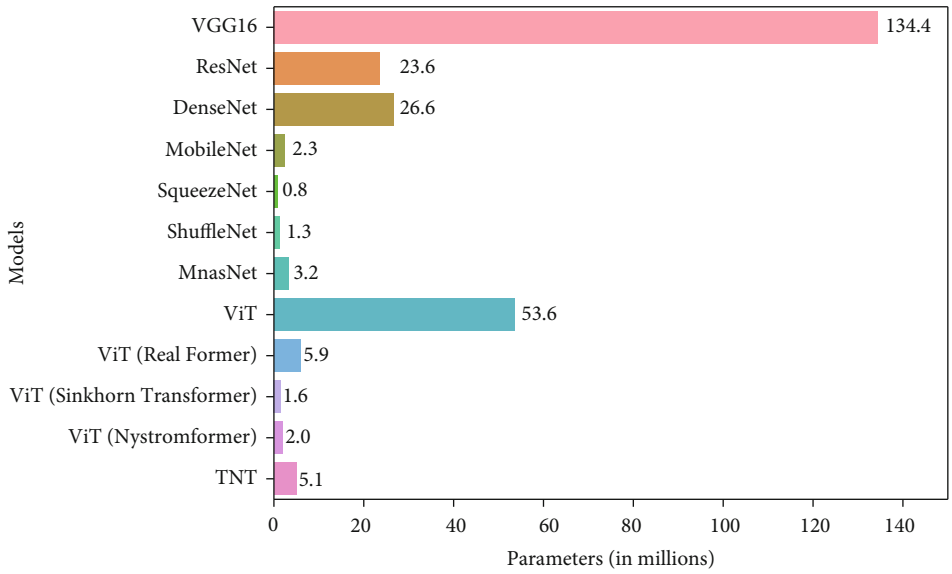


Figure 6: The comparison of model parameters.

classes for the Indian traffic sign dataset, and 103 classes for the Chinese traffic sign dataset.

The data sample distributions of different classes are shown in Figure 1. For a better visualization, we plot the distribution in decreasing order and omit the traffic sign class names in the x-axis. We can further use Shannon entropy to evaluate the dataset imbalance. For example, the Chinese traffic sign data distribution is more skewed than the other two, which corresponds to the case in which the dataset is more imbalanced. The Shannon entropy H is calculated as follows:

$$H = -\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}, \tag{1}$$

where k is the number of classes, c_i is the number of images of class i, and n is the total number of images in a dataset. When the dataset is highly skewed, e.g., there is only one class, the entropy H is zero. In contrast, if the dataset is perfectly balanced, i.e., each class has n/k images, the entropy H is $\log k$.

Then, we can define a data balance metric that is normalized to $[0, 1]$ as follows:

$$\text{DataBalance} = \frac{H}{\log k}, \tag{2}$$

where H is the Shannon entropy and k is the number of classes. When there is only one class, the data balance is 0. When the dataset is perfectly balanced, the data balance is 1. Using this metric, we calculate the data balance for each traffic sign dataset and show the result in Table 1. The German traffic sign dataset has the highest data balance, while the Chinese traffic sign dataset has the lowest data balance, which is consistent with the skewness in Figure 1.

Image samples from the classes with the most number and the least number are both shown in Figure 2. While these traffic sign images are extracted from different sources, they are unified into the same data format with the size of 128×128, as demonstrated in Figure 2. We can also observe that these images are captured with different angles, lighting conditions, and occlusion situations, which makes the traffic sign classification task more challenging.

We further divide the three datasets into training/validation/testing subsets, and the data sizes of the traffic sign datasets are shown in Table 2. The German traffic sign dataset has the most images, as shown in Table 2, while the Chinese traffic sign dataset has the most classes, as shown in Table 1.

3.2. Models. In this study, we use the state-of-the-art Vision Transformer model (ViT) [9] and its variants, including RealFormer [10], Sinkhorn Transformer [11], Nyströmformer [12], and Transformer in Transformer (TNT) [13]. Since a full collection of Transformer structures is beyond the scope of this study, more variants can be found in recent surveys.

3.2.1. Vision Transformer. The overall structure of the standard Vision Transformer used in our study is shown in Figure 3. The notations are consistent with the original paper [9]. In the ViT structure, the first step is to divide the original images into small patches, which are similar to the tokens used in natural language processing tasks. For example, the 128×128 image is cut into $4 \times 4 = 16$ 32×32 image patches, as shown in Figure 3. Then, these patches are organized in a vector format and added to the positional encoding as the input to the Transformer encoder. Instead of using the fixed method in the original Transformer, positional encoding is learned as parameters in the Vision Transformer. Different variants of Transformer can be used as the encoders. Finally, the first output of the Transformer encoder is used as the input of the multilayer perceptron (MLP) classifier, which generates the final predicted traffic sign classes.

Table 5: Training settings.

Name	Value	
Batch size	64	
Optimizer	Adam	
Learning rate	$3e-5$	
Loss function	Cross entropy	
	CNNs	Transformers
Epoch	German & Chinese datasets	Indian dataset
	20	50	100

Table 6: Evaluation results on the traffic sign datasets.

Model	Germany			India			China		
	Training	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing
Convolutional neural networks									
VGG16	99.89%	99.94%	98.84%	99.77%	98.75%	98.44%	99.65%	99.52%	99.21%
ResNet	99.88%	99.82%	98.37%	99.92%	99.06%	97.47%	99.72%	99.41%	99.25%
DenseNet	99.97%	99.90%	98.82%	100.00%	99.38%	98.59%	99.95%	99.69%	99.42%
MobileNet	99.87%	99.56%	97.41%	99.77%	96.83%	95.98%	99.70%	98.40%	98.05%
SqueezeNet	99.52%	99.56%	96.69%	98.54%	96.21%	96.65%	99.21%	98.91%	98.24%
ShuffleNet	98.96%	98.81%	95.49%	99.92%	98.75%	99.11%	98.96%	98.84%	95.53%
MnasNet	99.96%	99.18%	96.17%	100.00%	98.10%	96.80%	99.67%	99.18%	96.26%
Vision Transformers									
ViT	98.27%	98.89%	83.77%	98.80%	96.54%	97.10%	94.35%	94.79%	93.53%
ViT (RealFormer)	98.45%	99.19%	86.03%	98.67%	95.94%	96.65%	93.62%	94.21%	94.22%
ViT (Sinkhorn Transformer)	94.69%	97.04%	82.29%	95.99%	94.02%	94.79%	80.68%	85.61%	84.71%
ViT (Nyströmformer)	79.15%	83.15%	62.41%	90.47%	80.13%	80.95%	86.97%	79.08%	79.10%
TNT	96.83%	97.73%	84.39%	97.71%	92.75%	94.42%	96.25%	94.52%	95.05%
Performance gap									
CNN (best)-Transformer (best)		12.81%			2.01%			4.37%	

The original Transformer encoder structure is shown in Figure 4(a). In each block, there are the following components: multihead self-attention (MSA) [9], MLP, and layer normalization (LN). We denote the original image as $\mathbf{x} \in \mathscr{R}^{H\times W\times C}$, where $(H, W) = (128,128)$ is the resolution of the original image and $C = 3$ is the number of channels. Then, we reshape the original image into a sequence of patches $x_p \in \mathscr{R}^{N\times(P^2\cdot C)}$, where (P, P) is the resolution of each image patch and $N = HW/P^2$ is the total number of patches. The Transformer encoder encodes the image patch into a vector with the size of D. We add the traffic sign class x_{class} as a learnable embedding. The input of the Transformer encoder is denoted as $\mathbf{z}_0$, and the output is denoted as $\mathbf{z}_L$, where L is the number of duplicate blocks used in the encoder. Then, the overall encoder process can be stated as follows [9]. In the first step, the input image patches are added with positional encoding:

$$\mathbf{z}_0 = \left[x_{\text{class}}; x_p^1\mathbf{E}; x_p^2\mathbf{E}; \cdots; x_p^N\mathbf{E}\right] + \mathbf{E}_{\text{pos}}, \tag{3}$$

where $\mathbf{E} \in \mathscr{R}^{(P^2\cdot C)\times D}$ is the input embedding and $\mathbf{E}_{\text{pos}} \in \mathscr{R}^{(N+1)\times D}$ is the positional encoding. Then, the MSA, MLP, and LN components are applied L times:

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \tag{4}$$

$$\mathbf{z}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{z}_{\ell-1}'\right)\right) + \mathbf{z}_{\ell-1}', \tag{5}$$

where $\ell = 1, \cdots, L$. Then, the final output $\mathbf{y}$ of the encoder is

$$\mathbf{y} = \text{LN}\left(\mathbf{z}_L^0\right), \tag{6}$$

where $\mathbf{z}_L^0$ denotes the first element of $\mathbf{z}_L$.

3.2.2. RealFormer. RealFormer [10] is a simple residual attention layer Transformer architecture, which is shown in Figure 4(b). The key idea is to add the residual connection to propagate attention scores between the multihead attention blocks, which is simple yet effective.

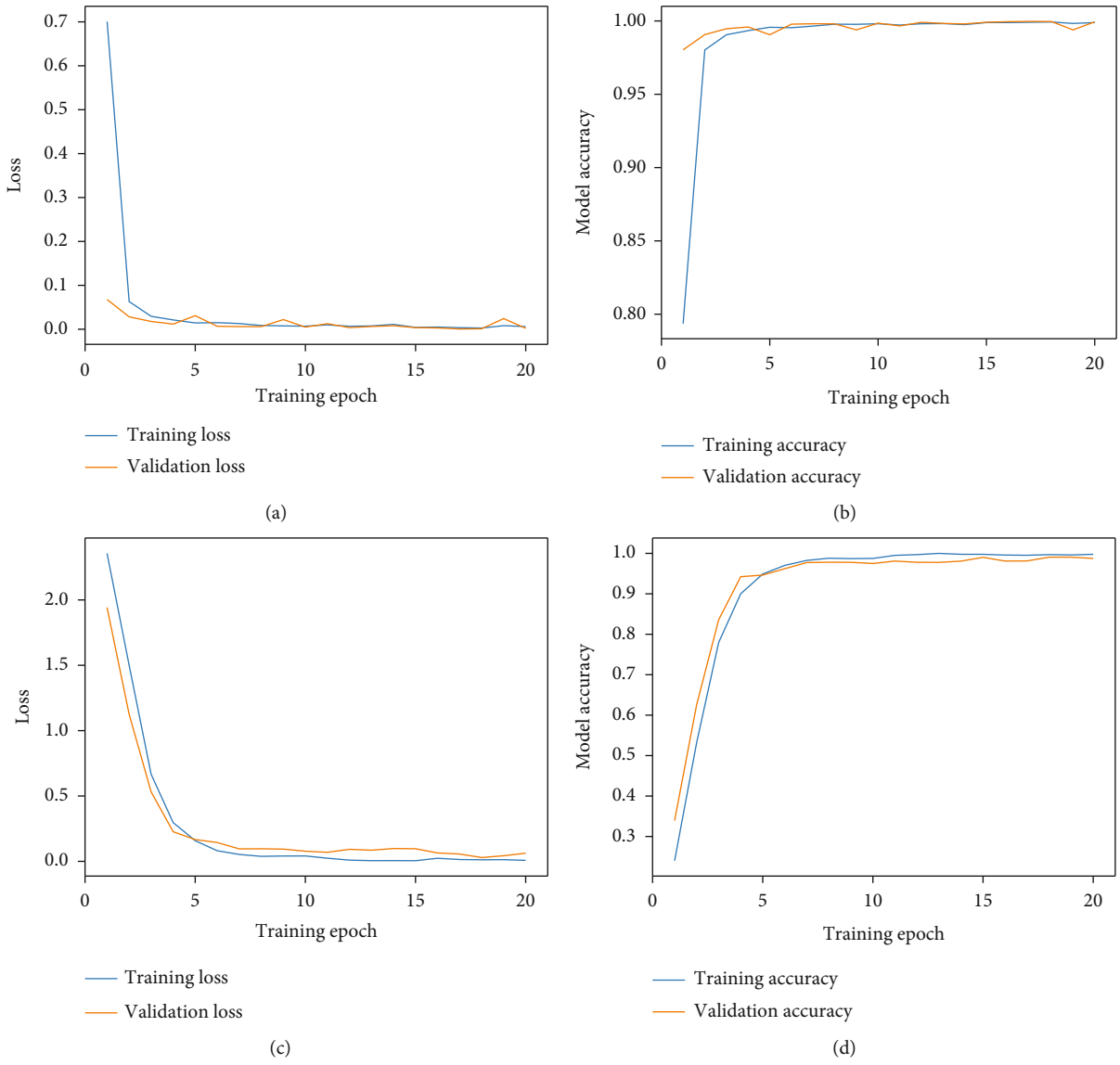


Figure 7: Continued.

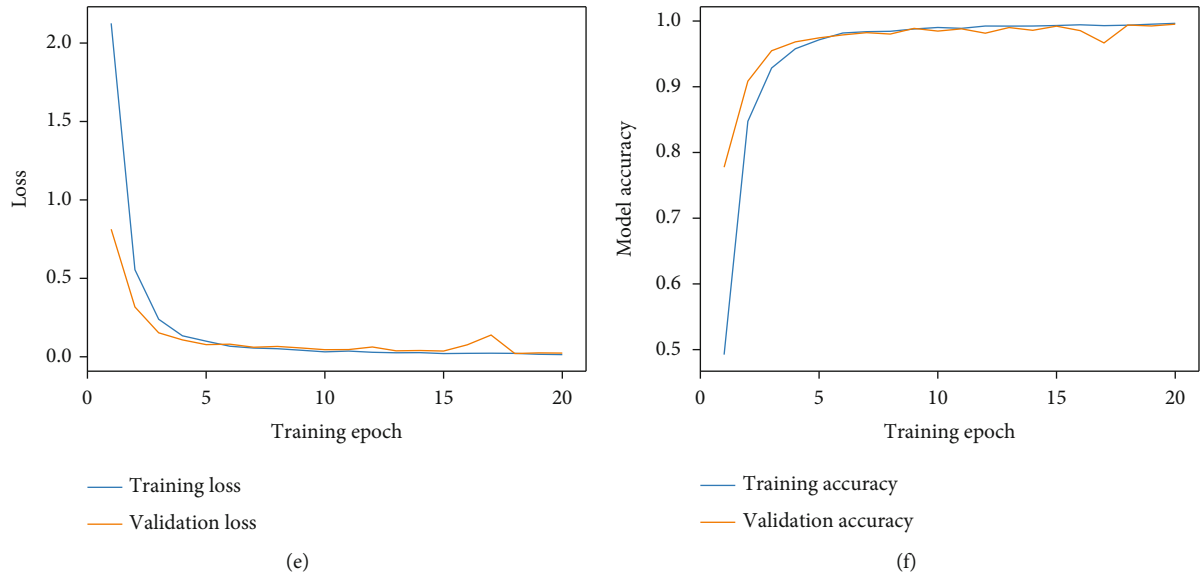


Figure 7: Training histories of VGG16: (a) loss versus epoch in the German traffic sign dataset; (b) accuracy versus epoch in the German traffic sign dataset; (c) loss versus epoch in the Indian traffic sign dataset; (d) accuracy versus epoch in the Indian traffic sign dataset; (e) loss versus epoch in the Chinese traffic sign dataset; (f) accuracy versus epoch in the Chinese traffic sign dataset.

3.2.3. Sinkhorn Transformer. The key idea of Sinkhorn Transformer [11] is to use a cost matrix to limit attention, with the implementation of causal Sinkhorn balancing and sortCut, which tailors Sinkhorn attention and minimizes model parameters greatly.

3.2.4. Nyströmformer. The key idea of Nyströmformer [12] is to use the Nyström method to approximate the standard self-attention operation. The standard self-attention operation has a complexity of $O(n^2)$ where n is the length of an input sequence, while Nyströmformer approximates the attention with a complexity of $O(n)$, which reduces both the memory consumption and running time.

3.2.5. Transformer in Transformer. The key idea of Transformer in Transformer (TNT) [13] is to use two Transformer blocks for both the patch embedding and pixel embeddings, as shown in Figure 5. The outer Transformer block (Outer T-Block) is used for patch embedding as usual, while the inner Transformer block (Inner T-Block) is added to extract local features from pixel embeddings. Then, the pixel-level features are mapped and added with the patch embedding with the linear layer. Transformer in Transformer achieves a better image classification performance by stacking the TNT block, when compared with the standard Vision Transformer model.

4. Experimental Results and Discussion

In this section, we present the evaluation experiments and the results analysis. All the experiments are performed on a desktop with a Windows 10 OS, Intel i5-9600K CPU, and NVIDIA GeForce RTX 2070 GPU. All the models are implemented with Python and the PyTorch package. We use the off-the-shelf CNN implementations provided by PyTorch with the default settings. The hyperparameter settings for ViT and its variants are shown in Table 3, while the hyperparameter settings for TNT are shown in Table 4.

The comparison of the total parameters is shown in Figure 6. Compared with the original Vision Transformer, all variants reduce the parameters with a ratio varying from $9x$ to $33x$. However, compared with CNNs, Transformers may not have the advantage of using fewer model parameters; e.g., SqueezeNet uses the fewest number of parameters among the models evaluated in this study.

The training settings are shown in Table 5. The major difference between the CNNs and Transformers is the choice of epochs. We use a smaller training epoch for CNNs because the off-the-shelf CNNs are pretrained on the ImageNet dataset and tend to converge faster than Transformers, as we show later. Another reason is that Vision Transformer requires a longer training time to be effective as shown in previous studies [29]. Furthermore, we choose a longer training time for the Transformers applied to the Indian dataset because this dataset is far smaller than the other two datasets, and by practice, we found that it would be better to train the Transformers even longer.

The comprehensive evaluation results are shown in Table 6. We first evaluate the performance gap between CNNs and Transformers by calculating the difference in the testing accuracy between the best CNN and the best Transformer evaluated in this study. We find that Transformers are not as competitive as CNNs for the traffic sign classification task. Specifically, there are performance gaps of up to 12.81%, 2.01%, and 4.37% for the German, Indian, and Chinese traffic sign datasets, respectively. We also find that the performance gap decreases with a smaller data size, with a longer training time cost used by Transformers.

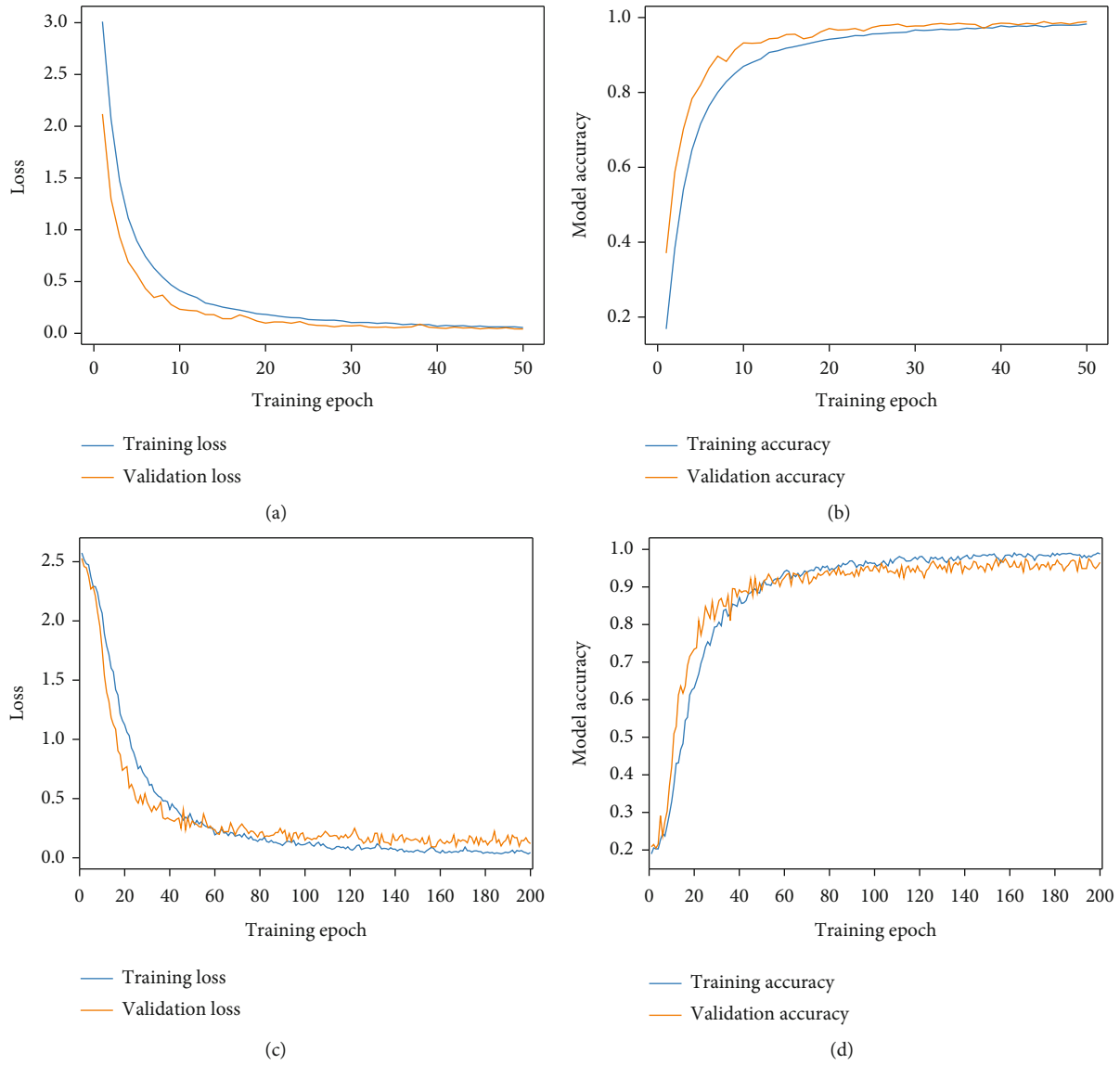


Figure 8: Continued.

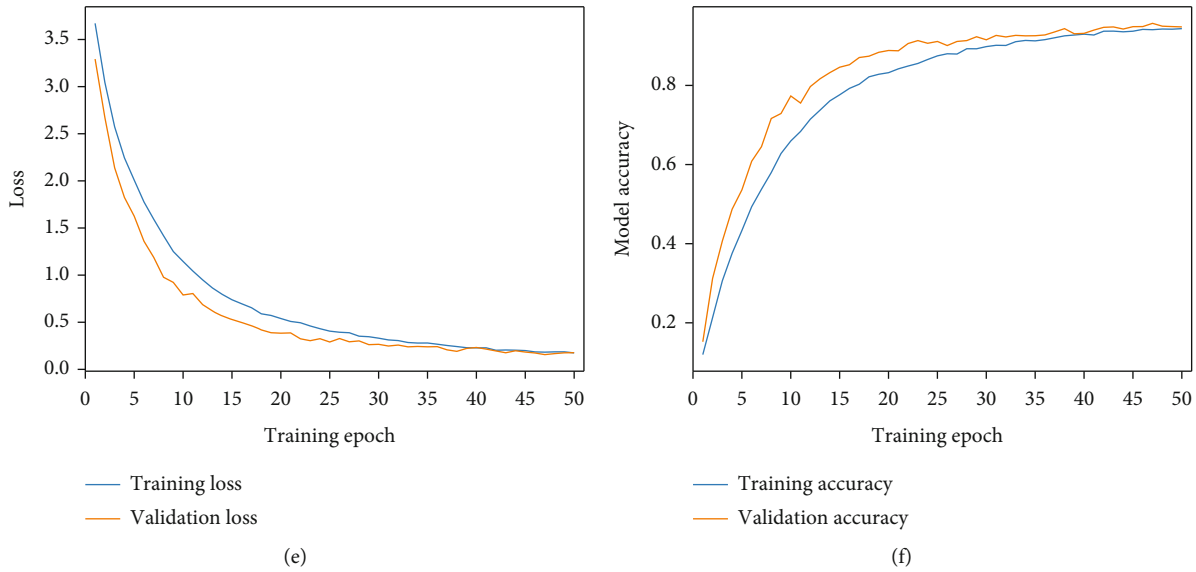


FIGURE 8: Training histories of ViT: (a) loss versus epoch in the German traffic sign dataset; (b) accuracy versus epoch in the German traffic sign dataset; (c) loss versus epoch in the Indian traffic sign dataset; (d) accuracy versus epoch in the Indian traffic sign dataset; (e) loss versus epoch in the Chinese traffic sign dataset; (f) accuracy versus epoch in the Chinese traffic sign dataset.

However, the generalization ability of Transformers is not as strong as that of CNNs, as shown in the German traffic sign dataset, when both training and validation accuracies are comparable with CNNs, but the testing accuracy drops dramatically. We also find the RealFormer and Transformer in Transformer are more effective variants of ViT, while Sinkhorn Transformer and Nyströmformer are less attractive for traffic sign classification.

For a better comparison between CNNs and Transformers, we choose VGG16 and ViT as representatives and present the loss and accuracy versus epoch plots during the training process in Figures 7 and 8. As we discussed earlier in this section, CNNs converge much faster, as shown in example VGG16. For the Transformer case, we observe that it converges more slowly on the small Indian dataset; thus, we use a longer training time.

Based on our evaluation results, some future directions can be summarized as follows. With an increasing number of Transformer variants being proposed for the visual tasks, it is still unclear which variant would be more effective from the theoretical perspective. Currently, trial-and-error is unavoidable for choosing the best Transformer structure that may surpass convolutional neural networks and be applied in real-world autonomous driving systems. Our evaluations could only be the first step, and more comparisons are needed. Another direction is to pretrain the Transformers on a larger image dataset and use the transfer learning scheme for other tasks. It has been proven effective in previous studies and is worthy of further investigation for traffic sign classification based on our proposed datasets which are publicly available. In addition, Transformer has more parameters and higher computational complexity, so we can try to put forward new structures to reduce computation in the future.

5. Conclusion

Researches on the application of CNN and Vision Transformer in computer vision have mushroomed in the past years. In image recognition, CNN has natural inductive bias for image problems, such as translation invariance and the bionic characteristics of CNN, which makes CNN easier on image problems; in contrast, Transformer does not have this advantage [30]. Besides, in driver distracted detection, CNN outperforms Vision Transformer for specific downstream tasks or dealing with a low amount of data in [31]. However, Transformer also shows that it can outperform CNN in some aspects, e.g., in Digital Holography (DH); Transformer can reach similar accuracy and is more robust than CNN with the problem of autofocusing as a classification problem [32]. As for weed and crop classification of high-resolution UAV images, ViT models also perform better compared to state-of-the-art CNN-based models EfficientNet and ResNet [33].

In this paper, we find that CNNs are more competitive than Transformers for the traffic sign classification task. Specially, there are performance gaps of up to 12.81%, 2.01%, and 4.37% for the German, Indian, and Chinese traffic sign datasets, respectively. As we can see, with a smaller data size, with a longer training time cost used, the performance gap is getting smaller.

Traffic sign recognition and autonomous driving remain to be very challenging tasks nowadays. The various issues for autonomous driving are not fully solved even with the development of deep learning which is powered by convolutional neural networks. Vision Transformers are proposed as a new promising solution; however, their performance is not as competitive as existing CNNs for the traffic sign classification task based on our experimental results, as we validated in this study. The performance gap between Transformers and CNNs may be filled with more effective Transformer

structures or pretraining strategies, e.g., the combination of convolution and attention, which are left for future studies.

Abbreviations

ADAS:	Advanced driver assistance system
GTSRB:	German Traffic Sign Recognition Dataset
RNNs:	Recurrent neural networks
GRU:	Gated recurrent unit
LSTM:	Long-short-term memory
CNNs:	Convolutional neural networks
ViT:	Vision Transformer
SPPN-CNN:	Spatial pyramid pooling convolutional neural network
CLAHE:	Contrast limited adaptive histogram equalization
SPP:	Spatial pyramid pooling
BN:	Batch normalization
CSUST:	Changsha University of Science and Technology
TNT:	Transformer in Transformer
LN:	Layer normalization
MSA:	Multihead self-attention
MLP:	Multilayer perceptron
Outer T-Block:	Outer Transformer block
Inner T-Block:	Inner Transformer block.

Data Availability

The datasets are available online. The URL is as follows: https://github.com/jwwthu/DL4Traffic/tree/main/Traffic_Sign_Datasets.

Conflicts of Interest

The authors declare that they have no competing interests.

Authors' Contributions

Yuping Zheng and Weiwei Jiang were responsible for the conceptualization, methodology, formal analysis and investigation, original draft preparation, review and editing, and resources. Weiwei Jiang was responsible for the supervision. The authors read and approved the final manuscript.

References

[1] L. Zhaohua and G. Bochao, "Radar sensors in automatic driving cars," in *5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, pp. 239–242, Nanchang, China, 2020.

[2] C. Yan, X. Wenyuan, and J. Liu, "Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicle," *Def Con*, vol. 24, no. 8, p. 109, 2016.

[3] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, Eds., "Deep learning vs. traditional computer vision," in *Science and information conference*, vol. 943, pp. 128–144, 2020.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for largescale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[6] G. Andrew and Z. Menglong, "Efficient convolutional neural networks for mobile vision applications," 2017, http://arxiv.org/abs/1704.04861.

[7] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[8] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: a multi-class classification competition," in *In The 2011 international joint conference on neural networks*, San Jose, CA, USA, 2011.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2020, http://arxiv.org/abs/2010.11929.

[10] R. He, A. Ravula, B. Kanagal, and J. Ainslie, "Realformer: transformer likes residual attention," 2020, http://arxiv.org/abs/2012.11747.

[11] Y. Tay, D. Bahri, L. Yang, D. Metzler, and D.-C. Juan, "Sparse sinkhorn attention," in *International Conference on Machine Learning*, pp. 9438–9447, 2020.

[12] Y. Xiong, Z. Zeng, R. Chakraborty et al., "Nyströmformer: a Nyström-based algorithm for approximating self-attention," 2021, http://arxiv.org/abs/2102.03902.

[13] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," 2021, http://arxiv.org/abs/2103.00112.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA, 2017.

[15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, http://arxiv.org/abs1602.07360.

[16] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffeNet: an extremely efficient convolutional neural network for mobile devices," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, USA, 2018.

[17] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, Eds., "Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, Long Beach, CA, USA, 2019.

[18] W. Jiang and L. Zhang, "Geospatial data to images: a deep-learning framework for traffic forecasting," *Tsinghua Science and Technology*, vol. 24, no. 1, 2018.

[19] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: a survey," 2021, http://arxiv.org/abs/2101.11174.

[20] S. BWali, M. A. Abdullah, M. A. Hannan et al., "Vision-based traffic sign detection and recognition systems: current trends and challenges," *Sensors*, vol. 19, no. 9, 2019.

[21] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, "Improved traffic sign detection and recognition algorithm for intelligent vehicles," *Sensors*, vol. 19, no. 18, p. 4021, 2019.

[22] R. Yazdan and M. Varshosaz, "Improving traffic sign recognition results in urban areas by overcoming the impact of scale

and rotation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 171, pp. 18–35, 2021.

[23] S. Natarajan, A. K. Annamraju, and C. S. Baradkar, "Traffic sign recognition using weighted multi-convolutional neural network," *IET Intelligent Transport Systems*, vol. 12, no. 10, 2018.

[24] X. Luo, J. Zhu, and Y. Qingying, "Efficient convNets for fast traffic sign recognition," *IET Intelligent Transport Systems*, vol. 13, no. 6, 2019.

[25] J. Zhang, W. Wang, C. Lu, J. Wang, and A. K. Sangaiah, "Lightweight deep network for traffc sign classiffication," *Annals of Telecommunications*, vol. 75, no. 7, 2020.

[26] D. E. N. G. Tianmin, F. A. N. G. Fang, and Z. H. O. U. Zhenhao, "Traffic sign recognition based on improved convolutional neural network with spatial pyramid pooling," *Journal of Computer Applications*, vol. 40, no. 10, 2020.

[27] R. Ayachi, M. Af, Y. Said, and M. Atri, "Traffic signs detection for real-world application of an advanced driving assisting system using deep learning," *Neural Processing Letters*, vol. 51, no. 1, pp. 837–851, 2020.

[28] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332–344, 2018.

[29] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: a survey," 2021, http://arxiv.org/abs/2101.01169.

[30] E. Arkin, N. Yadikar, Y. Muhtar, and K. Ubul, "A survey of object detection based on CNN and transformer," in *In 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, pp. 99–108, Chengdu, China, 2021.

[31] H. V. Koay, J. H. Chuah, and C.-O. Chow, "Convolutional neural network or vision transformer? Benchmarking various machine learning models for distracted driver detection," in *In TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*, Auckland, New Zealand, 2021.

[32] S. Cuenat and R. Couturier, "Convolutional neural network (CNN) vs visual transformer (ViT) for digital holography," 2021, http://arxiv.org/abs/2108.09147.

[33] R. Reedha, E. Dericquebourg, R. Canals, and A. Haane, "Vision transformers for weeds and crops classiffication of high resolution UAV images," 2021, http://arxiv.org/abs/2109.02716.