Proceedings of the 3rd International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/125/2025.21282

Comparative Analysis of Deep Learning Models for Efficient Traffic Sign Recognition

Tingting Lin[1,a,*]

[1]*College of Transportation Engineering, Tongji University, Shanghai, China*
a. 2251800@tongji.edu.cn
**corresponding author*

Abstract: With the rapid development of autonomous driving technologies and intelligent traffic systems, traffic sign recognition, as a key component of ensuring road safety and optimizing traffic management, draws widespread attention. Based on the German Traffic Sign Recognition Benchmark (GTSRB) dataset, the paper aims to achieve efficient and accurate recognition within a short period and then constructs three models, including customized convolutional neural networks (CNN), ResNet and MobileNet. The paper compares the performances and applicability of different models through test accuracy and inference time. The experimental results show that the customized CNN model takes only 2 seconds with 95.35% accuracy, achieving a good balance between accuracy and inference efficiency. The ResNet model has a slightly higher accuracy, but the time taken is 6 times longer than that of the customized CNN model, while the MobileNet model has an accuracy of only 87.77%. The research in this paper provides a theoretical basis for model selection in traffic sign recognition systems, and it is of vital importance in the development of intelligent transportation.

Keywords: Traffic sign recognition, CNN, ResNet-18, MobileNetV2.

1. Introduction

With the constant improvement of computational power and the development of big data technologies, as a powerful machine learning method, deep learning has achieved significant breakthroughs in various applications. It simulates the working mechanisms of neurons in human brains through multi-layer neural networks, thus automatically extracting features from massive data and making decisions [1]. In the field of Computer Vision, deep learning models show their good image processing ability, which enables them to complete multiple tasks efficiently, such as object recognition, object detection, image classification and image segmentation [2]. The Convolutional Neural Networks (CNN) model is one of the most important models in deep learning, and it greatly increases the accuracy and efficiency of image analysis by imitating the visual processing mechanisms in biological systems, becoming the mainstream method in image-related tasks currently [3].

As the core part of autonomous driving systems and intelligent traffic management, traffic sign recognition has attracted much attention in recent years [4]. Traffic signs play a crucial role in modern road transportation, they offer extensive information to drivers, including road conditions, speed limit, warnings and so on, so it is essential to recognize traffic signs accurately and instantly to enhance road safety and reduce traffic accidents [5].

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/125/2025.21282

However, a variety of circumstances, including weather, light variation, angle change and occlusion of signs, typically impact the images of traffic signs and these will make recognition more complex. The traditional methods based on hand-crafted features can not always work well when facing challenges, as a result, the methods based on deep learning models become the mainstream in such research gradually.

In their 2020 paper, Radu et al. suggested a CNN model for traffic sign identification that was built on the LeNet architecture. They use the German Traffic Sign Recognition Benchmark (GTSRB) dataset to train and test their model, implement real-time applications with a central processing unit(CPU), and finally achieve a recognition accuracy of 95%[6]. In 2021, Haque et al. proposed a novel lightweight CNN architecture, Deep Thin. It can be trained quickly without a graphics processing unit (GPU), and its recognition accuracy is over 99% when applied to both the GTSRB dataset and the Belgian Traffic Sign Classification (BTSC) dataset. This model solves the problem of how to recognize traffic signs accurately and efficiently with limited resources, which is of great importance for low-cost and energy-saving autonomous driving systems [7]. In 2024, Alawaji et al. proposed a deep learning model combining YOLOv7 object detection and multi-task learning(MLT), sharing convolutional layer parameters to increase the accuracy of traffic sign recognition. The model achieves an accuracy of 99.07% during the training phase and performs high effectiveness when taking a real-time test on the roads in Riyadh, Saudi Arabia [8].

The paper aims to complete the recognition task for the GTSRB dataset by constructing three CNN models and evaluates the capabilities and inference speeds of models to explore the feasibility of achieving high-accuracy traffic sign recognition within a short time. The paper will introduce the selected dataset at first, and then construct the customized CNN model, ResNet-18 model and MobileNetV2 model with a description of model architectures and parameters. Subsequently, through experimental results, the paper will conclude and propose prospects.

2. Experimental Data and Model Construction

The paper chooses the GTSRB as the dataset for research and completes the recognition task with a CNN model and its two variants, the ResNet-18 model and the MobileNetV2 model.

2.1. Dataset Description

GTSRB is a dataset published by the International Joint Conference on Neural Networks (IJCNN) in 2011, focusing on single-image and multi-class classification problems. It contains more than 50,000 images in total, all taken from real-world scenes, with some examples shown in Figure 1.



Figure 1: GTSRB dataset (Picture credit: Original).

DOI: 10.54254/2755-2721/125/2025.21282

The GTSBR dataset is composed of 39,209 training images and 12,630 test images, and these images are divided into 43 classes. They are taken under various conditions, including different sizes, angles, and lighting, which increases the difficulty of the classification task. Relevant information of training images and test images (including image width, length, and specific location of signs in the images and their categories) is stored in two CSV files, and classification-related information (including color, shape and sign ID) is stored in another CSV file.

2.2. Construction CNN

The architecture of the customized CNN model is shown in Figure 2, containing four convolutional layers, two max-pooling layers, three dropout layers, a flattened layer and a fully connected layer.

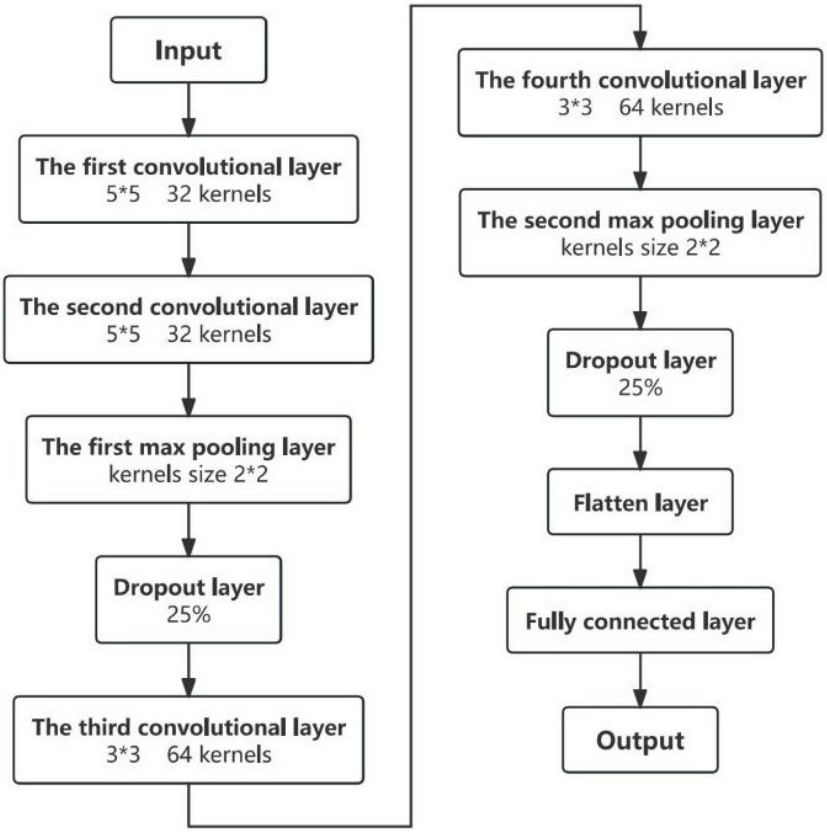


Figure 2: CNN architecture (Picture credit: Original).

The first and second convolutional layers use 32 filters with the size of 5*5 to extract the basic features of images and cover a bigger area, while the third and fourth convolutional layers use 64 filters with the size of 3*3 to deeply analyze the detailed and complex features. All convolutional layers use the Rectified Linear Unit (ReLU) function as an activation function. Pooling layers use max pooling with 2*2 filters to reduce the size of the feature map and decrease the spatial sensitivity of features. The dropout probabilities for the three dropout layers are set to 25%,25% and 50%, respectively, to avoid overfitting. After construction, the model contains 242,251 parameters in total, and all of them can be trained.

2.3. ResNet

The ResNet model was proposed by He et al. in 2016, and it effectively solves the degradation arising from deep network architectures by adding residual connections, which is a major historic

DOI: 10.54254/2755-2721/125/2025.21282

breakthrough in the field of deep learning [9]. The core idea of this model is similar to short-circuit connections in circuits, allowing the input to pass information straight to later layers while avoiding some intermediate layers, ensuring the effective flow of information in deeper networks, thus preventing the degradation phenomenon.

The paper chooses the ResNet-18 model to complete the GTSBR dataset classification task. It has 18 layers in total, which allows it to maintain high accuracy as well as reduce computational complexity to some extent. Compared to other deeper networks, ResNet-18 can effectively reduce training time and obtain great classification performance with its optimized residual modules. After construction, the model contains 11,198,571 parameters in total, and only 2,142,251 parameters can be trained.

2.4. MobileNet

MobileNet model is a series of lightweight neural proposed by Google, by reducing the number of parameters and computations to make it suitable for resource-constrained mobile devices[10]. MobileNetV1 was proposed in 2017, introducing depthwise separable convolution first and significantly reducing computational overhead. In 2018, MobileNetV2 introduced the inverted residual block and linear bottleneck, further optimizing feature extraction efficiency and achieving a better balance between performance and efficiency across multiple tasks. In 2019, the latest MobileNet model, MobileNetV3, combined the neural architecture search (NAS) and Squeeze-and-Excitation (SE) modules and was designed into two versions: MobileNetV3-Small and MobileNetV3-Large. The small version is suitable for scenarios with low computing needs while the large version is suitable for scenarios with high performance requirements, and they are both widely applied in tasks such as image classification, semantic segmentation and so on.

The paper chooses the MobileNetV2 model to complete the GTSBR dataset classification task. Its design continues the lightweight idea of MobileNetV1 but retains a certain capacity to ensure accuracy and real-time performance on edge devices when reducing the computation. The inverted residual block enhances the effectiveness of information propagation through the network, while the linear bottleneck effectively reduces non-linear computations, thereby improving both computational efficiency and speed, making it well-suited for an environment with limited computational resources. After construction, the model contains 3,613,803 parameters in total, and only 1,355,819 parameters can be trained.

3. Experimental Results and Discussion

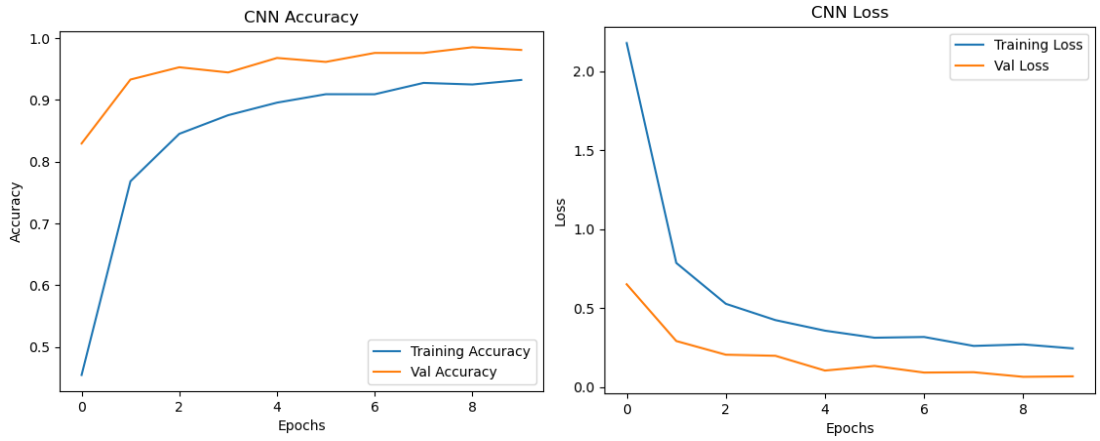
The experiment is conducted using the three models mentioned above, with Python 3 as the programming language and the TensorFlow-backed Keras 2.0 GPU version as the model framework. The development and execution environment for the experiment is Jupyter Notebook.

The experimental process includes the following parts: training the models with the training set, visualizing the training process, testing the models with the test set and observing the test results. These three models all use cross entropy as the loss function, the Adam optimizer for training and accuracy as the evaluation metric.

3.1. Model Training Process

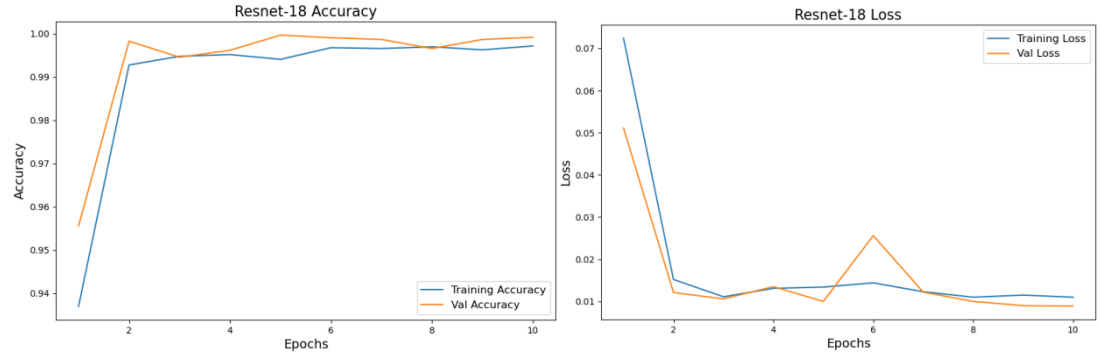
After training, the training process is visualized by plotting the changes in training accuracy, validation accuracy, training loss, and validation loss as the training epochs increase, as shown in Figures 3, 4 and 5.

DOI: 10.54254/2755-2721/125/2025.21282



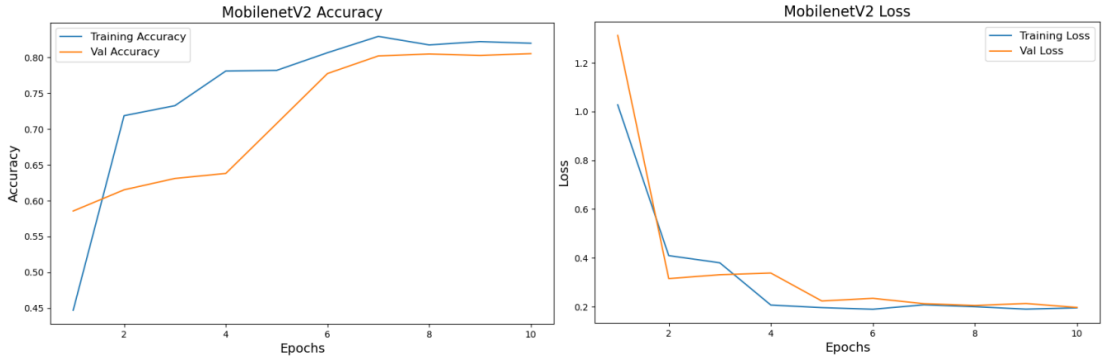
(a) (b)

Figure 3: Accuracy and Loss Trends of the customized CNN model. (a)is the accuracy trends, (b) is the loss trends (Picture credit: Original).



(a) (b)

Figure 4: Accuracy and Loss Trends of the ResNet-18 model. (a)is the accuracy trends, (b) is the loss trends (Picture credit: Original).



(a) (b)

Figure 5: Accuracy and Loss Trends of the MobileNetV2 model. (a) is the accuracy trends, (b) is the loss trends (Picture credit: Original).

As shown in Figure 3, from the customized CNN model accuracy curve, it can be observed that both training accuracy and validation accuracy increase rapidly in the initial phase, then stabilize, with the final training accuracy approaching 0.92 and the validation accuracy around 0.98. This indicates that the model performs well on the training set and exhibits good generalization ability on

DOI: 10.54254/2755-2721/125/2025.21282

the validation set. From the customized CNN model loss curve, it can be seen that both the training loss and validation loss decrease rapidly in the early stages and then stabilize. The validation loss is lower than the training loss, suggesting that the model does not exhibit significant overfitting.

For the ResNet-18 model, as shown in Figure 4, the accuracy curve indicates that after stabilizing, both training accuracy and validation accuracy are very high, almost reaching 1.0. This suggests that the model demonstrates excellent classification performance on both the training and validation sets. However, although the loss curves for ResNet-18 decrease rapidly in the first two epochs, a spike in the validation loss occurs around the 6th epoch during training, which may indicate a temporary decline in the ability to generalize to the validation set during certain stages of training. Nevertheless, the loss eventually converges to around 0.01, maintaining a generally low loss value.

For the MobileNetV2 model, as shown in Figure 5, its training accuracy and validation accuracy increase rapidly in the initial stages and stabilize around the 8th epoch, with the final training accuracy approaching 0.82 and the validation accuracy around 0.8. Its performance on both the training and validation sets is slightly inferior to that of the customized CNN and ResNet-18 models. From the MobileNetV2 loss curve, it can be seen that both training loss and validation loss decrease rapidly in the first few epochs, stabilizing after the 5th epoch, with almost no gap between them and without any significant overfitting.

In comparison, all three models converge during training. ResNet-18 achieves the best performance in terms of accuracy, but its validation loss fluctuations need further attention. The customized CNN model performs balanced in both training and validation accuracy, with stable loss. However, MobileNetV2 shows limitations as its performance with regard to both accuracy and loss is inferior to the previous two models.

3.2. Comparison of Model Test Results

The test results of the three models are shown in Table 1.

Table 1: Test results.

Model	Test Accuracy	Test Time
CNN	95.35%	2s
ResNet-18	96.19%	13s
MobileNetV2	87.77%	8s

According to the information in Table 1, the ResNet-18 model achieved the highest accuracy on the test set, reaching 96.19%, which indicates its strong capacity in the image classification task. Despite some fluctuation in the validation loss, it is still able to provide relatively accurate predictions. However, the longer test time (13 seconds) reflects its large amount of computation in parameter calculation. It may create computational bottlenecks in real-time applications.

In comparison, the customized CNN model's accuracy is 95.35%, which is a little less than that of the ResNet-18 model. However, it shows a clear advantage in test time, requiring only 2 seconds. That is likely due to the smaller number of layers and training parameters in the model, resulting in lower computational cost, which enables fast inference without sacrificing much accuracy. At the same time, its training and validation loss remain relatively stable, making it a promising model.

For the MobileNetV2 model, it has the lowest accuracy of 87.77%. That means, its feature extraction capability is inferior to the other two models when processing the images in the GTSBR dataset, so it fails to fully capture the features of complex traffic signs, resulting in a significantly lower accuracy. Despite having a slightly faster test time than the ResNet-18 model, its overall performance in this task is comparatively weaker.

DOI: 10.54254/2755-2721/125/2025.21282

In general, although the ResNet-18 model has a larger computational overhead, it performs the best in terms of accuracy; because the customized CNN model better balances speed and accuracy, it can be used in situations where speed is a higher priority; but the poorer performance in accuracy of MobileNetV2 suggests that it is less suitable for this task compared to the other two models.

4. Conclusion

The paper aims to probe the feasibility of achieving accurate traffic sign recognition within a limited time and constructs three CNN models, training and testing them on the GTSRB dataset. The experiment shows that the model can ensure high accuracy as well as achieve a high level of computational efficiency by choosing appropriate model architectures to cater to the real application requirements. Specifically, the ResNet-18 model has the best performance on test accuracy, but it requires the most time, and the customized CNN model demonstrates a good balance between test speed and stability. Compared to the other two models, the MobileNetV2 model has no advantage in terms of accuracy and runtime, although it is capable of completing the task under resource-constrained conditions. However, the research still has some limitations: the experiment only used the GTSRB dataset, lacking data validation from other real-world scenarios; at the same time, some models required excessive training time, which needs to be further optimized to better adapt to real-time applications. In the future, the research will explore a wider variety of network architectures and strive to achieve a better balance between model inference speed and recognition accuracy to meet the different requirements of various hardware platforms. This research will promote the advances in autonomous driving technologies and the applications of traffic sign recognition in intelligent transportation systems, which is of great practical significance and has a wide range of utilization prospects.

References

[1] *Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications, 6, 100134.*

[2] *O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., ... & Walsh, J. (2020). Deep learning vs. traditional computer vision. In Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1 (pp. 128-144). Springer International Publishing.*

[3] *Hussain, M., Bird, J. J., & Faria, D. R. (2019). A study on CNN transfer learning for image classification. In Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, (pp. 191-202).*

[4] *Zhu, Y., & Yan, W. Q. (2022). Traffic sign recognition based on deep learning. Multimedia Tools and Applications, 81(13), 17779-17791.*

[5] *Chen, J., Jia, K., Chen, W., Lv, Z., & Zhang, R. (2022). A real-time and high-precision method for small traffic-signs recognition. Neural Computing and Applications, 34(3), 2233-2245.*

[6] *Radu, M. D., Costea, I. M., & Stan, V. A. (2020). Automatic traffic sign recognition artificial inteligence-deep learning algorithm. In 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (pp. 1-4).*

[7] *Haque, W. A., Arefin, S., Shihavuddin, A. S. M., & Hasan, M. A. (2021). DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. Expert Systems with Applications, 168, 114481.*

[8] *Alawaji, K., Hedjar, R., & Zuair, M. (2024). Traffic Sign Recognition Using Multi-Task Deep Learning for Self-Driving Vehicles. Sensors, 24(11), 3282.*

[9] *He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).*

[10] *Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017. arXiv preprint arXiv:1704.04861.*