A Comparative Analysis of CNN Models and Data Augmentation for Optimizing Traffic Sign Recognition

Ruizhe Zhu

Department of Computer Science, Purdue University, West Lafayette, IN, 47906, USA

zhu1023@purdue.edu

Abstract. Correctly and efficiently recognizing traffic signs plays a crucial role in autonomous driving. his capability ensures that self-driving cars can obey traffic rules, respond to signals in real-time, and make appropriate decisions to avoid accidents and ensure the safety of all road users. However, the highly complex real-world settings still make this task challenging. This research aims to explore and evaluate multiple models and data augmentation strategies to find the optimal balance between performance and computational cost. Specifically, this research utilized the Traffic Sign Recognition Benchmark dataset and compared ResNet50, MobileNetV2, and a self-defined convolutional neural network model to evaluate their test accuracy and the total number of parameters used. By analyzing the results, the ResNet50 model achieved the highest accuracy. Meanwhile, MobileNetV2 demonstrated greater robustness, generalizability, and lower computational cost, while still maintaining a reliably high accuracy. Among all the data augmentation methods tested, adjusting brightness and contrast outperformed the image data generator and color jitter in almost all models and input dimensions.

Keywords: Traffic sign recognition; convolutional neural network; data augmentation.

1. Introduction

With the rapid development of autonomous driving technology, the recognition of road signs has become increasingly important [1,2]. Road signs provide crucial information that significantly affects the decision-making process during driving in upcoming road segments. The ability to automatically, correctly, and efficiently recognize traffic signs in real-time is therefore vital for a reliable autonomous driving system.

Despite the pivotal role of traffic sign recognition, consistently and successfully recognizing traffic signs in real-world conditions, where the environment is highly complex, is challenging. Various unideal situations introduce significant noise into actual recognition task [3,4]. Additionally, the computational requirements involved also challenge the trained models to be deployed effectively in the real world. Developing a model that is highly accurate, robust, and computationally efficient is of paramount importance. This is also the motivation behind this research.

To address the challenges and demands discussed above, a convolutional neural network (CNN) is designed for this purpose [5]. Transfer learning is also introduced to fully leverage the advantages of CNNs in feature extraction, reduce computational costs, and explore the robustness of different models [6].

This research constructs, trains, and analyzes CNNs and transfer learning models. The goal is to find the best model and data augmentation method that balances high accuracy with computational cost, ensuring good performance when applied to real-world situations. By comparing the results yielded by different models and data augmentation methods, and analyzing their advantages and limitations, this work could determine which model and data augmentation strategy provide the best outcomes, as well as the model that offers the best balance for practical application.

2. Method

2.1. Dataset

The dataset used in the experiment is the "Traffic Sign Dataset - Classification," available on Kaggle [7]. It contains 58 classes, including various speed limits, No Entry, Stop Sign, Danger Ahead, School Zone, and more. The images in the dataset feature different backgrounds such as buildings, plants, and sky, which can effectively reduce the model's reliance on certain backgrounds and diminish the potential bias caused by background uniformity. Additionally, the images exhibit a range of different lighting conditions, pixel qualities, clarities, and shades of various objects. These factors significantly enhance the model's robustness when dealing with the unpredictable and constantly changing realities of the real world.

2.2. Models

In this article, the CNN model mentioned above that is not using any transfer learning will be addressed as CNN for later convenience. Similarly, The MobileNetV2 model and ResNet50 model will be abbreviated to be called MobileNet and ResNet respectively.

2.2.1 CNN

In this experiment, Convolutional Neural Network is employed to process the input image of traffic signs to achieve the task of recognizing. The main structure of the CNN model is as follows: (1) Convolutional Layer and Pooling Layer. This model consists of 2 convolutional modules, each module includes 2 convolutional layers, and a max-pooling layer. The first convolutional layer uses 60 filters sized 5×5 in order to capture patterns and features of relatively bigger size. The second convolutional layer uses 30 filter of size 3×3 so that the smaller features can also be noticed. Each convolutional layer is followed by a 2×2 max-pooling layer to ensure to shrink the size of the features so that the calculation requirement is lower. (2) Dropout Layer. Two dropout layers are introduced between the convolutional module and fully connected layer for the purpose of increasing the generalizability and preventing the model from overfitting the dataset. The dropout rate is set to be 0.5, which means that randomly half of the neurons are discarded, so that the model does not overlearn and is less affected by potential bias in the dataset. The same dropout layer is also applied to the transferred pretrained model that will be introduced below. (3) Fully Connected Layer and the Output Layer. In the Fully Connected Layer, a Flatten layer is used to flatten the feature map. After that, the flatten feature map is connected to a fully connected layer of 500 nodes, which takes ReLU function as the activation function to ensure the robustness of the model while having a low computational demand. Lastly, the model uses a fully connected SoftMax output layer to explicitly have outputs to their corresponding class [8].

2.2.2 MobileNetV2

MobileNetV2 is used as a lightweight pretrained CNN model to transfer onto the traffic sign recognition task [9]. Different from the CNN that employed above, in the MobileNetV2 model, the pooling method of global average pooling in used so that the all the convolution features will be compressed into a 1-dimensional vector to be inputted into the fully connected layer which has 256 nodes that uses activation function ReLU same as the CNN above. SoftMax activated one hot encoding output layer is used as above. The same structured output layer is also what is used in ResNet50 Model that will be introduced below.

2.2.3 ResNet50

ResNet50 model is another pretrained CNN model that is deployed. Unlike the lightweight MobileNetV2 model, ResNet50 takes much more parameters, which means this is slower and requires much higher computation [10]. The same pooling method as MobileNetV2 model introduced is used in order to control the variable and add to more comparability to both transfer learning models.

2.3. Evaluation Metrics

The purpose of this experiment is to find a model and data augmentation method that fits the traffic sign recognition task at a balance of high Accuracy to safely apply that onto real-world scenarios and low computational cost to make the real time computation feasible when applied. Therefore, the following parameters are chosen to evaluate the performance of models.

Test Accuracy is used as the parameter to evaluate the reliability of the model. This is because that test accuracy can more intuitively and accurately measure the prediction on the unseen scenarios. Compared to other potential candidates to be evaluation scores such as F1-score, recall and precision scores, test accuracy gives a straightforward overall assessment of the model as the traffic signs recognition is a multi-class classification task. A desired data augmentation method should constantly outperform other methods in this scope.

To measure the model Complexity, the total numbers of parameters are recorded and analyzed. The total number of parameters directly reflects the computational cost and memory usage. The lower the total number of parameters is, the lower the demand for computational resources is.

By combining these two parameters, this work is able to compare the performances in between models. An ideal model should have relatively smaller size of total input parameters on the basis of ensuring a reliable test accuracy.

3. Experiments and Results

3.1. Training Details

In this experiment, in order to ensure that all variables that are not to be compared remain constant, the same parameters are applied to all models. A batch size of 16 is used. The Epoch number is fixed to be 20 with 100 steps in each epoch. The learning rate selected is either 0.001 or 1e-5 depending on the model configuration, in both cases, Adam optimizer is applied. The convolution is constructed in the way that the first convolutional block with two layers uses 60 filters of size (5×5); while the second convolutional block with two layers uses 30 filters of size (3×3). Size 2x2 is chosen for the size of pool. To avoid the Neural Networks from overfitting the data, 0.5 dropout rate is applied. For data augmentation, 3 different approaches are used, namely image data generator(imageGen), adjusting brightness contrast (brightContrast), and color jitter(colorJitter).

3.2. Performance Comparison

The purpose of the experiment shown in Table 1 is to compare the performances among different models. All models are trained with the input size 224×224×3 for 20 epoch base model trainable with 20 epoch and 0.001 learning rate.

Table 1. Performance comparison of different models and data augmentation methods.

	ResNet	MobileNet	CNN
imageGen	0.988	0.973	0.851
brightContrast	0.998	0.988	0.985
colorJitter	0.995	0.980	0.983

By comparing the accuracies across different models, a clear pattern emerges: among the pretrained models used for transfer learning, the ResNet model yields the highest accuracy in all three data augmentation approaches, while the MobileNet model yields the second-highest performance. The custom convolutional neural network does not perform as well as the pretrained models. However, for the data augmentation approaches of adjusting brightness and contrast and applying color jitter, the performance differences between the convolutional neural network and MobileNet are minimal.

Although all models show good performance, the ResNet model stands out as exceptionally well-performing across all three data augmentation approaches. It is appropriate to conclude that ResNet is the model with the best performance for this task.

Table 2 demonstrates the experiments exploring which data augmentation approach yields higher performance. All models were trained with an input size of 224×224×3 for 20 epochs, with the base model trainable for another 20 epochs at a learning rate of 1e-5. In the table, a model tagged with "True" indicates that the base model is set to be trainable, while "False" denotes that the base model is locked.

It could be observed that the approach of adjusting brightness and contrast is notably the best among all models. The second-best data augmentation approach is color jitter, while the image data generator performs the worst. A conclusion can be drawn that adjusting brightness and contrast is the most suitable approach for enhancing the generalizability of the traffic sign recognition model.

Table 2. Performance comparison of different model settings data augmentation methods.

Model	Trainable	imageGen	brightContrast	colorJitter
ResNet	False	0.755	0.799	0.772
ResNet	True	0.988	0.998	0.995
MobileNet	False	0.981	0.986	0.983
MobileNet	True	0.974	0.980	0.983
CNN	True	0.995	0.980	0.983

In the experiment shown in Table 3, the learning rate is set to 0.001 with 20 epochs. The only variables are the data augmentation methods and the input sizes. Models with different input size of 100×100×3 and 224×224×3 are compared in this experiment. Each model has the base model locked in order to explore into the generalizability for the pretrained model transferring onto the task of traffic sign recognition.

Table 3. The sensitivity comparison of pre-trained models to input size.

	Resnet 100x100x3	Resnet 224x224x3	Δ	MobileNet 100x100x3	MobileNet 224x224x3	Δ
imageGen	0.185	0.755	0.571	0.938	0.981	0.043
brightContrast	0.249	0.799	0.549	0.983	0.986	0.002
colorJitter	0.242	0.772	0.530	0.983	0.983	0.005

It could be observed that with 2 different input sizes, ResNet models all exhibited significant changes. With the input size of 100×100×3, the pretrained model ResNet yields suboptimal test accuracies that ranges from 0.1847 to 02494. While switching to the input size of 224×224×3, the performance of ResNet improves significantly with achieving the test accuracies up to 0.7986. The changes in performance for the two different input sizes are all greater than 0.5 in terms of test accuracy. This indicates a significant sensitivity to the size of input image.

On the other hand, the pretrained model MobileNet yields a significantly higher and more consistent performance on both input sizes. The test accuracy constantly exceeds 0.9 and is mostly greater than 0.98. For different input sizes, the differences in performance are all negligibly small. It can be concluded that MobileNet model is not noticeably sensitive to the input size of the image. This demonstrates that MobileNet model has an enhanced resilience against variable input and has greater robustness in handling various situations.

Meanwhile, it could be observed that MobileNet model yields substantially better test accuracies than ResNet model under all sets of identical conditions as long as the base model is locked. This shows that as a pretrained model, MobileNet has higher generalizability and adaptability to transfer onto traffic sign recognition tasks and are more consistent with different sizes of input. In both dimensions of accuracy and consistency to variables, MobileNet model outperforms ResNet model.

In experiment, demonstrated in Table 4, for the purpose of comparing the best performance that the pretrained models can yield, all models are trained with input size of 224×224×3, base model trainable with 20 epoch with 0.001 learning rate

In all 3 data augmentation approaches, the ResNet models and MobileNet models are trained to compare their test accuracies and numbers of total parameters. Every approach exhibits that MobileNet yields test accuracies approximately 0.01 lower than the ResNet model. However, the MobileNet employs 21526336 less parameters than ResNet does, namely about 9.277 times less. This represents a considerable improvement in terms computational efficiency.

Table 4. Comparison of model accuracy and the number of parameters.

Model	Data Augmentation	Test Accuracy	Total Parameter
ResNet	imageGen	0.988	24127162
MobileNet	imageGen	0.974	2600826
Δ	imageGen	-0.014	-21526336
ResNet	brightContrast	0.998	24127162
MobileNet	brightContrast	0.988	2600826
Δ	brightContrast	-0.010	-21526336
ResNet	colorJitter	0.995	24127162
MobileNet	colorJitter	0.983	2600826
Δ	colorJitter	-0.012	-21526336

As illustrated in Table 3, MobileNet model has higher robustness and generalizability and adaptability to traffic sign recognition tasks. Furthermore, Table 4 reaffirms MobileNet model's computational efficiency with substantially fewer parameters, while the test accuracy being only negligibly lower than ResNet model. All these enhances MobileNet model's capability to the task of traffic sign recognition and leads to the reasonable conclusion that MobileNet is the more suitable pretrained model for transfer learning for this task.

4. Discussion

One of the dominant limitations is the lack of special conditions that may be present in real-world scenarios. Although the dataset covers various lighting conditions and pixel qualities, the absence of reflections, occlusions, and extreme weather conditions can make deploying the model in the real world still face ambiguities when encountering such challenging situations. Moreover, the dataset lacks scenarios with multiple overlapping images, which often appear in real-world settings. For instance, the model has not learned scenarios where road signs are partially obscured by other objects or overlap with other signs. The absence of these real-world settings may constrain the model's generalization ability for various real-world tasks.

Additionally, while the ResNet model yields the highest accuracy, its computational cost is too high, which restricts its deployment in real-time applications. On the other hand, the MobileNet model is a practical alternative that demands much less computation. However, given that there are still missing scenarios in the dataset that have not been learned or tested, MobileNet may not always provide the optimal balance between accuracy and computational efficiency for all scenarios. There is no guarantee that the MobileNet model can be generalized to scenarios such as reflections, potential blockages, or overlaps with high robustness and consistent performance.

The future work should aim on improving the dataset to include more complex scenarios that covers what is discussed above. This would help enabling a more comprehensive and robust ability to recognize road signs in more complex and realistic situations.

Furthermore, future work should focus on evaluating the different models' performances on the improved dataset to explore the balances of accuracy and computation demands under more realistic and comprehensive conditions. Building on that, in addition, explore optimization in model parameters to reduce computational cost on all input conditions to find an optimum model architecture that balances computational efficacy and accuracy.

5. Conclusion

In this research, by analyzing the CNN model, ResNet50, and MobileNetV2 with three different data augmentation methods, an evaluation of the three models was conducted on the dimensions of test accuracy and computational cost. The ResNet50 model yields the highest accuracy but lacks robustness and is very sensitive to input sizes. The MobileNetV2 model, on the other hand, has greater robustness, generalizability, and lower computational cost, with the difference in accuracy compared to the ResNet50 model being negligibly small. Therefore, MobileNetV2 strikes a good balance between accuracy and computational efficiency. Adjusting Brightness Contrast is the best data augmentation method, outperforming the image data generator and color jitter in almost all models and across all input dimensions.

References

[1] Bousarhane, Btissam, Saloua Bensiali, Driss Bouzidi. Road signs recognition: state-of-the-art and perspectives. International Journal of Data Analysis Techniques and Strategies, 2021, 13(1-2): 128-150.

[2] Greenhalgh Jack, Majid Mirmehdi. Real-time detection and recognition of road traffic signs. IEEE transactions on intelligent transportation systems, 2012, 13(4): 1498-1506.

[3] Houben Sebastian, Stallkamp Johannes, Salmen Jan, et al. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. The 2013 international joint conference on neural networks, 2013: 1-8.

[4] Zhu Zhe, Liang Dun, Zhang Songhai, et al. Traffic-sign detection and classification in the wild. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2110-2118.

[5] Li Zewen, Liu Fan, Yang Wenjie, et al. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 2021, 33(12): 6999-7019.

[6] Gu Jiuxiang, Wang Zhenhua, Kuen Jason, et al. Recent advances in convolutional neural networks. Pattern recognition, 2018, 77: 354-377.

[7] Traffic Sign Dataset – Classification. Kaggle. URL: https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification. Last Accessed: 2024/10/25

[8] Wu Jianxin. Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 2017, 5(23): 495.

[9] Sandler Mark, Howard Andrew, Zhu Menglong, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[10] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.