Optimizing Traffic Sign Recognition: Custom CNN vs. Pretrained Models

Onkar Kunte
Yeshiva University
Okunte@mail.yu.edu

Shashidhar Reddy
Yeshiva University
Sainala@mail.yu.edu

Abstract

The rapid improvement of computer vision and automotive technologies has made autonomous vehicles a hot issue in recent years. Accurately identifying traffic signs is a critical component in autonomous vehicles' capacity to operate safely and effectively. For this reason, one essential element of autonomous driving systems is the identification of traffic signs. Researchers have been investigating a number of methods, including as machine learning and deep learning, for traffic sign identification in an effort to overcome this difficulty. Despite these initiatives, developing dependable traffic sign identification systems continues to face substantial obstacles due to the variety of traffic signs in various geographic locations, intricate background backgrounds, and variations in illumination.This study offers a thorough overview of the most recent developments in the field of traffic sign identification, including a number of important topics, such as datasets, feature extraction strategies, preprocessing techniques, classification strategies, and performance evaluation. The typically utilized datasets for traffic sign recognition and the difficulties they provide are also covered in the article. This work also clarifies the constraints and directions for future traffic sign recognition research.

Keywords: deep learning, machine learning, and traffic sign identification

1. Introduction

This work aims to investigate the most recent advances in deep learning algorithms for traffic sign identification. The dependability of traffic sign recognition algorithms is becoming more and more crucial for guaranteeing the safety of all road users as the demand for autonomous vehicles rises. Vehicles can now read and comprehend crucial road signs, like speed limit, hazard, and turn ahead signals, thanks to traffic sign recognition technology. Not only does this technology increase driver safety, but it also makes the road a safer place for all users by reminding users of crucial rules and delivering vital information. This study undertakes a thorough analysis of previous work on traffic sign identification in order to fully understand the state-of-the-art work in this subject.

The main contributions of this paper are as follows: A comprehensive review of state-of-the-art traffic sign recognition work, categorizing studies into conventional machine learning and deep learning approaches. A discussion of widely adopted traffic sign recognition datasets, their challenges, and limitations, as well as the future research prospects in this field.

2. Related work

The field of traffic sign recognition has seen a significant amount of research in recent years, particularly regarding the use of machine learning techniques to classify traffic signs accurately. In a study by Kerim and Efe (2021) [1], an Artificial Neural Network (ANN) was developed to incorporate various features, including Histograms of Oriented Gradients (HOG) and a combination of color, HOG, and Local Binary Patterns (LBP). This hybrid ANN was made up of 9 individual ANNs, each responsible for analyzing traffic signs based on a set of attributes present in the images. The authors used data augmentation techniques such as translation, rotation, and noising to improve the performance of the model. The results showed that the method combining color, HOG, and LBP features achieved an accuracy level of 95%, significantly outperforming the method using HOG features alone.

Another study by Soni et al., (2019) [2] used HOG and LBP descriptors with the Principal Component Analysis (PCA) and Support Vector Machines (SVM) to classify traffic signs. The study used the Chinese Traffic Sign Database (TSRD) with 58 classes and 6164 images, and the best performing method was the LBP with the PCA and SVM classifiers, achieving an accuracy level of 84.44%. In Namyang and Phimoltares (2020) [3], a combination of the Support Vector Machines (SVM) and Random Forest algorithms was used with HOG and the Color Layout Descriptor (CLD) to classify traffic signs. The authors collected a dataset of 408 training images and 216 testing images, consisting of 4 classes of traffic signs, namely regulatory, warning, construction, and guide signs. The images were first preprocessed to resize them to 120 × 80 pixels. The first stage of the method used HOG features with SVM and a radial basis function (RBF) kernel to classify regulatory signs. The construction class was then classified with SVM, while the warning and guidance signs were classified in the next stage using a hierarchical classification model with HOG and CLD. The method achieved an accuracy level of 93.98%.

Li et al., (2022) [4] presented an approach for traffic sign recognition with finely crafted features and dimension reduction. The authors utilized the color information of traffic signs and enhanced the discrimination between images using the improved color-histogram-based feature. Subsequently, the PCA algorithm was adopted to reduce the dimensions of the improved color-histogram-based feature, which increased the running speed of the method. Lastly, the expression ability of features was further enhanced by concatenating the improved color-histogram-based feature after dimensionality reduction with the HOG feature of images. The experimental results recorded an accuracy level of 99.99% on the German Traffic Sign Recognition Benchmark (GTSRB) dataset.

The paper by Madani and Yusof (2018) [5] presented a traffic sign recognition technique based on three key components: border color, shape, and pictogram information. The proposed technique consists of three independent stages: Firstly, the border colors are extracted using an adaptive image segmentation technique based on learning vector quantization. Secondly, the shape of the traffic sign is detected using a fast and simple matching technique based on the logical exclusive OR operator. Lastly, the pictogram is extracted and classified using a SVM classifier model. The proposed technique was tested on the German traffic sign recognition benchmark, achieving an overall recognition rate of 98.23%.

Sapijaszko et al., (2019) [6] proposed a traffic sign recognition system that comprises normalization, feature extraction, compression, and classification stages. The images are normalized using gray scaling and anisotropic diffusion techniques. The discrete wavelet transform and discrete cosine transform extract essential features from the images while reducing their size. Finally, a three-layer feedforward multilayer perceptron is used for analysis and classification. The best algorithms achieved a recognition accuracy of 96.0% on the Belgian Traffic Sign dataset (BTSD), 95.7% on the GTSRB, and 94.9% on the TSRD.

Aziz and Youssef (2018) [7] proposed a traffic sign recognition system that leverages feature extraction and the Extreme Learning Machine (ELM) algorithm. The authors evaluated three feature extraction techniques, namely HOG, Compound Local Binary Patterns (CLBP), and Gabor features, and passed the extracted features into ELM for classification. ELM operates on the assumption that learning models can be fed by randomly selected input weights without requiring specific distribution adjustment. The authors tested their proposed method on two datasets, the GTSRB and the BTSD, and achieved high accuracy rates of 99.10% and 98.30%, respectively.

Weng and Chiu (2018) [8] presented a traffic sign recognition that was divided into two stages. In the detection stage, potential traffic signs were detected using the Normalized RGB color transform and Single-Pass Connected Component Labeling (CCL). In the second stage, HOG was used to generate the descriptors of the signs, which were then classified using the SVM. The proposed method achieved a recognition rate of 90.85% when tested with the GTSDB dataset.

Wang (2022) [9] proposed a traffic sign classification system using three machine learning classifiers: Logistic Regression (LR), Multilayer Perceptron (MLP), and SVM. The authors used the Multinomial Logistic Regression classifier, which is a variation of LR that generates a probability distribution indicating the likelihood of each class. They applied the Softmax function to transfer the weighted sum of characteristics into a probability distribution. For MLP, the authors used a biological neuron model to determine its structure and the activation function. For SVM, the authors used the one-vs.-the-rest method with the LinearSVC algorithm. The authors conducted experiments on the GTSRB dataset and achieved accuracy rates of 97.75% for LR, 98.88% for MLP, and 95.51% for SVM.

In a recent study, Zhu and Yan (2022) [12] tackled the problem of traffic sign recognition using two deep learning methods: You Only Look Once (YOLO)v5 and the Single Shot MultiBox Detector (SSD). YOLOv5 is a real-time object recognition algorithm that processes the entire image with a single neural network and divides it into parts to estimate the bounding boxes and probabilities for each part. The SSD, on the other hand, accelerates the process by eliminating the need for region proposal networks for each component. The authors collected a dataset of 2182 traffic sign images from 8 different classes, which was split as follows: 64% training set, 16% validation set, and 20% testing set. The models were trained using data augmentation techniques, such as rotation and resizing. The proposed method achieved an accuracy of 97.70% for YOLOv5 and 90.14% for SSD, demonstrating the effectiveness of the proposed approach in terms of its accuracy.

Li et al., (2019) [11] proposed a CNN-based solution for traffic sign recognition. The proposed CNN architecture included 2 convolutional layers, 6 max pooling layers, and 4 traffic sign modules aimed at extracting features from the images. The authors conducted experiments using two datasets, GTSRB and BTSD, with over 50,000 and 7000 images, respectively. The hyperparameters, such as a learning rate of 0.001, gamma set of 0.1, and step values of 24,000 and 48,000 for 60,000 iterations, were set in the experiments. The proposed method achieved an accuracy of 97.4% on the GTSRB dataset and 98.1% on the BTSD dataset.

Yazdan and Varshosaz (2021) [16] presented a novel approach for traffic sign recognition that leverages a minimal set of common images. The proposed method creates a new orthogonal image of the traffic sign and compares it to a database of single images shot in front of each sign, eliminating the need for multiple images in the training database. The orthogonal image is generated from stereo pictures and put through a template-matching procedure. This approach resulted in an accuracy of 93.1% for recognizing traffic signs.

Bangquan and Xiong (2019) [17] proposed a traffic sign recognition system using the Efficient Convolutional Neural Network (ENet), which combines two pretrained models, VGG16, and LeNet. The system was trained on the GTSRB dataset, which comprises 43 classes of traffic signs and was split into a training set of 39,209 and a test set of 12,630. The system was trained using the Adam optimizer with the softmax cross-entropy loss function. The experiment showed that the LeNet model performed better than the VGG16 model, with accuracy levels of 98.6% and 96.7% accuracy, respectively. ENet with the LeNet algorithm was slower and larger but more accurate, while ENet with the VGG16 algorithm was quicker and smaller but less precise. The system demonstrated excellent generalization skills by correctly classifying all images in a new dataset.

3. Traffic Sign Recognition Datasets

In the subject of traffic sign recognition, a number of datasets are frequently used to assess how well recognition algorithms function. Researchers are able to train and test their algorithms in realistic contexts thanks to these datasets, which offer a complete and diversified depiction of real-world scenarios and traffic signs.

3.1. German Traffic Sign Recognition Benchmark (GTSRB)

The German Traffic Sign Recognition Benchmark (GTSRB) is a well-established and widely utilized dataset in the field of traffic sign recognition. With a total of 51,839 high-resolution images covering 43 unique traffic sign classes, the GTSRB provides a comprehensive and reliable resource for evaluating the performance of traffic sign recognition algorithms. However, it is important to note that the GTSRB dataset primarily consists of German traffic signs, which may not accurately represent the diversity of traffic signs used in other regions. This limits the generalization of models trained on the GTSRB dataset and may result in a decreased performance when applied to other regions. Despite this limitation, the GTSRB dataset remains a popular resource due to its size, high-quality annotations, and real-world scenario representation, making it an excellent resource for researchers in the field of traffic sign recognition.

3.2. Belgium Traffic Sign Dataset (BTSD)

One of the most well-known datasets for traffic sign recognition is the Belgium Traffic Sign Dataset (BTSD) , which consists of more than 7095 high-resolution photos that represent 62 different traffic sign classes that may be found in Belgium and the Netherlands. The photographs were chosen with care from a variety of actual situations, offering a varied portrayal of occlusions, weather,[19] and lighting. Despite being smaller than other datasets in the area, the BTSD dataset is nevertheless a useful tool for academics, especially when it comes to confirming the accuracy of models that were trained on bigger datasets like GTSRB or JTSRB. The BTSD is a valuable resource for academics working on traffic sign identification algorithms in particular because of its diverse depiction of real-world settings and high-quality photos.

3.3. Chinese Traffic Sign Database (TSRD)

A comprehensive archive of photographs of traffic signs is the Chinese Traffic Sign archive (TSRD), which is supported by the National Nature Science Foundation of China (NSFC). The TSRD offers a rich and realistic dataset for training and assessing traffic sign identification algorithms, with a total of 6164 images split into 58 distinct classes. The photos in the TSRD were gathered from a variety of sources, such as BAIDU Street View and cameras positioned in actual environments[18]. This led to a wide variety of photos taken in varied circumstances, including various weather conditions, lighting conditions, and environments.Images of partially obscured signs are also included in the database, which offers a difficult scenario for evaluating the reliability of recognition algorithms and modeling real-world scenarios. Because of its accurate and varied representation of traffic signs, the TSRD is an invaluable tool for academics studying traffic sign identification.

The selection of a dataset to utilize will be contingent upon the particular demands and specifications of the recognition system under development, as each dataset possesses distinct advantages and disadvantages. When choosing the best dataset for a specific use case, factors like the size of the dataset, the range of traffic sign classes it represents, and the image quality all need to be taken into account.

Then we focused and finalized the German Traffic Sign Recognition Benchmark (GTSRB), a dataset that is integral to the field of automated traffic sign detection and recognition. Consisting of over 50,000 images spanning 43 classes, GTSRB provides a diverse set of traffic signs which in-

Table 1. Summary of Traffic Sign Datasets

Dataset	Total Images
GTSRB (German Traffic Sign Recognition Benchmark)	51,839
BTSD (Belgium Traffic Sign Dataset)	7,095
TSRD (Chinese Traffic Sign Database)	6,164

cludes variations in angle, illumination, and physical condition. This dataset serves as a substantial foundation for developing and benchmarking our traffic sign recognition algorithms, ensuring that the models are trained on high-quality, real-world data representative of the variability encountered in actual driving scenarios.

4. Methodology

4.1. Model Architecture

Our proposed model, named GTSRB MODEL, is a convolutional neural network (CNN) specifically designed for Traffic Sign Recognition. The architecture is carefully crafted to balance complexity and performance, making it suitable for real-time applications.

4.1.1 Layer Configuration

The model initiates with an input dimension defined by the user, adaptable to various image resolutions, and an output dimension corresponding to the number of traffic sign classes. The architecture comprises several layers, each with a specific function:

- **Convolutional Layers:** The core of the model consists of six convolutional layers. The first layer (conv1) takes a 3-channel input and increases the depth to 32 channels. Subsequent layers (conv2 to conv6) progressively increase the channel depth from 32 to 1024. These layers are pivotal in feature extraction, capturing various aspects of traffic signs from basic edges to complex shapes.

- **Batch Normalization:** Following the second, fourth, and sixth convolutional layers, batch normalization layers are employed. These layers standardize the inputs to the next layer, aiding in stabilizing and accelerating the training process.

- **Activation and Pooling:** The ReLU activation function is applied after each batch normalization, intro-

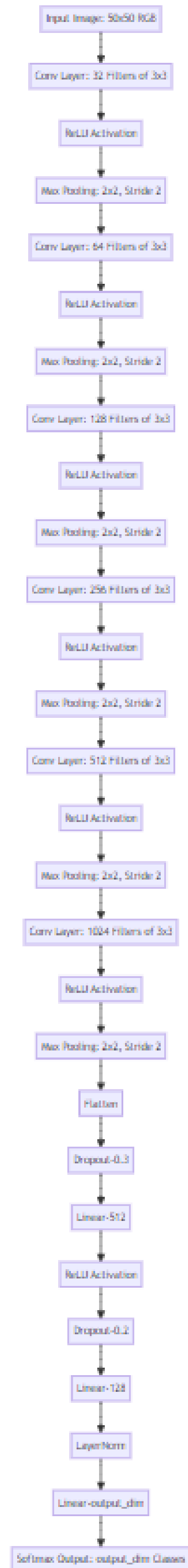


Figure 1. Neural network architecture of the GTSRB model.

ducing non-linearity. Max pooling is used after each set of two convolutional layers, reducing the spatial dimensions and thus, the computation required for deeper layers.

- **Dropout:** Dropout layers with rates of 0.3 and 0.2 are strategically placed in the network to prevent overfitting, ensuring the model's generalization to new, unseen data.

4.1.2 Flattening and Dense Layers

Post convolutional layers, the network flattens the output for the dense layers. The first dense layer (l1) reduces the feature vector to a size of 512. This is followed by a second dense layer (l2) that further compresses the representation to 128 dimensions. A final linear layer (l3) maps these features to the output classes.

4.1.3 Training and Evaluation Metrics

The model incorporates functions for both training and evaluation metrics:

- **Training Metrics:** During training, the model calculates accuracy and loss. Accuracy is measured as the ratio of correctly predicted instances to total instances, providing an intuitive measure of the model's performance.

- **Validation Metrics:** For validation, the model computes loss and accuracy on a separate dataset, ensuring the model's effectiveness on unseen data.

4.1.4 Training Procedure

The training procedure is an integral part of the model's design. It involves several key steps:

1. **Epochs and Batching:** The model is trained over a specified number of epochs, iterating through the training data in batches.

2. **Optimization:** A loss function and optimizer are defined, guiding the model's learning process. Backpropagation is employed to update the weights.

3. **Learning Rate Scheduling:** A learning rate scheduler adjusts the learning rate during training, optimizing the convergence speed and stability.

4. **Progress Tracking:** Throughout the training, progress is monitored and displayed, offering insights into the learning process.

4.1.5 Implementation Details

This subsection elaborates on the specific implementation details of the GTSRB MODEL, focusing on its instantiation, optimization, and loss computation for the training process.

4.1.6 Model Instantiation

The model is instantiated with specific parameters tailored to the task of traffic sign recognition. The input dimension is set to $3 \times 50 \times 50$ 3×50×50, accommodating for 50x50 pixel images with three color channels (RGB). The output dimension is set to 43, corresponding to the number of unique traffic sign classes in the dataset.

```
model = GTSRB_MODEL(35050, 43).to(device)
```

This line of code initializes the model and moves it to the appropriate computational device (like a GPU), ensuring efficient training.

4.1.7 Optimization Setup

The optimization of the model is a critical aspect of its training. We employ the Adam optimizer, known for its efficiency in handling sparse gradients and adaptive learning rates.

```
optimizer = Adam(params=model.parameters(), lr=0.0008)
```

The learning rate is set to 0.0008, a value determined empirically to provide a good balance between training speed and model convergence.

4.1.8 Learning Rate Scheduler

A learning rate scheduler is used to adjust the learning rate during training. This approach helps in fine-tuning the model and avoiding local minima.

```
lr_s = lr_scheduler.LinearLR(optimizer, start_factor=1.0,
end_factor=0.5, total_iters=10)
```

The scheduler linearly decreases the learning rate from its initial value to half across 10 iterations. This gradual reduction allows for finer adjustments as the model approaches optimal performance.

4.1.9 Loss Function

The choice of loss function is pivotal in guiding the training process. For our multi-class classification problem, the Cross-Entropy Loss is utilized.

```
loss = nn.CrossEntropyLoss()
```

This loss function is well-suited for classification tasks with multiple classes, as it calculates the difference between the predicted probability distribution and the actual distribution.

4.2. Utilization of Pretrained Models

In our research, we employ six different pretrained neural network models to address the challenge of Traffic Sign Recognition (TSR). These models, renowned for their efficacy in various image recognition tasks, are fine-tuned to our specific use case. Each model is briefly described in its respective subsection.

4.2.1 EfficientNetGTSRB

Model Description: The EfficientNetGTSRB is based on the EfficientNet architecture, particularly the EfficientNet-B0 version. Known for its balance between accuracy and computational efficiency, EfficientNet is a suitable choice for TSR.

Model Adaptation: The pretrained EfficientNet-B0 is fine-tuned by adjusting its final classifier layer to match the 43 traffic sign classes. This adaptation ensures that the model, initially trained on a broader range of classes, is now specifically tuned for TSR.

4.2.2 InceptionGTSRB

Model Description: The InceptionGTSRB model utilizes the Inception v3 architecture. This model is characterized by its use of multiple kernel sizes in the convolutional layers, enabling it to capture features at various scales.

Model Adaptation: Similar to EfficientNetGTSRB, the final fully connected layer of Inception v3 is modified to output 43 classes. Special attention is paid to the handling of auxiliary outputs during training, which is a unique feature of Inception v3.

4.2.3 ResNetGTSRB

Model Description: ResNetGTSRB is based on the ResNet-50 architecture. ResNet, or Residual Network, is known for its deep architecture and the use of skip connections, which help in alleviating the vanishing gradient problem in deep networks.

Model Adaptation: The ResNet-50 model, pretrained on a large image dataset, is adapted by altering its final layer to classify 43 traffic sign categories, making it apt for our TSR task.

4.2.4 VGGGTSRB

Model Description: The VGGGTSRB employs the VGG-16 model, famous for its simplicity and depth. VGG-16's architecture consists of sequential convolutional layers followed by fully connected layers.

Model Adaptation: The adaptation involves modifying the final fully connected layer of VGG-16 to output 43 classes, tailoring the model to recognize traffic signs.

4.2.5 DenseNetGTSRB

Model Description: DenseNetGTSRB is built upon the DenseNet-121 architecture. DenseNet is unique for its dense connections between layers, promoting feature reuse and reducing the number of parameters.

Model Adaptation: The pretrained DenseNet-121 model is fine-tuned by adjusting its classifier to output 43 classes, aligning it with the requirements of TSR.

4.2.6 MobileNetGTSRB

Model Description: MobileNetGTSRB utilizes the MobileNetV2 architecture, which is designed for mobile and edge devices. It is known for its lightweight structure and the use of depthwise separable convolutions.

Model Adaptation: The final classification layer of the pretrained MobileNetV2 is modified to classify 43 traffic sign types, making it suitable for our study in TSR.

5. Comparative Analysis and Results

This section provides a comparative analysis of various deep learning models employed in the study of Traffic Sign Recognition (TSR). The models were evaluated based on their testing accuracy after being trained for 20 epochs with a consistent batch size of 64. The purpose of this analysis is to determine the most effective model for TSR under the same computational constraints.

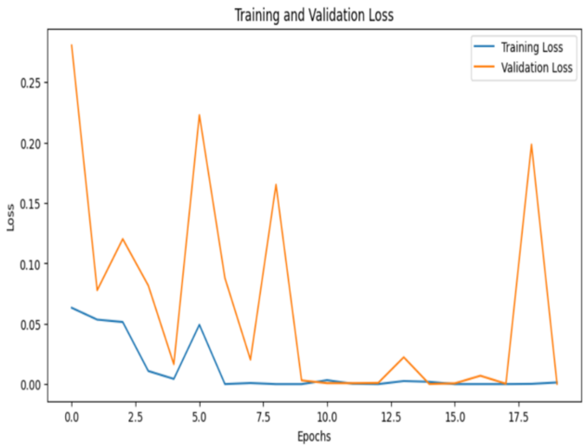
5.1. Model Performance

The our custom Convolutional Neural Network (CNN), our baseline model, achieved a remarkable testing accuracy, indicative of its robust feature extraction capabilities tailored specifically for TSR. In comparison, standard architectures such as ResNet50, EfficientNet, and Inception also demonstrated high accuracy, benefiting from their depth and complex structures that capture a broad range of features from traffic sign images.

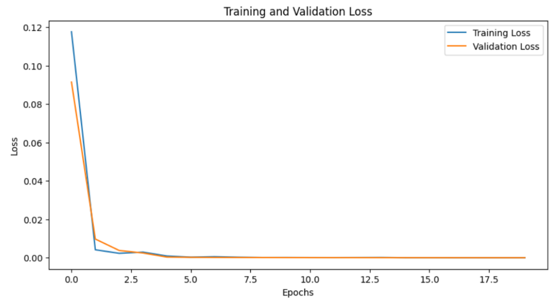
5.2. Testing Accuracy

The testing accuracy serves as the primary metric for comparison:

- The **custom CNN** model achieved the highest testing accuracy, which stands at approximately 99.17 99.17
- **DenseNet** followed closely with an accuracy of 98.06%, demonstrating the effectiveness of its densely connected layers.
- Both **EfficientNet** and **Inception** models performed similarly well, with accuracies of 97.82% and 97.90%, respectively.

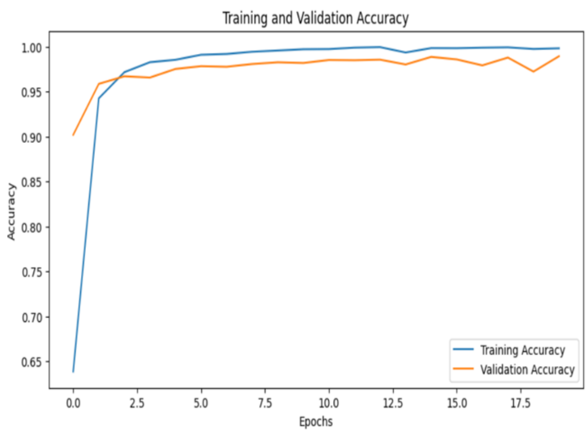


(a) Before Complex CNN Architecture

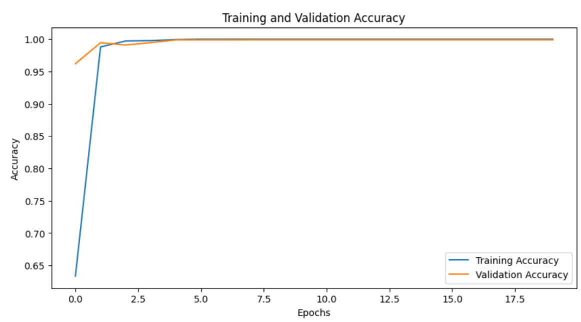


(b) After Complex CNN Architecture

Figure 2. Training and validation loss comparison before and after the application of a complex CNN architecture



(a) Before Complex CNN Architecture



(b) After Complex CNN Architecture

Figure 3. Training and validation accuracy comparison before and after the application of a complex CNN architecture

- The **ResNet50** model achieved an accuracy of 95.50%, which is slightly lower compared to the others but still significant.
- The **MobileNet** model, designed for mobile applications with constraints on computational resources, reached an accuracy of 97.43%.
- The **VGG** model, however, showed an accuracy of 5.94%, which is considerably lower than expected. This anomaly suggests a potential issue in either the model configuration or the evaluation process, necessitating further investigation.

5.3. Batch Size

All models were trained with a uniform batch size of 64 64, ensuring that the comparison remains fair with respect to the batch processing capabilities of each network.

Table 2. Comparative Performance of Deep Learning Models for Traffic Sign Recognition

Model	Epoch	Test Accuracy	BatchSize
CNN (Custom)	20	0.991	64
ResNet50	20	0.9550	64
EfficientNet	20	0.9782	64
Inception	20	0.9790	64
MobileNet	20	0.9743	64
DenseNet	20	0.9806	64
VGG	20	0.0594	64

6. Discussion

In our quest to advance the state of Traffic Sign Recognition (TSR), we initially employed a straightforward neural network, which we will refer to as the Simple GTSRB MODEL. This model was constructed with minimal layers, specifically two convolutional layers for feature extraction followed by max pooling and fully connected layers for classification. The intent was to create a baseline by which we could measure the improvement of more complex models.

The Simple GTSRB MODEL architecture is as follows:

- Two convolutional layers, with the first layer transforming the input channels to 32 channels and the second layer increasing this to 64 channels, each followed by a ReLU activation function.
- Max pooling to reduce spatial dimensions after each convolutional operation.
- Flattening of the feature maps into a vector form to be fed into the fully connected layers.
- Two fully connected layers, with the second one mapping to the output dimension representing the traffic sign classes.

While this model achieved a respectable level of performance, the quest for higher accuracy led us to experiment with a more complex architecture. The new GTSRB MODEL introduced additional layers, including batch normalization and dropout layers, to enhance learning and generalization capabilities. With this complex architecture, we witnessed a significant uplift in the model's ability to classify traffic signs accurately.

Upon training, we observed through our metrics that the complex GTSRB MODEL not only outperformed the simple model but also surpassed the accuracy of pretrained models, a testament to the effectiveness of custom architectures tailored to specific tasks. The complex GTSRB MODEL achieved a testing accuracy of 99.17% which was substantially higher than that of the pretrained models we compared against.

Figures 1 and 2 visually encapsulate the improvement brought by the complex model over the simple one. In Figure 1(a), we see the training and validation loss before the application of the complex CNN architecture, which indicates the initial performance of our Simple GTSRB MODEL. Figure 1(b) shows a marked improvement in loss reduction post the application of the complex architecture. Similarly, Figure 2(a) illustrates the training and validation accuracy prior to the architectural enhancements, and Figure 2(b) demonstrates the superior accuracy post-enhancements.

These visual representations corroborate our findings, showcasing that the enhanced model not only reduces overfitting as indicated by the closer convergence of training and validation loss—but also generalizes better, as evidenced by the consistent and high validation accuracy.

In the development of our neural network model for traffic sign recognition, we encountered the pervasive challenge of data imbalance, a common obstacle in machine learning that can significantly skew the performance of predictive models. Data imbalance occurs when the number of instances of certain classes significantly outnumbers others, leading to a model that may become biased towards the majority class and perform poorly on underrepresented classes.

To mitigate this issue, we implemented a series of data augmentation techniques aimed at synthesizing additional training data to create a more balanced distribution. The augmentation pipeline for training data included:

- **ColorJitter**: Modifying brightness, contrast, saturation, and hue to simulate varying lighting conditions and camera settings.
- **RandomEqualize**: Applying random histogram equalization to enhance image contrast.
- **AugMix**: Combining diverse augmentations in a stochastic manner to increase robustness.
- **RandomHorizontalFlip and RandomVerticalFlip**: Flipping images horizontally or vertically to simulate different orientations of traffic signs.
- **GaussianBlur**: Applying a Gaussian blur to emulate variations in focus and atmospheric conditions.
- **RandomRotation**: Rotating images within a 30-degree range to account for various angles of traffic signs in real-world scenarios.

These transformations were carefully calibrated to maintain the integrity of the traffic signs while introducing sufficient variability to train the model effectively. For validation data, we employed a more conservative approach with *Resize* and *ToTensor* operations to ensure the model's ability to generalize to new, unaltered data.

Through these augmentation strategies, we were able to enhance the model's learning process, allowing it to achieve high accuracy while maintaining robustness against a diverse set of real-world conditions. This approach proved vital in addressing the data imbalance and contributed significantly to the reliability of our traffic sign recognition model.

7. Conclusion

In conclusion, the experimental results of our study have demonstrated a remarkable improvement in traffic sign recognition accuracy. Our custom Convolutional Neural Network (CNN) achieved a test accuracy of 99.1%, outperforming six other pre-trained models including ResNet50, EfficientNet, Inception, MobileNet, DenseNet, and VGG. The superiority of our model's performance can be attributed to a well-calibrated combination of architectural choices and training optimizations that were specifically tailored to the unique challenges of traffic sign classification.

This comparison from Table 2 underscores the potential of using custom CNN architectures over pre-trained models for specific image recognition tasks. Not only did our model achieve higher accuracy, but it also demonstrated robustness and efficiency, making it a promising approach for real-world applications in traffic sign recognition. Future studies may explore further enhancements to this model and extend its application to other domains in image classification.

References

[1] A. Kerim and M. Ö. Efe, "Recognition of Traffic Signs with Artificial Neural Networks: A Novel Dataset and Algorithm," in *Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, Republic of Korea, April 13-16, 2021, pp. 171–176. 1

[2] D. Soni, R. K. Chaurasiya, and S. Agrawal, "Improving the Classification Accuracy of Accurate Traffic Sign Detection and Recognition System Using HOG and LBP Features and PCA-Based Dimension Reduction," in *Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Amity University Rajasthan, Jaipur, India, January 20-22, 2019. 1

[3] N. Namyang and S. Phimoltares, "Thai traffic sign classification and recognition system based on histogram of gradients, color layout descriptor, and normalized correlation coefficient," in *Proceedings of the 2020-5th International Conference on Information Technology (InCIT)*, Chonburi, Thailand, October 21-22, 2020, pp. 270–275. 1

[4] W. Li, H. Song, and P. Wang, "Finely Crafted Features for Traffic Sign Recognition," *Int. J. Circuits Syst. Signal Process.*, vol. 16, pp. 159–170, 2022. 2

[5] A. Madani and R. Yusof, "Traffic sign recognition based on color, shape, and pictogram classification using support vector machines," *Neural Comput. Appl.*, vol. 30, pp. 2807–2817, 2018. 2

[6] G. Sapijaszko, T. Alobaidi, and W. B. Mikhael, "Traffic sign recognition based on multilayer perceptron using DWT and DCT," in *Proceedings of the 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Dallas, TX, USA, August 4-7, 2019, pp. 440–443. 2

[7] S. Aziz and F. Youssef, "Traffic sign recognition based on multi-feature fusion and ELM classifier," *Procedia Comput. Sci.*, vol. 127, pp. 146–153, 2018. 2

[8] H. M. Weng and C. T. Chiu, "Resource efficient hardware implementation for real-time traffic sign recognition," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, April 15-20, 2018, pp. 1120–1124. 2

[9] B. Wang, "Research on the Optimal Machine Learning Classifier for Traffic Signs," in *SHS Web of Conferences*, EDP Sciences, Les Ulis, France, 2022, vol. 144, p. 03014. 2

[10] I. Siniosoglou, P. Sarigiannidis, Y. Spyridis, A. Khadka, G. Efstathopoulos, and T. Lagkas, "Synthetic Traffic Signs Dataset for Traffic Sign Detection & Recognition In Distributed Smart Systems," in *Proceedings of the 2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Pafos, Cyprus, July 14-16, 2021, pp. 302–308.

[11] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, pp. 975–984, 2018. 2

[12] Y. Zhu and W. Q. Yan, "Traffic sign recognition based on deep learning," *Multimed. Tools Appl.*, vol. 81, pp. 17779–17791, 2022. 2

[13] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Procedia Eng.*, vol. 201, pp. 718–725, 2017.

[14] D. A. Alghmgham, G. Latif, J. Alghazo, and L. Alzubaidi, "Autonomous traffic sign (ATSR) detection and recognition using deep CNN," *Procedia Comput. Sci.*, vol. 163, pp. 266–274, 2019.

[15] W. Li, D. Li, and S. Zeng, "Traffic sign recognition with a small convolutional neural network," in *IOP Conference Series: Materials Science And Engineering*, IOP Publishing, Bristol, UK, 2019, vol. 688, p. 044034.

[16] R. Yazdan and M. Varshosaz, "Improving traffic sign recognition results in urban areas by overcoming the impact of scale and rotation," *ISPRS J. Photogramm. Remote Sens.*, vol. 171, pp. 18–35, 2021.

[17] B. X. Bangquan and W. X. Xiong, "Real-time embedded traffic sign recognition using efficient convolutional neural network," *IEEE Access*, vol. 7, pp. 53330–53346, 2019. 3

[18] A. Zaibi, A. Ladgham, and A. Sakly, "A lightweight model for traffic sign classification based on enhanced LeNet-5 network," *J. Sensors*, vol. 2021, Article ID 8870529, 2021. 3

[19] K. V. N. Sreya, “Traffic Sign Classification Using CNN,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, pp. 1952–1956, 2021. 3

[20] U. S. Abudhagir and N. Ashok, “Highly sensitive Deep Learning Model for Road Traffic Sign Identification,” *Math. Stat. Eng. Appl.*, vol. 71, pp. 3194–3205, 2022. 3