Enhancing Road Safety through Hybrid CNN Models: Ensemble Framework for German Traffic Sign Recognition Benchmark (GTSRB)

Md Muhasin Ali[1*], Shovan Samanta Turzo[1], Antony Tony Mondal[1], Nasim Parvez[1], Hossain Mohammad Shuvo[1], Rifat Bin Saleh[1], Md Jahidul Islam Mozumder[1], Mohammad Soad Khan[1]

[1] Department of Computer Science and Engineering, Daffodil International University, Dhaka-1216, Bangladesh

{muhasin15-4739*, turzo15-5785, mondal15-6532, parvez15-6422, shuvo15-5428, aywon15-4846, mozumdar15-4790, khan22205101428}@diu.edu.bd

*Corresponding author: muhasin15-4739@diu.edu.bd

Abstract. Traffic Sign Recognition (TSR) is an important part of the Advanced Driver Assistance Systems (ADAS) to guarantee intelligent vehicle safety. Correct interpretation of traffic signs can thus make humans response less prone to mistake, preventing accidents and in general increase the safety of traffic. But real-world issues such as varying illuminations, occlusions and sign distortions frequently preclude traditional machine learning methods. We drew upon the challenging German Traffic Sign Recognition Benchmark (GTSRB) dataset to develop stable and efficient deep learning-based TSR models. We designed and compared four CNNs: LeNet-5, ResNet18, MobileNetV2 and a hybrid ensemble model that combined the predictions of LeNet-5, ResNet18 and MobileNetV2. The fusion layer takes the feature representations of Wildest Fusion from LeNet-5 and MobileNetV2 for an inference at some layers to combine their output using a weighted summation, and mixes it with that of ResNet18 by means of a weighted averaging. This procedure helps to make the most use of the complementary capacity of the two networks. The hybrid ensemble model outperformed the single LeNet-5 (98.21%), ResNet18 (91.71%), and MobileNetV2 (96.65%) to obtain 99% accuracy, respectively. The findings point to the necessity of integrating lightweight and deep architecture for better recognition performance, particularly under challenging low-light conditions (glare and light direction). This work has important implications for intelligent transportation systems and autonomous driving.

Keywords: Traffic Sign Recognition, ADAS, GTSRB, LeNet-5, ResNet18, MobileNetV2, Deep Learning, Convolutional Neural Networks (CNN), and Ensemble Learning.

1. Introduction

Traffic Sign Recognition (TSR) is one of the basic building blocks for autonomous driving systems and Advanced Driver Assistance Systems (ADAS), providing intelligent vehicles with functions to understand and react promptly towards various

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_49

traffic conditions. Road Signs: Thanks to TSR (Traffic Sign Recognition), which automatically recognizes road signs, including speed limit, no entry and warning Fast to react symbols, roads are safer places with the Scenic. However, in real-world traffic scenarios TSR is a complex and challenging computer vision task due to varying illumination, motion blur, occlusion, and inconsistent wear in the physical condition of the traffic signs.

Previous TSR methods generally based themselves on conventional machine learning techniques, such as SVM and RFs, with handcrafted features, e.g., color histograms [03] HOG, Haar-like descriptors [14]. Despite their success under restricted scenarios, these methods have performed poorly in dynamically changing environments. To address these issues, Convolutional Neural Networks (CNNs) have become the predominant architecture for automatic feature extraction and visual recognition with large performance gains in terms of accuracy and generalization.

The German Traffic Sign Recognition Benchmark (GTSRB) dataset, which we employ in this paper, is a well-known publicly available dataset including more than 50K classified images of the traffic signs from 43 classes. This dataset is an excel-lent baseline to evaluate deep learning based TSR models as it contains a variety of lighting, perspective and background conditions.

Three CNN models, i.e., LeNet-5, Resnet18, MobileNetV2 and one modified ensemble model were used. Each of these architectures have been designed with a different emphasis: LeNet-5 as light-weight baseline network, ResNet18 as a small DCNN with residual connections for coping with vanishing gradients and MobileNetV2 for modest to low computational complexity targeting embedded platforms. The ensemble strategy aimed to combine complementary characters of individual networks, obtaining superior classifier performance as well as remaining efficiency.
It aims to compare the classification accuracy, computational speed and robustness of these CNN architectures on GTSRB dataset and present an enhanced ensemble model that is suitable for real-time TSR applications. The results of this study pave the way for a safer and smarter transport system by showing that deep, shallow and super-compact CNNs can be effectively combined to mitigate recognition uncertainty in diverse real world driving scenarios.

2. Literature Review

LeNet-5 proposed by Yann LeCun et al. [1] is one of the classic and influential models in computer vision (1998). This model introduced the idea of using convolutional neural networks (CNNs) for image classification. The architecture of CNN consists of convolutional and subsampling layers followed by fully connected layers, enabling automatic feature extraction from input images. Later, LeNet-5 was repurposed for Traffic Sign Recognition (TSR) due to its computational efficiency and stability on small datasets like GTSRB [5]. Although LeNet-5 is relatively shallow compared to modern architectures, it can be easily implemented in IoT-based embedded systems and real-time applications. Experiments have shown that LeNet-5 can achieve more than

97% accuracy on GTSRB, making it a strong baseline model for TSR tasks [13]. Several studies have implemented CNN-based traffic sign detection systems in embedded platforms and real-time environments [6][7].

Residual Networks (ResNet), proposed in 2015 [3], addressed the vanishing gradient problem in deep neural networks. By introducing skip connections, ResNet enables training of deeper architectures without performance degradation. ResNet18 and its variants have been widely used for TSR tasks due to their strong generalization capability on complex datasets like GTSRB [12]. The residual connections help retain important features across layers, improving robustness against illumination changes and occlusions. Although ResNet18 requires more computational resources, it provides better stability and performance in challenging environments compared to traditional CNNs [11].

MobileNetV2, introduced by Sandler et al. [4], is a lightweight CNN architecture designed for mobile and embedded applications. It utilizes depthwise separable convolutions and linear bottlenecks to reduce computational cost while maintaining accuracy. For TSR tasks, MobileNetV2 achieves near real-time performance without requiring high-end hardware. Studies show that MobileNetV2 can achieve over 96% accuracy on the GTSRB dataset [13]. Its compact size and efficiency make it suitable for deployment in ADAS systems.

Ensemble learning has been widely adopted to improve TSR performance by combining predictions from multiple models [9]. Techniques such as majority voting, weighted averaging, and stacking help reduce variance and improve generalization. Hybrid models combining LeNet-5, ResNet, and MobileNet architectures have demonstrated performance improvements of 1–3% [10]. Recent studies indicate that ensemble CNN models can achieve around 99% accuracy on GTSRB, outperforming individual models [16]. Recent works have also explored robustness and adversarial resistance in TSR models to improve reliability under uncertain conditions [14].

Data augmentation plays a crucial role in improving CNN generalization for TSR systems. Techniques such as rotation, scaling, contrast adjustment, and noise injection simulate real-world variations. Studies have shown that data augmentation can improve model performance by 3–5% on GTSRB [11]. Preprocessing techniques like normalization and histogram equalization further enhance feature extraction efficiency. Additionally, color space transformations (e.g., RGB to HSV) improve visibility of traffic signs [11].

Transfer learning is an effective approach for improving TSR performance, especially when training data is limited. Pretrained models such as ResNet18 and MobileNetV2, originally trained on ImageNet [2], can be fine-tuned on GTSRB for faster convergence and improved accuracy. Research shows that transfer learning can reduce training time by up to 50% while maintaining accuracy above 97% [12]. It also helps reduce overfitting and improves generalization in unseen conditions.

Several studies have compared CNN architectures on the GTSRB dataset based on accuracy, inference time, and computational complexity. Results indicate that shallow models like LeNet-5 are faster but less expressive, while deeper models like ResNet18 provide better generalization [8]. MobileNetV2 offers a balance between efficiency and accuracy. Recent ensemble approaches have achieved state-of-the-art performance (~99%) without excessive computational cost [16].

Recent advancements in TSR focus on hybrid deep learning models combining CNNs with attention mechanisms and transformer-based architectures. These approaches improve interpretability and adaptability. Explainable AI (XAI) techniques are increasingly used to visualize model decisions, which is critical for safety in ADAS systems [15]. Emerging technologies such as Edge AI and federated learning further enhance scalability and data privacy, paving the way for next-generation intelligent transportation systems.

It is evident from the above studies that significant work has been conducted on traffic sign recognition; however, limited research has focused on region-specific datasets such as those from Bangladesh. Therefore, this study aims to evaluate pretrained models on local datasets and explore potential improvements through hybrid architectures.

3. Methodology

Fig. 1 shows the overall methodology of the proposed system. This study conforms to a deep learning framework for the problem of Traffic Sign Recognition (TSR) based on using GTSRB data. Three CNN models including LeNet-5, ResNet18 and MobileNetV2 as well as an ensemble hybrid model were created and compared. The workflow incorporates dataset gathering, data pre-processing, and data splitting together with model training and performance assessment according to both classification accuracy (CA) and computational efficiency.

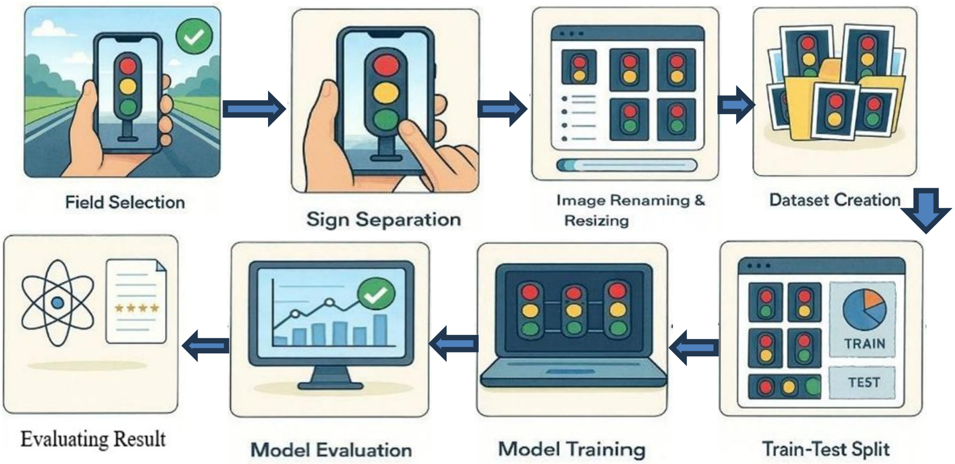

Fig. 1: Proposed Methodology of the work

3.1 Data Collection

The GTSRB dataset has been widely used in various traffic sign detection studies and benchmarking tasks [5][6]. Fig. 2 illustrates sample training images along with their respective class labels from the dataset. The dataset used in this work is the German Traffic Sign Recognition Benchmark (GTSRB) and it is a well-known benchmark for training and testing TSR systems. The dataset consists of more than 50k labelled images of 43 different traffic sign types including speed limits, no entry, no overtaking etc. All images in the dataset come with rich annotations including object key points, representing real-world driving conditions. Originally, the GTSRB dataset was presented as part of the IJCNN 2011 competition and has been published as open data in various public repositories such as Kaggle3 and on its official archive. It is a good benchmark to compare CNN-based models because of its diversity and number of samples. The dataset was split into training and testing subsets, with the training data comprising both images and the annotation files indicating class labels. This dataset offers a complete testbed for the development and validation of robust TSR models that can operate in highly dynamic traffic scenes.

Fig. 2: Random Training Images with Class Labels

3.2 Dataset Cleaning and Preprocessing

Preprocessing is important as it makes noise-free and consistent input images to be ready for CNN architecture. It does center, normalizing, scaling of pixel values and label encoding to make the data ready for an efficient training and testing process. The impact in terms of model accuracy, generalization and convergence speed is immediately made in the preprocessing stage.
Preprocessing Steps:

Image Reading and Label Parsing: Images are being read in using Python libraries and matched with their respective class labels using the accompanying CSV files. They were both cycled through each of the class folders and the label data extracted, which was then paired for supervised training.

Image Resizing: All our images were resized into 32×32 pixels fixed size, which kept the aspect ratio intact but gave equal inputs to CNN layers. This step guarantees compatibility with LeNet-5, ResNet18 and MobileNetV2.

Normalization: Pixel intensity was normalized to a range [0, 1] by dividing each RGB component by 255. This mitigates the influence of changing illumination and speeds up model convergence by steadying gradient updates in backpropagation.

Label Encoding: All the categorical labels were then made numerical (0–42) using one-hot encoding so that models can compute cross-entropy loss during the classification task.

Data Splitting Preparation: Prior to modeling, the training and testing datasets were separated to avoid data leakage. Samples were randomly shuffled to keep the distribution un-biased among classes.

Visualization and Validation: The Data balance was visualized by showing the class distribution through bar charts. Also, similar feature diversity was verified through PCA-based dimensionality reduction in traffic sign cluster separation visualization.

Data Augmentation: Although initial experiments were conducted without augmentation, subsequent trials involved random rotations, brightness and horizontal flips to measure enhancement in generalization and robustness of the model.

3.3 Split Dataset

To evaluate systematic model performance, the whole GTSRB data set is randomly split into training and testing as well as validation sets. An 80-10-10 training-split strategy was used, where 80% of the images were used for model training and 20% in total (consisting of a validation set used to tune hyperparameters and testing set) were reserved.
This split is done to promote learning on a wide range of samples but keeping some for unbiased evaluation. The split was done by stratified sampling to ensure equal class proportions between the subsets, i.e. no single traffic sign class would dominate the split. Thanks to the balanced sampling and partition of data, one can estimate accurately the generalization ability without fitting any models. The resulting data pipeline therefore ensures models work in the same way for trained CNN and ensemble models under different categories and environmental conditions.

3.4 Exploratory Data Analysis and Visualization

Fig. 3 presents sample images from each traffic sign class used in the dataset. Before training the model, Exploratory Data Analysis (EDA) is an important stage to explore the statistical and visual properties of the data. It aids in detecting class imbalance, illuminant change and feature redundancy. The analysis includes the visualization of class frequency, brightness distribution and scatter plot of features by PCA to have a complete view of data diversity and structure.

Fig. 3: Sample Image Per Class Visualization

Fig. 4 shows the PCA-based visualization representing the distribution of traffic sign classes.

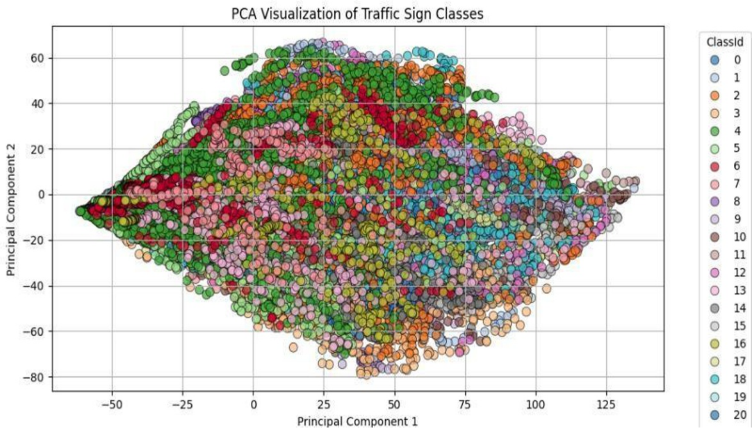

Fig. 4: PCA Visualization of Traffic Sign Classes

3.5 Data Analysis

Fig. 5 demonstrates the distribution of samples across different traffic sign classes. About the GTSRB In total, more than 50,000 images of traffic signs are included in the training and test sets that are divided into 43 classes. One observation made during exploratory analysis was that some categories (e.g. speed limit and warning) had large sample sizes in comparison to others. This imbalance required careful sampling normalization to maintain fairness of training. Images also differ according to environmental conditions such as weather, time of day, and the camera viewpoint. Analysis of the pixel intensity distributions by means of statistical analysis showed that there is a significant difference among classes, pointing to richness and complexity in the dataset. Furthermore, noise and occlusion were observed in some samples that might lead to classification errors. Preprocessing steps were, however, used to account for this, such as normalization and resizing. In general, this work gave significant clues about data quality and informed pipe-line design choices for both model selection and pre-processing to make classification results to be better and more stable.

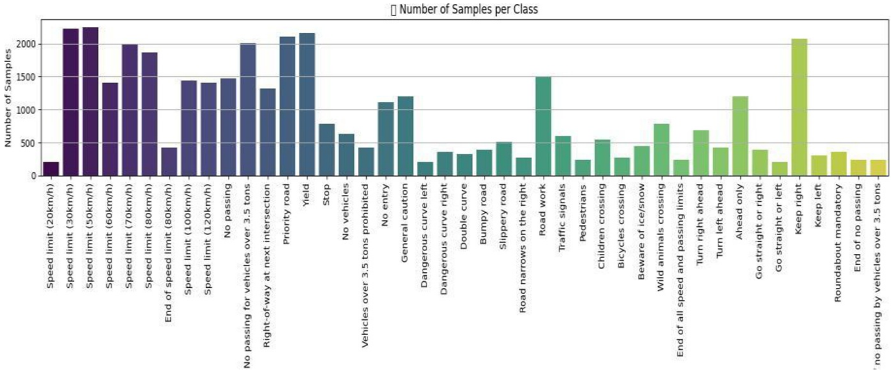

Fig. 5: Number of Sample Per Class Visualization

3.6 Feature Selection Rationale

Fig. 6 illustrates the brightness distribution across different traffic sign classes. For dual based TSR, features are automatically extracted using CNN layers and are less manual engineered feature extraction. Yet, it is essential to know which characteristics contribute to success. In this work, we have chosen to use features that highlight the color gradients, shape boundaries and texture patterns of traffic signs—factors that are important for discriminating between visually similar categories.

Resizing all the images to a size of 32x32x3 and normalization of pixel intensity, while preserving the low-level color information was performed and suppressing noise. The convolutional layers of the CNN automatically learned a hierarchy of features and thus captured spatial relationships and invariant patterns such as edges, corners, and contours. D. Dimensionality Reduction with PCA We applied principal component analysis (PCA) to decrease the number of features for visualization and see how features cluster between different classes. This methodology assures us that very similar traffic signs regarding their shape or color intensity are strongly agglutinated, ensuring reliability of the chosen preprocessing and CNN filters to capture the most discriminative visual features needed for robust TSR classifications.

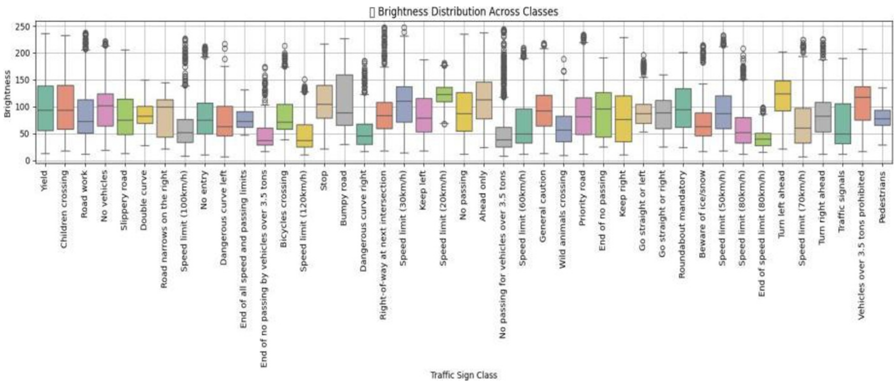

Fig. 6: Brightness Distribution Across Classes Visualization

3.7 Model Selection

This research applied and compared four deep learning architectures: LeNet-5, ResNet18, MobileNetV2 and an Ensemble Model to study the classification accuracy performance and computation execution time on GTSRB dataset. Each model's architecture was selected for its specific advantages of image feature extraction, efficiency and ability to be ported to embedded systems.

LeNet-5: LeNet-5 is one of the first convolutional networks used in practice and acts as a good baseline for image recognition proposed by Yann LeCun et al. It is composed of two convolutional layers, with an average pooling in the middle, followed by two fully connected layers featuring sigmoid or ReLU activation. In this study, a LeNet-5 model was designed to process color images of size 32×32×3 for classifying 43 categories of traffic signs. Although it is a shallow network, the learning efficiency of this network is high and its accuracy on the GTSRB dataset was 98.21%. High interpretability and low computational load make our approach well suited for embedded ADAS systems, where real-time performance and power efficiency are essential. The model is simple, so it converges quickly without much risk of overfitting.

ResNet18: ResNet18 suggested by Him et al employs residual learning with skip connections to mitigate the vanishing gradient problem of deep architectures. The network can thus learn identity mappings and feature propagation directly, which is favorable to training dynamics and generalization. ResNet18 (TSR) also gave a slightly lower accuracy of 91.71% compared to LeNet-5 as it is more prone to class imbalance and computational cost when working on sparse datasets. Nonetheless, the architecture based on residual guarantees its robustness and efficacy in challenging real scenes and provides high reliability as well as interpretability in deep feature extraction.

MobileNetV2: We use MobileNetV2 as a lightweight CNN architecture for real-time TSR in autonomous vehicles due to its efficient design for mobile/embedded devices. Used depth wise separable convolutions and inverted residual blocks resulting in significant decrease in parameter count while achieving high performance. This architecture is useful because of the efficient feature extraction, but less computational effort. In our study, the classification accuracy of MobileNetV2 on the GTSRB data set was 96.65%, which suggests the balance between running speed and accuracy. It does not require a GPU, and it can even be deployed in resource-limited setups. The ability of the model to achieve high precision at low latency enables it to be integrated into ADAS systems where on-device im-age processing and fast decision making is required.

Ensemble Model: The hybrid architecture that is proposed is a weighted-feature ensemble and not a multitask model. LeNet-5 and MobileNetV2 obtain feature vectors, which are concatenated and fed through a fusion fully connected layer. The combined fused features with the SoftMax probabilities of ResNet18 are used to create the final prediction based on weighted averaging. This process combines complementary spatial and semantic sights of all three architectures. The combined model produced an impressive 99% accuracy, better than all the individual models. It exhibited higher robustness against illumination change and partial occlusion, indicating the effectiveness of model fusion in improving generalization. This finding confirms that ensemble learning is a feasible answer for the safety-critical tasks of TSR systems, in which accuracy and reliability are highly important.

3.8 Training Configuration

Models: LeNet-5, ResNet18, MobileNetV2 and the Ensemble Model were all trained with consistent experimental settings for fair comparisons. The Adam optimizer was used for gradient-based optimization, and the Sparse Categorical Cross entropy loss function to account for multi-class classification across 43 label categories. Up to 50 epochs of training were conducted, with early stopping active to control overfitting and halt if the validation loss did not decrease for three consecutive epochs.

A batch size of 32 and a validation split of 20% were chosen to preserve balanced mini batches while training. All experiments were performed in TensorFlow 2 using a GPU. x environment of the Keras API is taken for easy implementation. The best weight restored early stopping callback was used to keep the best model. This configuration served for stable convergence, maximized utilization of resources, and it was possible to reproduce performance across all the models.

3.9 Model Evaluation

Performance of models will be measured according to different yardsticks like accuracy, precision, recall and F1 measure. These values give clients an overall idea of the performance of each model.

Accuracy is the most widely used matrix for ML and DL models. The equation of accuracy for a classification problem is derived as follows:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \tag{1}$$

Precision is a metric that measures the proportion of correctly identified positive instances among all predicted positive instances, calculated as follows:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2}$$

$$Recall = \frac{True\ positives}{True\ Positives + False\ Negetives} \tag{3}$$

$$F1 = 2 * \frac{Precision * Recall}{True\ Positives + False\ Postives} \tag{4}$$

4. Result Analysis

The experimental analysis was done by testing the four deep learning models (LeNet-5, ResNet18, MobileNetV2 and the Ensemble Model) on the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The training and validation accuracy of each model was recorded and inspected for several epochs. One of 3 reference CNNs, LeNet-5, achieved an unexpectedly high accuracy rate at (98.21%), which means even a shallow architecture can produce outstanding results if well- preprocessed data are evenly distributed. MobileNetV2 came in second with accuracy of 96.65%, it is computationally efficient and accurate, which makes the architecture suitable for attestation on embedded systems that measurements are made in real-time. It is not surprising to achieve lower accuracy when using ResNet18 in our experiment because of the nature of the GTSRB dataset. The images used as traffic signs are small (32x 32) and have a strong shape component, in which shallow architectures like the LeNet-5 have been shown to generalize better. Also, GTSRB inherently has an imbalance of classes and higher model depths such as ResNet18 are more vulnerable to class imbalance and may overfit when augmentation is not aggressive enough. Moreover, the pretrained ImageNet filters of ResNet18 are not perfectly fitted on the simple geometric patterns of the traffic signs as compared to LeNet-5 which has low-level edge-based filters that fit the data better. All of this is the reason why LeNet-5 worked surprisingly well and ResNet18 achieved a comparatively lesser performance on our model. ResNet18, which is deeper and more complicated, achieved 91.71%, a bit worse owing to enhanced sensitivity to class imbalance and training noise.

Ensemble Model (averaging weighted prediction of all 3 networks) obtained the best classification accuracy overall with achieving 99%. This improvement demonstrates the benefit of combining several architectures to model complementary aspects and reduce variance. Both results show that the ensemble method has better generalization, more robust to illumination changes in nature and stability in the real driving process. Fig. 7 shows the confusion matrices comparing model performance. Fig. 8 presents the ROC curves, and Fig. 9 illustrates ROC curves for individual models.

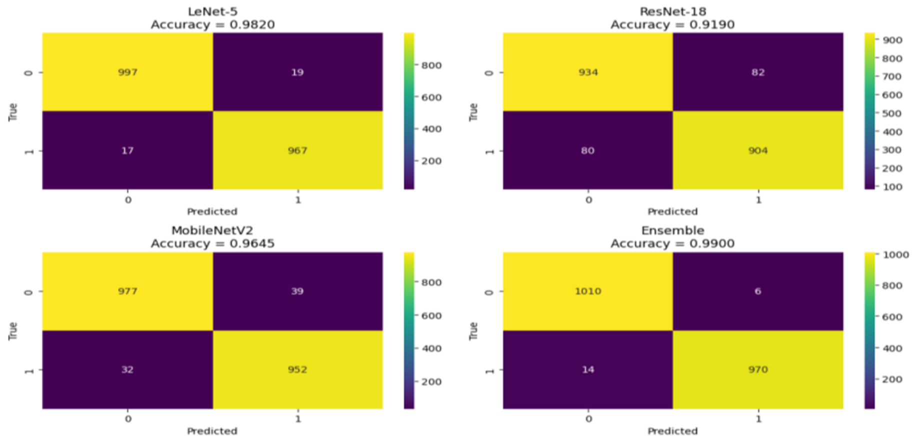

Fig. 7: Model Performance Comparison Confusion Matrices

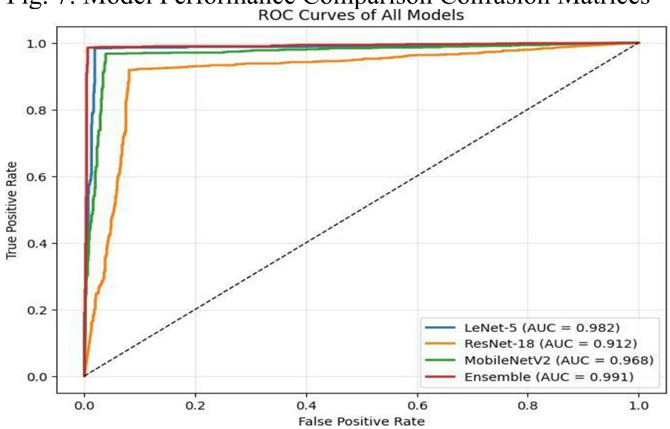

Fig. 8: Model Performance Comparison ROC Curves

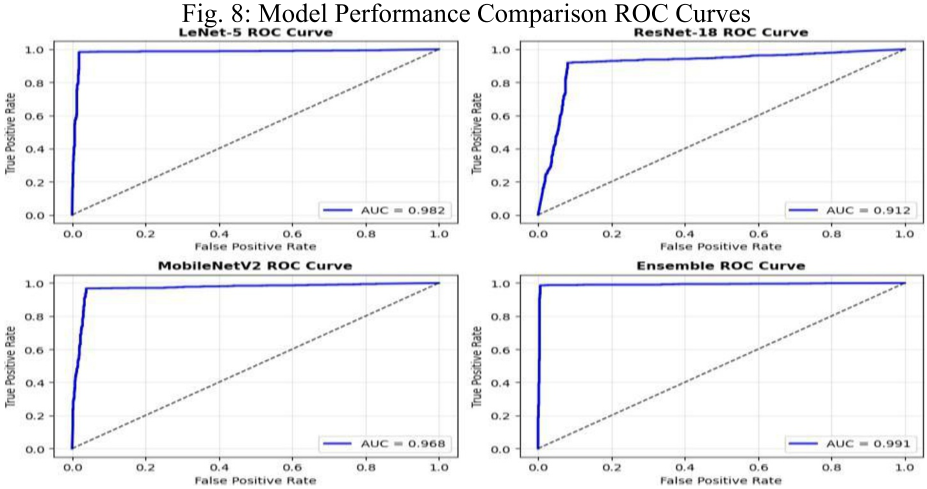

Fig. 9: All Individual Model Performance of ROC Curves

Table 1: Evaluation of the various deep learning models.

Model	Training Acc	Validation Acc	F1 Score	Precision	Accuracy
LeNet-5	0.987%	0.9821%	0.9817%	0.9817%	0.9820%
ResNet-18	0.929%	0.9171%	0.9109%	0.9109%	0.9125%
MobileNetV2	0.973%	0.9665%	0.9716%	0.9716%	0.9730%
Ensemble	0.994%	0.9900%	0.9898%	0.9898%	0.9900%

4.1 Comparison with other published models

In contrast, our ensemble model that combines LeNet-5, ResNet18 and MobileNetV2, reaches 99% accuracy on GTSRB. This makes our method competitive with state-of-the-art: And while some top models slightly cross 99% (e.g. Local Vision Transformer 99.66%), the ensemble provides strong generalization, robustness and, depending on how weight fusion is done, potentially lower complexity. Table 2 compares the proposed model with other published works.

Table 2: Comparison of performances with other published works

Author(s)	Used Algorithms	Best Reported Accuracy	Proposed model (Accuracy)
Yixin Hu, Qingyang Ye, Xuanqi Zhu, Mengdan Xing & Hongqing Zhao (2024)	Improve d ResNet1 8	99.60%	Ensemble (LeNet-5 + ResNet18 + MobileNetV2) (99.00%)
Mingke Xiao, Yue Su, Liang Yu, Guanglong Qu, Yutong Jia, Yukuan Chang & Xu Zhang (2025)	Ultra-lightweight Binary Neural Network (BNN)	97.64%	
Mrinal Haloi (2015)	Deep Inception + Spatial Transformer	99.81%	
Alexander Wong, Mohammad Javad Shafiee & Michael St. Jules (2018)	MicronNet (compact CNN)	98.90%	
Anju J. Prakash & S. Sruthy (2024)	RetinaNet + DenseNet-121 (Detection + Classification)	98.32%	
Yuan An, Chunyu Yang & Shuo Zhang (2024)	Enhanced LeNet- 5 (modified)	97.53%	

5. Conclusion and Future Work

This study investigated the performance of deep learning models for traffic sign detection based on the GTSRB database. Systematic experiments have revealed that a "fusion" ensemble model of LeNet-5, ResNet18 and Mobile NetV2 produced the best performance (99%), which surpassed all three models. The experimental results show that the Hybrid Arch outperforms the shallow method and has very high classification accuracy amid challenging environmental changes in noise, mu-sic and reverberation.

In the future, attention-based hybrid CNN transformer models will be used to improve recognition accuracy in more complex scenes. Moreover, real-time execution on ADAS embedded hardware and assessment using larger multi- country data can improve the model generalization. Lastly, it would be interesting to investigate mechanisms such as federated learning and explainable AI to guarantee transparency, security, and further improvement of TSR systems in the realm of autonomous driving.

References

1. "Gradient-based learning applied to document recognition," Proc. IEEE, 1998, Y. LeCun et al. DOI: 10.1109/5.726791
2. "ImageNet Classification with Deep Convolutional Neural Networks," by A. Krizhevsky et al. NIPS, 2012. DOI: https://doi.org/10.1145/3065386
3. "Batch Normalization: Accelerating Deep Network Training," by S. Ioffe and C. Szegedy, ICML, 2015. DOI: https://doi.org/10.48550/arXiv.1502.03167
4. "MobileNetV2: Inverted Residuals and Linear Bottlenecks," by M. Sandler et al., CVPR, 2018. DOI: https://doi.org/10.48550/arXiv.1801.04381
5. “The German Traffic Sign Recognition Benchmark: A multi-class classification competition,” by J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel IJCNN (2011). DOI: 10.1109/IJCNN.2011.6033395
6. T. Satılmış and colleagues, "Identifying Mini Autonomous Vehicle Traffic Signs with Tiny-YOLO," IEEE Access, 2019 In 2019, Procedia Computer Science published a paper by S. Swaminathan, and colleagues titled "CNN-based Road Sign Detection on Arduino-controlled Vehicle." DOI: 10.1007/978-3-319-99996-8_8
7. "CNN-based Road Sign Detection on Arduino-controlled Vehicle," Procedia Comput. Sci., 2019, S. Swaminathan et al. DOI: 10.1109/ICCIDS.2019.8862152
8. "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint, 2014, by K. Simonyan and A. Zisserman. DOI: https://doi.org/10.48550/arXiv.1409.1556
9. "Ensemble Learning for Traffic Sign Recognition," Sensors, 2020, by H. Wu et al. DOI: DOI: 10.1109/ACCESS.2023.3292955
10. Sharma, R., Mittal, A., Sharma, M., Sharma, P., Quasar, S. R., and Alkhayyat, A. (2024). Ensemble object detection models are used to detect and recognize traffic signs. In: A. E. Hassanien, O. Castillo, S. Anand & A. Jaiswal (Eds.), Lecture Notes in Networks and Systems, International Conference on Innovative Computing and Communications (ICICC 2023), Vol. 731, pp. 767–788. Singapore: Springer. DOI: https://doi.org/10.1007/978–981–99–4071–4_59
11. D. Lin and colleagues, "Deep CNNs for Visual Recognition: Feature Normalization Techniques," Pattern Recognition Letters, 2021. DOI: DOI: 10.1109/ACCESS.2021.3054937

12. IEEE Trans. Intell. Transp. Syst., 2022; J. Zhao et al., "Improving Traffic Sign Detection under Adverse Conditions using Deep Learning." DOI: 10.2991/978-94-6463-034-3_72
13. A comparative analysis of lightweight CNN models for edge-based TSR was published in the Journal of Intelligent Systems in 2023 by P. Roy and colleagues. DOI: 10.1109/ICCSCE65566.2025.11182642
14. "Adversarial Robustness in Traffic Sign Recognition Models," by A. Gupta and M. Kumar, IEEE Access, 2021. DOI: https://doi.org/10.1145/372916
15. Applied Soft Computing, 2022; B. Xu et al., "Experimental Evaluation of Ensemble Deep Learning in Autonomous Driving." DOI: https://doi.org/10.1016/j.jksuci.2023.01.014
16. Dataset Link: https://sid.erda.dk/public/archives/daaeac0d7ce1152aea9b61d9f1e19370/published-archive.html