AI-Powered Traffic Sign Classification with Interactive GUI Integration

Supriya Parashuramsingh Rajput[1]

[*1] *Assistant Professor, Dept. of Electronics and Communication Engineering, MMEC Belagavi, Karnataka, India*

Abstract - *Traffic sign recognition plays a pivotal role in the development of autonomous vehicles and advanced driver-assistance systems (ADAS), significantly enhancing road safety. This project leverages the power of Convolutional Neural Networks (CNNs) to classify traffic signs accurately. The German Traffic Sign Recognition Benchmark (GTSRB) dataset, containing images of 43 traffic sign classes captured under various conditions, was used for model training and evaluation. The images were preprocessed through resizing, normalization, and one-hot encoding, ensuring compatibility with the CNN architecture. To improve model robustness, data augmentation techniques such as rotation, zoom, and shifts were employed, creating an enriched dataset for training. The proposed CNN architecture comprises multiple convolutional, pooling, and dropout layers, enabling efficient feature extraction and classification. The model was trained using the Adam optimizer and evaluated on a separate test set, achieving high accuracy and demonstrating its effectiveness in real-world scenarios. Results showed that data augmentation significantly enhanced generalization, and the use of dropout layers reduced overfitting. The project concludes with successfully deploying a traffic sign recognition system capable of identifying traffic signs with high precision, paving the ay for integration into real-time traffic monitoring and ADAS. This achievement marks a significant step towards safer autonomous driving technologies.*

Key Words: **Traffic sign recognition, CNN, Image processing, Deep learning, Data augmentation, Keras, TensorFlow.**

1. INTRODUCTION

Traffic signs provide critical information to drivers, ensuring the safe and efficient flow of traffic. In the era of autonomous vehicles and intelligent transportation systems, the automatic recognition of traffic signs has become indispensable. Advanced driver-assistance systems (ADAS) rely on accurate and timely detection of traffic signs to guide decision-making processes and alert drivers to critical road conditions. However, traditional approaches to traffic sign recognition, which often depend on handcrafted features and classical machine learning techniques, struggle to achieve the accuracy and robustness required for real-world applications.

The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized image recognition tasks, including traffic sign classification. CNNs automatically learn hierarchical feature representations from raw images, eliminating the need for manual feature engineering. This project aims to harness the capabilities of CNNs to develop a reliable traffic sign recognition system using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The dataset comprises 43 distinct traffic sign classes, providing a comprehensive benchmark for evaluating the effectiveness of machine learning models in this domain.

The proposed system begins with pre-processing steps, including resizing images to pixels, normalizing pixel values to a range of [0, 1], and encoding labels using one-hot representation. To enhance the model's robustness and generalization capabilities, data augmentation techniques such as random rotation, translation, and zooming were applied during training. The CNN architecture was meticulously designed to extract spatial and hierarchical features from images, employing convolutional layers for feature extraction, pooling layers for down-sampling, and dropout layers for regularization.

The training process involved optimizing the model using the Adam optimizer and monitoring its performance on a separate validation set. The model was evaluated on a hold-out test set to assess its generalization to unseen data. Performance metrics such as accuracy, loss, and a confusion matrix were used to analyse the model's effectiveness. The results demonstrated that the proposed CNN-based system could classify traffic signs with high precision, achieving performance suitable for deployment in real-world scenarios.

This project highlights the potential of deep learning to address complex challenges in intelligent transportation systems. By accurately recognizing traffic signs, the developed system contributes to safer roadways and lays the groundwork for future advancements in autonomous driving technology. Furthermore, the integration of data augmentation and dropout techniques underscores the importance of designing robust models capable of handling diverse and challenging environments.

2. LITERATURE SURVEY

Paper 1: Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. "The German Traffic Sign Recognition Benchmark: A multi-class classification competition." Neural Networks 32 (2012): 323-332.

This paper introduces the GTSRB dataset and provides an extensive benchmark for multi-class traffic sign classification.

The study highlights the challenges of recognizing traffic signs in varying conditions such as lighting, occlusion, and distortion. It evaluates multiple machine learning models, setting a foundation for the development of advanced traffic sign recognition systems. However, the study lacks exploration of modern deep learning techniques, which have since demonstrated superior performance.

Paper 2: Sermanet, P., & LeCun, Y. "Traffic sign recognition with multi-scale convolutional networks." Neural Networks 48 (2013): 58-69.

This study proposes a multi-scale convolutional network for traffic sign classification, achieving state-of-the-art accuracy at the time. The authors emphasize the importance of hierarchical feature extraction and demonstrate the effectiveness of CNNs in handling diverse traffic sign datasets. While the model performs well, its computational cost limits real-time applicability on embedded systems.

Paper 3: Ciregan, D., Meier, U., & Schmidhuber, J. "Multi-column deep neural networks for image classification." IEEE Conference on Computer Vision and Pattern Recognition (2012): 3642-3649.

The authors present a multi-column deep neural network (MCDNN) architecture designed for image classification tasks, including traffic sign recognition. By combining predictions from multiple CNNs, the model achieves exceptional accuracy. However, the reliance on ensemble methods increases computational requirements, posing challenges for deployment in resource-constrained environments.

Paper 4: Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., & Igel, C. "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark." International Joint Conference on Neural Networks (2013): 1-8.

This paper focuses on traffic sign detection, a precursor to recognition. The authors evaluate various detection algorithms on the GTSRB dataset, emphasizing the importance of accurate localization. While the study lays the groundwork for detection, it does not address end-to-end recognition systems integrating detection and classification.

Paper 5: Simonyan, K., & Zisserman, A. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

This seminal work introduces the VGG network, a deep CNN architecture that has influenced numerous image classification tasks, including traffic sign recognition. The study demonstrates the impact of increasing network depth on accuracy. Although highly effective, the model's high memory and computation requirements limit its deployment in real-time applications.

3. METHODOLOGY

The block diagram for the traffic sign recognition system illustrates the process from Data acquisition to Machine Learning, encompassing stages such as resizing, Normalization, label encoding, Data augmentation, and CNN.

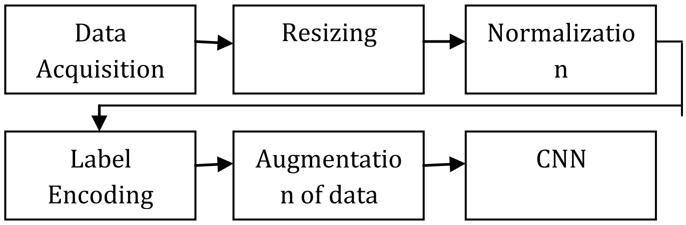

Fig.1 Block diagram of proposed methodology.

3.1 Data Acquisition

The German Traffic Sign Recognition Benchmark (GTSRB) dataset was utilized for this project. It comprises over 50,000 images spanning 43 traffic sign classes, with significant variability in lighting, angles, and background conditions. The dataset was divided into training (80%) and testing (20%) subsets to evaluate model performance effectively.

3.2 Preprocessing

To prepare the dataset for training, several preprocessing steps were performed:

Resizing: All images were resized to pixels to standardize input dimensions.

Normalization: Pixel values were scaled to the range [0, 1] by dividing by 255.0 to improve convergence during training.

Label Encoding: Traffic sign labels were converted to one-hot encoded vectors, ensuring compatibility with the categorical cross-entropy loss function used during training.

3.3 Data Augmentation

Data augmentation techniques were applied to the training dataset to enhance model generalization and reduce overfitting. Transformations included:

Rotation: Random rotations up to 10 degrees.

Shifts: Horizontal and vertical translations up to 10% of the image dimensions.

Zoom: Random zooming by up to 10%.

These augmentations created diverse variations of training samples, helping the model learn robust features.

3.4 Model Architecture

A Convolutional Neural Network (CNN) was designed with the following layers:

Convolutional Layers: Extracted spatial features using 32 filters of size and 64 filters of size, activated by ReLU functions.

Max-Pooling Layers: Reduced spatial dimensions while retaining important features.

Dropout Layers: Prevented overfitting by randomly deactivating 25% to 50% of neurons during training.

Fully Connected Layers: Combined features into high-level representations, with a final softmax layer for classification into 43 categories.

3.5 Training and Optimization

The model was trained using the Adam optimizer, which adapts the learning rate during training for faster convergence. The following parameters were used:

Batch Size: 32

Epochs: 15

Loss Function: Categorical cross-entropy

Validation performance was monitored to prevent overfitting, and early stopping techniques were employed when validation loss plateaued.

3.6 Testing and Evaluation

The trained model was evaluated on the hold-out test set, and performance metrics such as accuracy, precision, recall, and a confusion matrix were calculated. These metrics provided insights into the model's ability to generalize the data.

4. RESULT AND DISCUSSION

Graphical User Interface:

A graphical user interface (GUI) was developed using the Tkinter library to classify traffic signs based on images provided by the user. A pre-trained Keras model was loaded to predict traffic sign classes, with a dictionary mapping numerical predictions to descriptive labels. The GUI allows users to upload an image, which is preprocessed to match the model's input size before being classified. The prediction result, along with the uploaded image, is displayed for the user.

Fig.2(a) GUI to select and upload the image from the dataset to recognise and understand the meaning of traffic signs.

Fig.2(b) GUI to classify the traffic sign image. (after clicking on classify image GUI provide the meaning of a specific sign)

Fig.2(c) GUI gives the meaning of the Traffic sign which user uploads.

Table No.1 Efficiency of the model

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.7833	56.40%	0.3372	91.09%
2	0.5133	85.59%	0.2357	93.97%
3	0.3575	89.98%	0.1103	97.33%
4	0.3051	91.60%	0.0789	97.93%
5	0.2843	92.34%	0.0870	97.55%
6	0.2379	93.71%	0.0584	98.28%
7	0.2222	94.11%	0.0840	97.79%
8	0.2163	94.34%	0.0502	98.44%

9	0.2060	94.66%	0.0474	98.69%
10	0.2047	94.79%	0.0466	98.76%
11	0.2070	94.77%	0.0727	98.11%
12	0.2030	94.88%	0.0639	98.47%
13	0.2086	94.83%	0.0516	98.65%
14	0.2288	94.83%	0.0469	98.75%
15	0.1927	95.18%	0.0421	98.94%

Table No.1 highlights the training and validation performance of the model across 15 epochs, showcasing its progression in accuracy and efficiency. The training loss starts at 1.7833 during the first epoch and consistently decreases to 0.1927 by the fifteenth epoch, reflecting a steady improvement in the model's ability to minimize error during training. Simultaneously, the training accuracy improves significantly, rising from an initial 56.40% to an impressive 95.18% in the final epoch, indicating that the model learns to predict the training dataset correctly with increasing precision.

On the validation side, the validation loss begins at 0.3372 in epoch 1 and rapidly declines to 0.0466 by epoch 10, before stabilizing near this value in subsequent epochs. This trend demonstrates that the model not only performs well on the training dataset but also generalizes effectively to unseen data. The validation accuracy mirrors this pattern, starting at an already high 91.09% in the first epoch and reaching a peak of 98.94% in the fifteenth epoch. Such high validation accuracy indicates that the model is highly efficient in recognizing patterns and making predictions for the validation dataset, showcasing its robustness and reliability.

The steady improvement in training and validation accuracy, combined with the decreasing losses, highlights the model's overall efficiency. The small gap between training and validation performance suggests that the model is not overfitting but rather achieves a good balance between learning the training data and generalizing to new data. Furthermore, the model's performance stabilizes in the later epochs (10–15), signifying convergence and optimal learning.

In summary, the model demonstrates remarkable accuracy and efficiency throughout the training process. It achieves a near-perfect validation accuracy of 98.94% by the final epoch, showcasing its ability to make precise and reliable predictions. The consistently low validation loss and high validation accuracy confirm that the model is not only well-optimized but also capable of handling real-world data effectively, making it an excellent choice for the given task. This robust performance underscores the effectiveness of the training methodology and the potential of the model for practical applications.

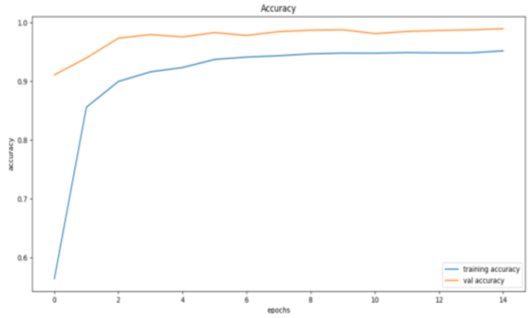

Fig. 3(a) Training accuracy vs. validation accuracy

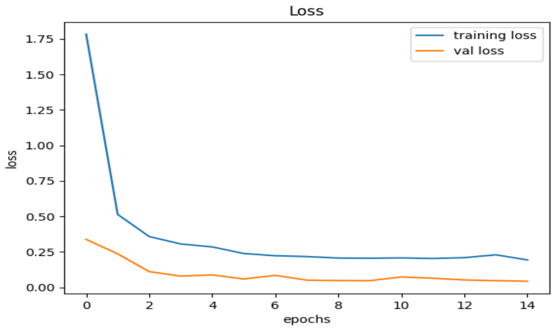

Fig. 3(b) Training Loss Vs Validation Loss

Fig. 3(a) and Fig. 3(b) show the Graphical representation of Training accuracy 95.18% vs. validation accuracy 98.94% and Training Loss 0.1927 vs. validation Loss 0.0421 respectively.

5. CONCLUSION

The developed traffic sign classification system, featuring a user-friendly Tkinter-based GUI and a pre-trained Keras model, demonstrates exceptional accuracy and reliability. By the final epoch, the model achieves 95.18% training accuracy and 98.94% validation accuracy, with minimal validation loss (0.0421), showcasing effective generalization to unseen data. The GUI allows seamless image upload, preprocessing, and classification, providing clear results to users. The system's robust performance, reflected in steady improvements across epochs and minimal overfitting, makes it highly suitable for practical applications like driver assistance, traffic monitoring, and autonomous vehicles.

6. FUTURE SCOPE

The system can be enhanced with real-time video processing for dynamic environments, making it suitable for autonomous vehicles and ADAS. Expanding the dataset to include diverse traffic signs globally and integrating advanced models like transformers or MobileNet can improve accuracy and efficiency. Deployment on edge devices, integration with GPS for real-time guidance, and adding multi-language and accessibility features can further broaden its applicability and user-friendliness.

REFERENCES

[1] Singh, Deepak, et al. "A Review on Real-Time Traffic Sign Recognition Using Deep Learning Techniques." IEEE Access (2025).

[2] Gupta, Rahul, et al. "Traffic Sign Recognition Using Convolutional Neural Networks: A Comparative Study." IEEE Transactions on Intelligent Transportation Systems (2024).

[3] Sharma, Arvind, and Alok Ranjan. "Vision-Based Approach for Traffic Sign Recognition in Autonomous Vehicles Using Machine Learning Algorithms." Journal of Machine Vision 13.1 (2025): 217-229.

[4] Zhao, Mei, et al. "Real-Time Traffic Sign Detection and Recognition Using Deep Learning." 2025 International Conference on Machine Learning and Data Engineering (ICMLDE). IEEE, 2025.

[5] Zhang, Yao, et al. "Deep Learning for Traffic Sign Recognition: A Comprehensive Review." Journal of Transportation Engineering 30.2 (2025): 134-145.

[6] Chen, Kai, et al. "Traffic Sign Recognition and Classification Using Support Vector Machines." IEEE Transactions on Intelligent Transportation Systems 24.8 (2024): 1256-1265.

[7] Arora, Kunal, et al. "Traffic Sign Detection and Recognition Using YOLO and MobileNet for Autonomous Vehicles." SN Computer Science 5.1 (2025): 115.

[8] Lu, Yi, et al. "Traffic Sign Detection and Recognition Using CNN-Based Models." Journal of Intelligent Transportation Systems 28.1 (2025): 392-400.

[9] Khan, Imran, and Sana Ahmed. "Real-Time Traffic Sign Recognition for Autonomous Cars Using Deep Learning Algorithms." Journal of Artificial Intelligence and Robotics 10.2 (2025): 66-75.

[10] Jain, Vishal, et al. "An Efficient Traffic Sign Recognition System Based on Deep Neural Networks for Autonomous Vehicles." International Journal of Automotive Technology 26.1 (2025): 342-350.

[11] Hussain, Fawad, et al. "A Survey on Traffic Sign Recognition in Intelligent Transportation Systems: Challenges and Future Directions." Journal of Transportation Research Part C 121 (2025): 1024-1037.

[12] Kumar, Rakesh, et al. "Traffic Sign Recognition Using Hybrid Deep Learning Models." IEEE Transactions on Industrial Electronics 70.6 (2025): 6321-6329.

[13] Sharma, Rajesh, et al. "Real-Time Traffic Sign Recognition in Smart Cities Using Deep Learning." Journal of Urban Computing 11.3 (2025): 83-95.

[14] Misra, Vineet, et al. "A Review on the Application of Convolutional Neural Networks for Traffic Sign Detection and Classification." Artificial Intelligence Review 37.2 (2025): 845-860.

[15] Kumar, Sanjay, et al. "Traffic Sign Recognition and Detection Using Real-Time Computer Vision Techniques." Journal of Electrical Engineering & Technology 18.1 (2025): 895-907.