International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04171
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Real-Time Traffic Sign Recognition Using CNN and Flask-Based Web Application

[1] Sumit K. Lokhande, [2] Pritish P. Bisne

[1] Sumit K. Lokhande Master of Computer Application, Trinity Academy Of Engineering, Pune, India
[2] Pritish P. Bisne of Master of Computer Application, Trinity Academy Of Engineering, Pune, India

---***---

Abstract - This research presents a real-time Traffic Sign Recognition (TSR) system combining deep learning and web technologies to enhance road safety. The model uses Convolutional Neural Networks (CNNs) for accurate sign classification and is trained on the GTSRB dataset. A Flask-based web interface allows real-time detection using webcam input. The system achieves 98.90% test accuracy with minimal latency, making it practical for intelligent vehicle applications.

Key Words**:** Traffic Sign Recognition, CNN, Flask, GTSRB, Real-Time Detection, OpenCV, AI, Deep Learning, Web App, Automation

1.INTRODUCTION

Traffic signs provide crucial visual information for maintaining road safety and order. Recognizing these signs automatically is vital for the development of Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. Traditional TSR systems used manual image processing, which struggled under varying conditions. Deep learning, particularly CNNs, offers a more reliable solution by learning directly from image data.

This study introduces a TSR system trained on the GTSRB dataset and deployed through a Flask-based web application. The model processes real-time webcam input and identifies multiple signs with high accuracy. The system is designed for ease of use, scalability, and real-world performance.

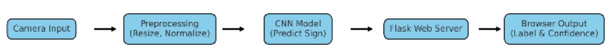

Figure 1: Flowchart for working of the Application

2. LITERATURE REVIEW

Early TSR systems relied on color segmentation and shape analysis. Machine learning approaches like SVM and KNN improved accuracy using handcrafted features. However, they required complex tuning.

With the rise of CNNs, TSR performance improved significantly. CNNs automatically learn features, improving reliability across varied conditions. Datasets like GTSRB enabled benchmarking of deep learning models. Recent work also integrates TSR with real-time applications using Flask, OpenCV, and lightweight CNN architectures.

3. METHODOLOGY

The GTSRB dataset is used, containing over 50,000 labeled images across 43 classes. Images are resized and normalized. Data augmentation is applied to prevent overfitting.

The model uses a CNN with convolutional, pooling, dropout, and dense layers. It is trained using Adam optimizer and categorical cross-entropy loss. Performance is monitored using accuracy and loss metrics.

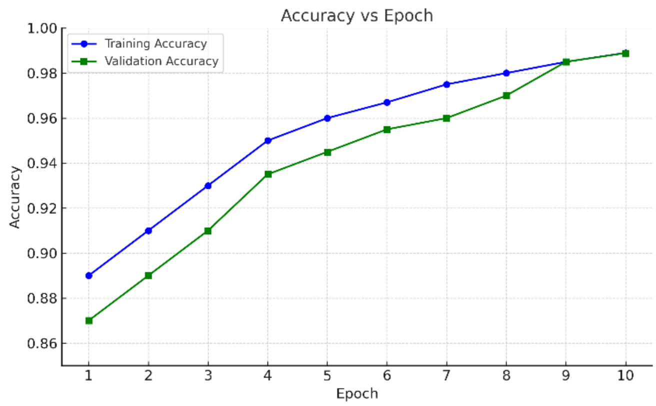

Figure 2: Accuracy vs Epoch Graph

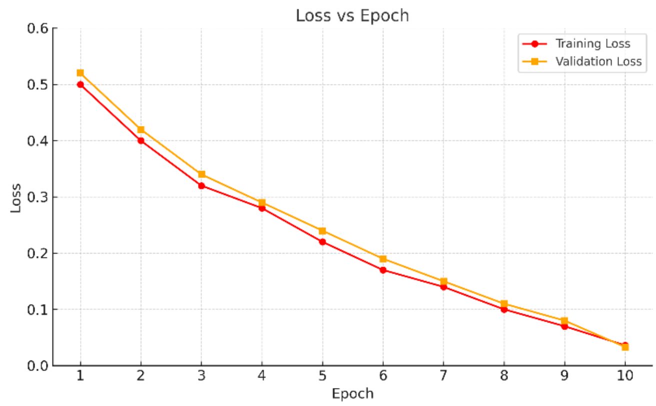

Figure 3: Loss vs Epoch Graph

The trained model is saved and loaded into a Flask web app. Real-time video input from OpenCV is processed for prediction.

4. SYSTEM ARCHITECTURE

The system has five key components: input handler, preprocessing, CNN classifier, Flask server, and OpenCV-based webcam interface. Input can be live video or images. Preprocessing prepares the data. The CNN model performs classification.

Flask handles user interaction, while OpenCV streams video. Predictions are displayed with confidence levels. The architecture supports quick, interactive recognition.

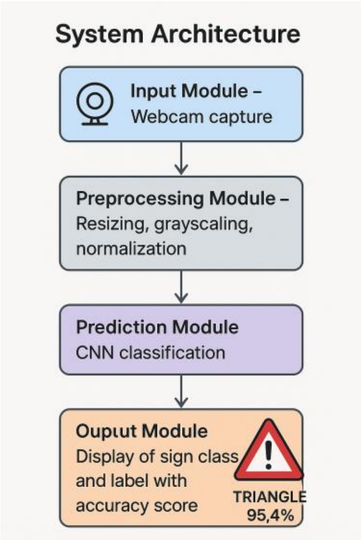

Figure 4: System Architecture

5. MODEL PERFORMANCE

The model showed consistent results across all classes. It achieved 98.90% test accuracy and 98.89% validation accuracy. Final test loss was 0.0360 and validation loss 0.0331.

Training and validation accuracy improved steadily over epochs. Confusion matrix and class-wise accuracy confirm minimal misclassification.

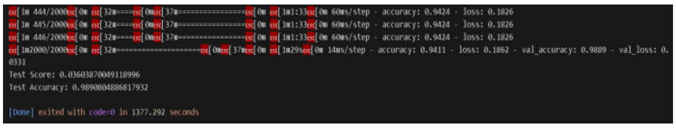

Figure 5: Training Per Epoch:

6. RESULTS

The system accurately identified signs from real-time webcam streams. Detection latency was under 150 ms. The Flask app displayed predictions and confidence scores instantly.

Tests under various conditions showed robust performance. Screenshots from the interface confirmed real-time classification.

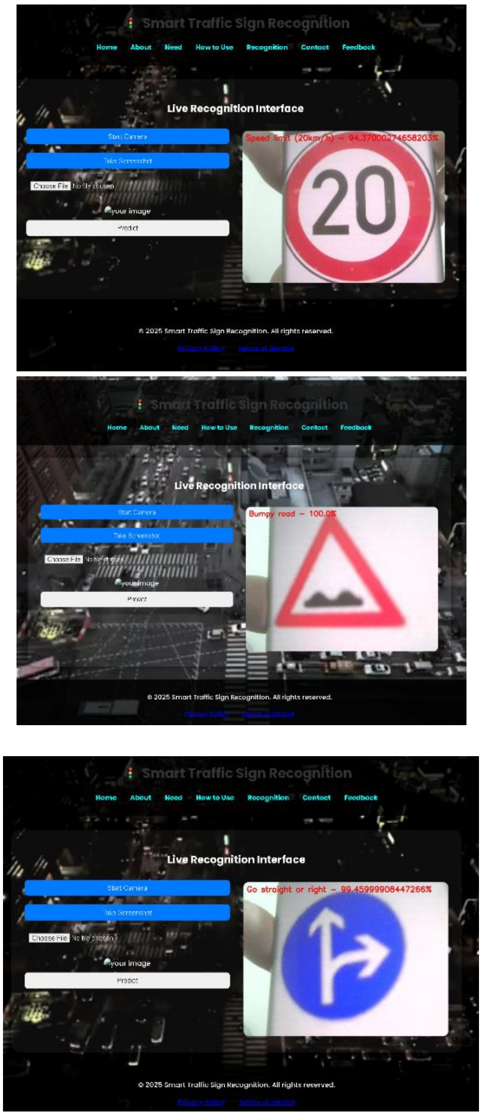

Figure 6: Real Time Recognition Result

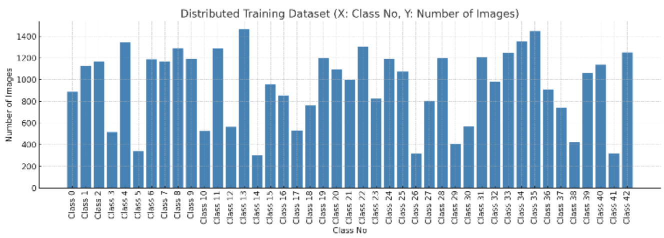

Figure 7: Distribution of training Data Set

CONCLUSIONS

A CNN-based TSR system was developed using the GTSRB dataset and deployed with Flask and OpenCV. It achieved high accuracy and real-time performance.
The system is suitable for ADAS and educational tools. Future work may involve mobile deployment or audio alerts.

ACKNOWLEDGEMENT

We thank the Department of MCA, Trinity Academy of Engineering, Pune, for their support. We also acknowledge peers and mentors for their helpful insights and feedback..

REFERENCES

1. Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 32, 323-332.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
3. Chollet, F. (2017). Deep Learning with Python. Manning Publications.
4. Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
5. OpenCV Documentation: https://docs.opencv.org/
6. Flask Documentation: https://flask.palletsprojects.com/

BIOGRAPHIES

Sumit K. Lokhande is a PG student in the Department of MCA at Trinity Academy of Engineering, Pune. His research interests include computer vision, deep learning, and real-time web applications.

Pritish P. Bisne is a Guide in the Department of MCA at Trinity Academy of Engineering, Pune. He has expertise in machine learning, data science, and intelligent systems.