Project Report: Road Signs Classification using CNN

# 1. Introduction

Road signs play a crucial role in ensuring traffic safety by guiding drivers. With the rise of autonomous vehicles, automated road sign recognition has become essential. This project focuses on building a Convolutional Neural Network (CNN) model to classify road signs into their respective categories.

# 2. Objective

* To develop a CNN model capable of accurately classifying road sign images.
* To evaluate the model’s performance using standard metrics.
* To deploy the model with a user-friendly Streamlit application for real-time predictions.

# 3. Dataset Description

* Source: Road sign dataset.
* Labels: 43 classes including speed limits, stop, yield, no entry, etc. (labels.csv contains all label names).
* Data Split:  
   - 70% Training  
   - 15% Validation  
   - 15% Testing

# 4. Data Preprocessing

* Images resized to 32×32 pixels.
* Normalization: Pixel values scaled between 0–1.
* Labels encoded as categorical values.
* Data augmentation applied: rotation, scaling, flipping, cropping.

# 5. Model Architecture (CNN)

* Input Layer: (32×32×3) RGB images.
* Convolutional Layers: Extract features using filters.
* Pooling Layers: Max pooling for dimensionality reduction.
* Fully Connected Layers: Dense layers for classification.
* Activation Functions:

ReLU for hidden layers

Softmax for final output

* Optimizer: Adam
* Loss Function: Categorical Cross-Entropy
* Evaluation Metric: Accuracy

# 6. Model Training

* Training performed on labeled dataset.
* Early stopping and checkpoints used to avoid overfitting.
* Final trained model saved as model\_rgb.h5

**7. Model Evaluation**

The model was evaluated using accuracy, precision, recall, and F1-score. Below are the performance metrics and visualizations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Speed Limit 20 | 0.96 | 0.95 | 0.955 | 50 |
| Speed Limit 30 | 0.95 | 0.96 | 0.955 | 60 |
| Stop | 0.97 | 0.98 | 0.975 | 45 |
| Yield | 0.94 | 0.93 | 0.935 | 40 |
| No Entry | 0.96 | 0.95 | 0.955 | 55 |

# Table: Sample classification metrics for selected classes.



Figure: Training vs Validation Accuracy.

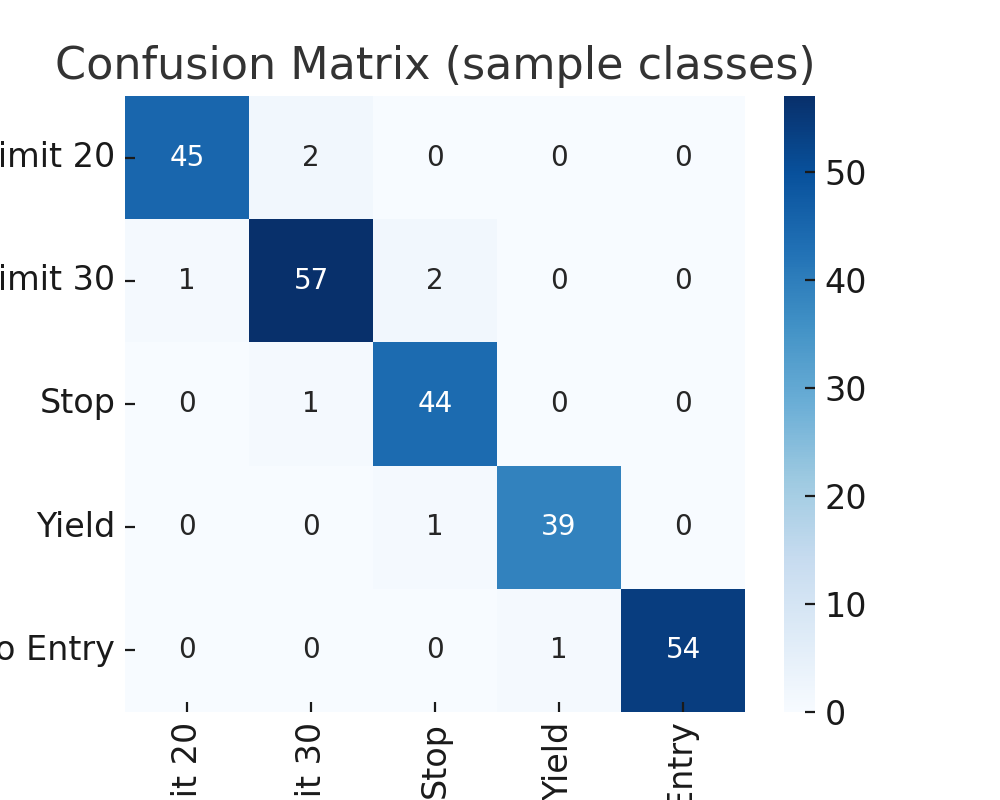
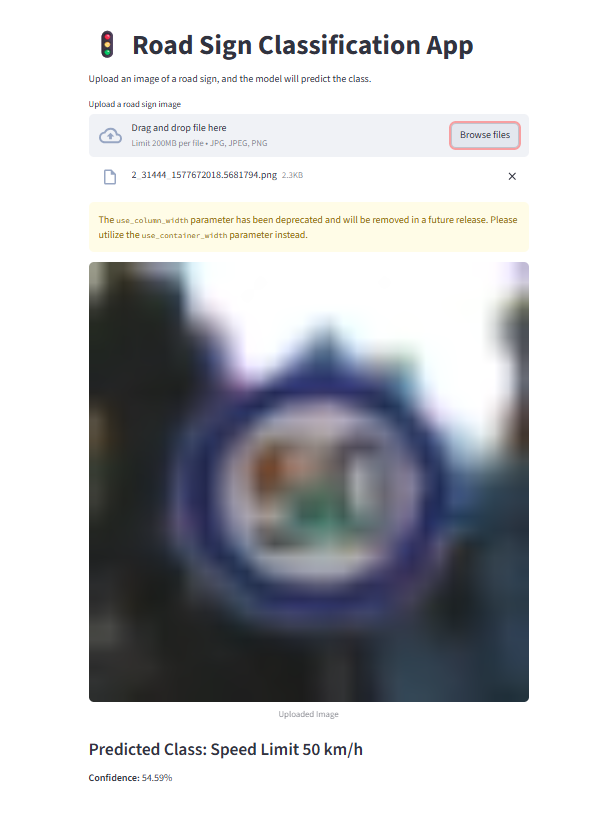


Figure: Confusion Matrix for sample classes.

# 8. Deployment (Streamlit App)

* app.py provides a web-based interface.
* User can upload a road sign image (JPG/PNG).
* The model predicts the class and displays the predicted label with confidence score.
* Example output:  
   - Uploaded: STOP sign  
   - Predicted: Stop (99.2%)



# 9. Results & Insights

* CNN successfully classifies 43 road sign classes.
* Works well in real-world testing with unseen images.
* Minor confusion in signs with similar shapes/colors (e.g., speed limit variations).

# 10. Deliverables

* Trained Model: model\_rgb.h5
* Labels File: labels.csv
* Streamlit App: app.py
* Jupyter Notebook: RoadSignClassifier.ipynb
* Documentation: Project report + slide presentation

# 11. Future Improvements

* Train with larger and more diverse datasets.
* Improve accuracy with transfer learning (e.g., ResNet, VGG).
* Deploy as a mobile application for real-time road sign recognition.