Traffic Sign Detection Using Keras and CNN

A PROJECT REPORT

Submitted by

Yogeshwar Yadav

Enrollment No: AJU/231956

in partial fulfilment for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS (BCA) with Specialization in Artificial Intelligence

Arka Jain University

Internship Duration: 7 June 2025 - 7 July 2025

ABSTRACT

Traffic sign detection is a crucial component of intelligent transportation systems and autonomous vehicles. This project presents a deep learning-based approach using Convolutional Neural Networks (CNNs) implemented with Keras and TensorFlow to automatically classify traffic signs from road images. The German Traffic Sign Recognition Benchmark (GTSRB) dataset is used for training and evaluation. The model achieves a test accuracy of approximately 95%, demonstrating its effectiveness and reliability. The project also features a user-friendly Streamlit interface for image upload and real-time prediction, making the solution practical and accessible.

1. INTRODUCTION

1.1 Background

With the rise of autonomous vehicles and advanced driver assistance systems (ADAS), robust traffic sign recognition has become essential. Deep learning, and particularly CNNs, offer a powerful solution by learning features directly from data.

Traffic Sign Detection Using Keras and CNN

1.2 Motivation

- Improve driver assistance and reduce human error.
- Enable autonomous vehicles to interpret road signs accurately.
- Support smart city infrastructure and traffic management.

1.3 Scope

- Develop a CNN model using Keras for traffic sign classification.
- Train and evaluate the model on the GTSRB dataset.
- Deploy the model via a Streamlit app for easy user interaction.

2. PROBLEM DESCRIPTION

Classify traffic sign images into one of 43 predefined classes using a deep learning model. Key challenges include variability in sign appearance, occlusion, low resolution, and the need for real-time inference.

3. OBJECTIVES

- Build and train a robust CNN using Keras and TensorFlow.
- Achieve high classification accuracy on the GTSRB dataset.
- Provide a user-friendly interface for model inference.

4. METHODOLOGY

4.1 Data Acquisition

- Dataset: German Traffic Sign Recognition Benchmark (GTSRB)
- Classes: 43
- Images: Over 50,000

4.2 Data Preprocessing

- Resized images to 30x30 pixels
- Normalized pixel values
- Encoded labels as one-hot vectors
- Applied data augmentation (rotation, shift, zoom, etc.)

4.3 Model Development

- Custom CNN architecture with convolutional, pooling, dropout, and dense layers
- Activation: ReLU; Output: Softmax
- Compiled with categorical cross-entropy loss and Adam optimizer

4.4 Model Training & Evaluation

- Trained for 30 epochs with batch size 32
- Used validation split and class weighting
- Evaluated using accuracy, precision, recall, F1-score, and confusion matrix
- Achieved ~95% test accuracy

4.5 User Interface

- Developed a Streamlit app for image upload and prediction
- Displayed predicted sign names and confidence scores

5. IMPLEMENTATION

5.1 Technologies Used

- Python 3.x
- Keras with TensorFlow backend
- Streamlit for frontend UI
- NumPy, Pandas, Matplotlib

5.2 System Architecture

- Data Loader: Loads and splits data
- Preprocessing Module: Resizes, normalizes, and augments images
- Model Module: Defines and compiles the CNN
- Trainer: Handles training and validation
- Evaluator: Computes metrics and plots
- UI: Streamlit app for user interaction

6. RESULTS AND ANALYSIS

- Test Accuracy: ~95%
- Precision/Recall/F1-score: High across most classes
- Training Curves: Show stable convergence with minimal overfitting
- User Interface: Allows easy image upload and displays predictions in real time

7. DEPLOYMENT

- The trained model is integrated into a Streamlit web app for demonstration and use.
- Users can upload images and instantly receive predictions with confidence scores.

8. LIMITATIONS & FUTURE WORK

- Performance can degrade under severe occlusion or poor lighting.
- Small or distant signs are harder to classify.
- Future work: integrate real-time detection, extend to video streams, deploy on edge devices.

9. CONCLUSION

This project demonstrates the effectiveness of deep learning for traffic sign recognition. The modular design, robust performance, and user-friendly interface make it a strong foundation for further research and real-world deployment.

REFERENCES

- Stallkamp, J., et al. "The German Traffic Sign Recognition Benchmark: A multi-class classification competition." IEEE IJCNN (2011).
- Keras Documentation: https://keras.io/
- TensorFlow Documentation: https://www.tensorflow.org/

ATTACHMENTS

- Source code (data_loader.py, preprocess.py, model.py, train.py, app.py)
- Trained model file
- Sample results and screenshots
- Documentation (README, usage guide)

Project Duration: 7 June 2025 - 7 July 2025

Submitted by: Yogeshwar Yadav

Enrollment No: AJU/231956

BCA (AI), Arka Jain University