Automated Traffic Sign Classification

Using Python & Deep Learning

Presented by

Kashish Singh(63)

Gayatribala Balamurgan(64)

Shreya Yadav(70)

Guided by

Prof. Prajakta Gotarne

November 27, 2025

Introduction

- Traffic signs play a crucial role in driving, informing drivers of road conditions and potential hazards.
- These signs are typically simple in shape and have eye-catching colours, making them valuable for driver-assistance systems, highway maintenance, and self-driving vehicles.
- Traffic sign recognition involves two main steps: traffic sign detection and traffic sign classification.
- Although popular in driver assistance systems, traffic sign recognition poses challenges due to size variations, colour deterioration, and partial occlusion.
- This project aims to develop a deep learning model using Python and CNN to recognize and classify traffic signs from the TSR dataset.
- By leveraging the power of CNNs and optimizing model performance, we strive to achieve high accuracy and reliability in traffic sign recognition.

Literature Survey

Paper Title	Authors	Observations
TRAFFIC SIGN RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS, (Year-2018)	Ervin Miloš, Aliaksei Kolesau, Dmitrij Šešok	The paper shows that using CNNs with preprocessing, batch normalisation, dropout, and data augmentation significantly improves traffic sign recognition accuracy, achieving up to 94%. The proposed approach outperforms many standard models but suggests future work with spatial transformer layers for even better results.
Flexible, High Performance Convolutional Neural Networks for Image Classification	Dan C. Ciresan et al.	Observation 1: The paper focuses on using (CNNs) for image classification, which aligns well with our project's goal. Observation 2: The authors highlight the benefits of using GPUs for efficient CNN training. Observation 3: The paper emphasizes the importance of deep CNN architectures with multiple layers for achieving high accuracy in image classification tasks.
The German Traffic Sign Recognition Benchmark: A multi-class classification competition (Year-2011)	Johannes Stallkamp, Marc Schlipsing, Jan Salmen, Christian Igel	GTSRB is a valuable benchmark: Its large, diverse dataset and precomputed features make it a valuable resource for traffic sign recognition research. Class imbalance and baseline performance: Addressing class imbalance and comparing algorithms to baseline methods could provide deeper insights. Evolving field: Traffic sign recognition has likely progressed since 2011, and newer datasets might be available.

Literature Survey

Paper Title	Authors	Observations
Design Principles of Convolutional Neural Networks for Multimedia Forensics(2019)	B. Bayar and M.C. Stamm	The paper introduces a CNN framework tailored for multimedia forensics by incorporating constrained convolutional layers to detect forensic artifacts like image manipulations. It emphasizes suppressing irrelevant features while enhancing forensic patterns, significantly improving manipulation detection accuracy.
Traffic Sign Recognition Using Deep Learning(2022)	T. Venkata Sumanth, Vinod Reddy, Leela Balaji, Kanhaiya Kumar	The paper focuses on using the German Traffic Sign Recognition Benchmark (GTSRB) dataset and employs HSV color space for preprocessing and Generalized Hough Transform for localization. Classification is done via CNNs, addressing challenges like noise, lighting, and environmental variations. Accuracy is evaluated using F-score metrics.

Problem Statement

- How can we develop a deep learning model using Python and CNN to accurately recognize and classify traffic signs from the dataset, achieving high accuracy and reliability, to support the development of intelligent transportation systems and improve road safety?
- Design and implement automated traffic sign recognition system to accurately detect and classify traffic signs in various environmental conditions.

Existing System

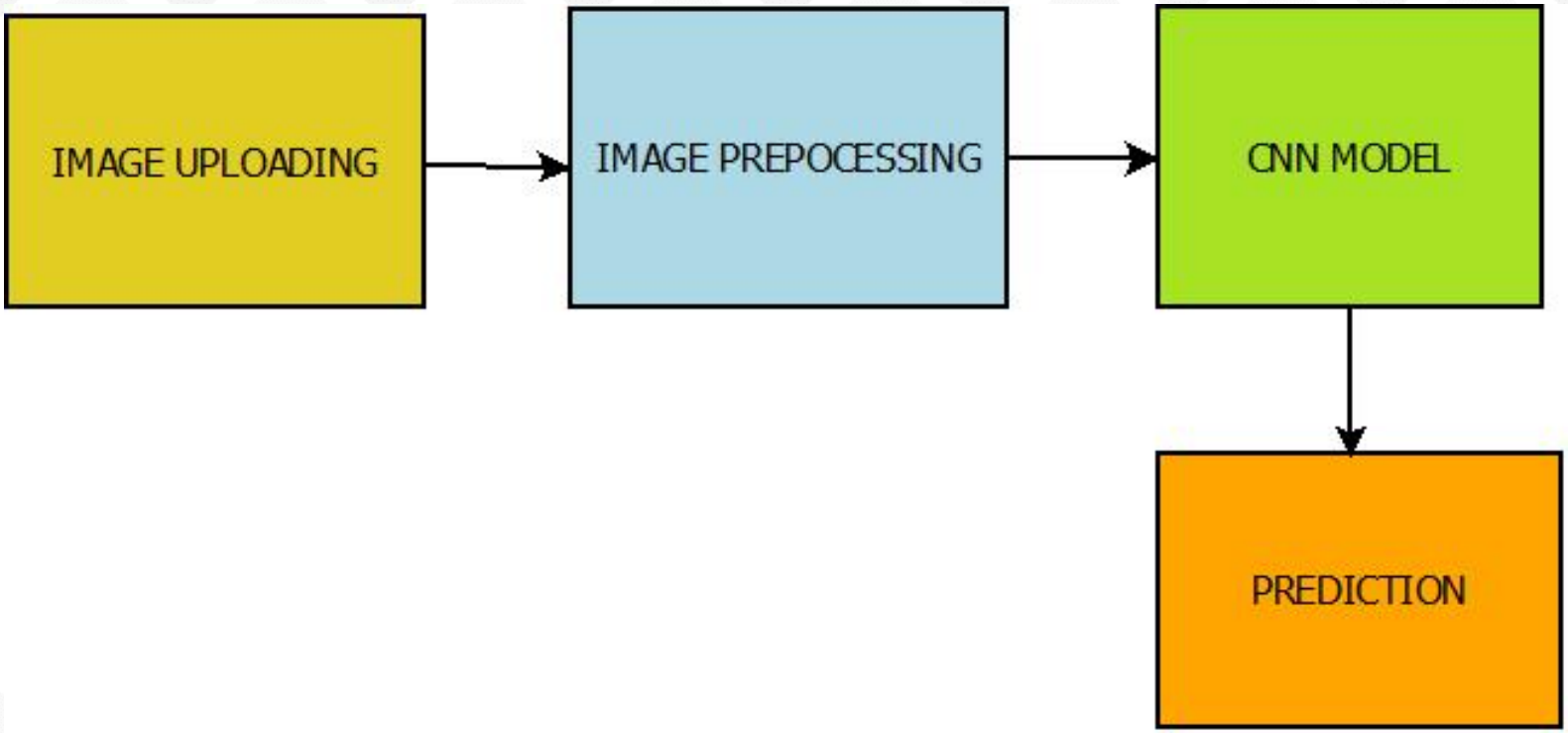


Figure 1 – Existing Model.

Proposed System

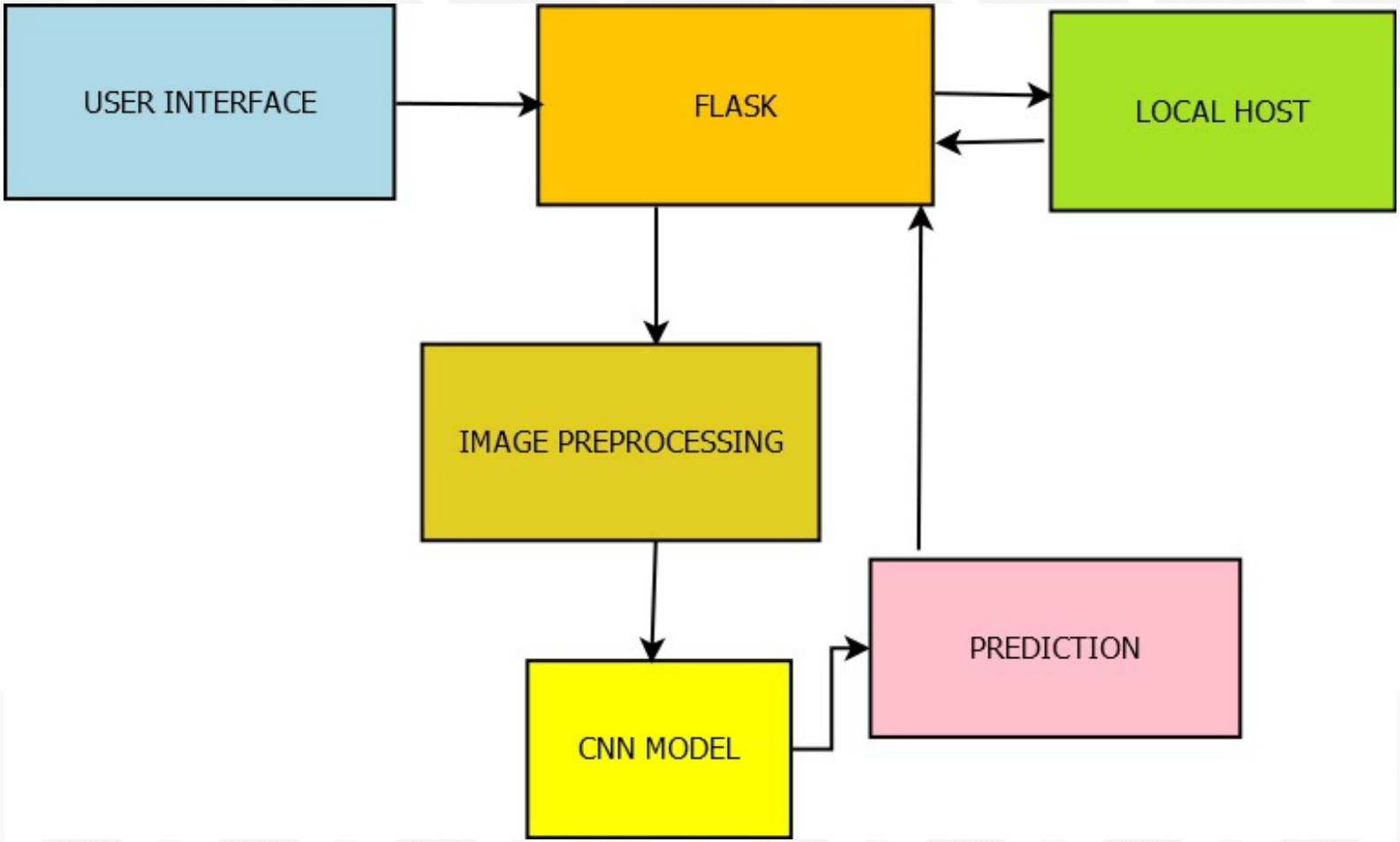


Figure 2 – Proposed Model.

Workflow Diagram

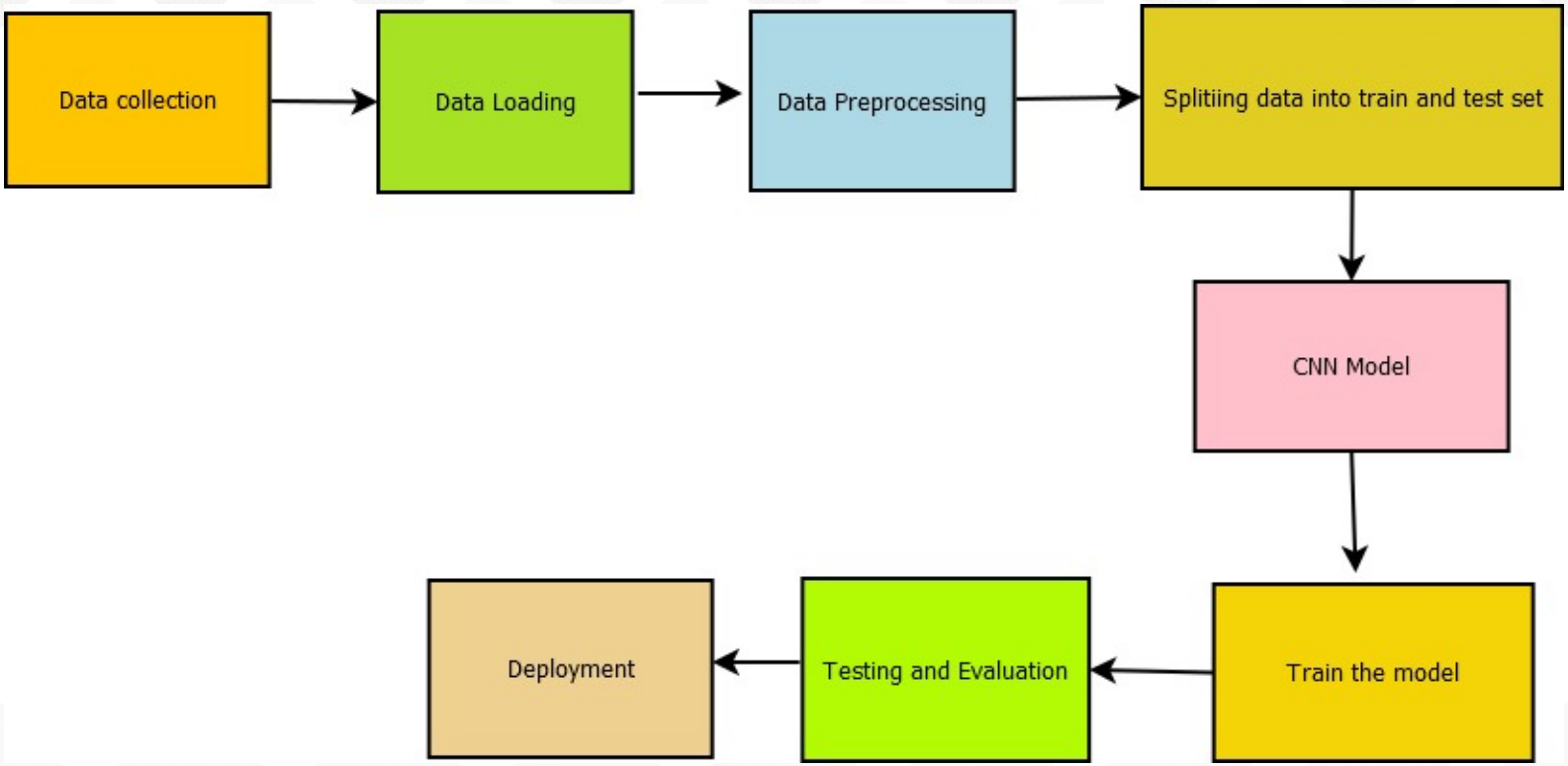


Figure 3 – Workflow diagram.

Methodology

- Step 1: Data Collection
  We gathered a diverse set of images showcasing various traffic signs in different conditions.
- Step 2: Data Preprocessing
  Preprocessing is crucial for improving model performance.
  Images are resized to 30x30 pixels and data is processed.
- Step 3: Data splitting
  Data is split into 80:20 ratio for training and testing set.

- Step 4: CNN Model
  The model is built using Tensorflow and Keras framework. Further, it is compiled and trained for 20 epochs with a batch size of 32.
- Step 5: Testing& Evaluation
  Evaluating accuracy and loss of train data and then testing the data available in the test folder.
- Step 6: Deployment
  The trained model is integrated into a Flask web application and deployed on a cloud platform.

Hardware & Software Requirements

- Hardware Requirements:

  Processor:Minimum quad-core processor(Intel i5 or equivalent)
  RAM:Minimum 8GB RAM storage.

- Software Requirements:

  Programming Language:Python 3.7 or above
  Libraries and Frameworks:

  Tensorflow/ Keras
  Flask
  Pandas & Numpy
  Matplotlib
  PIL(pillow)
  Kagle API
  IDE:Google Colab

Applications

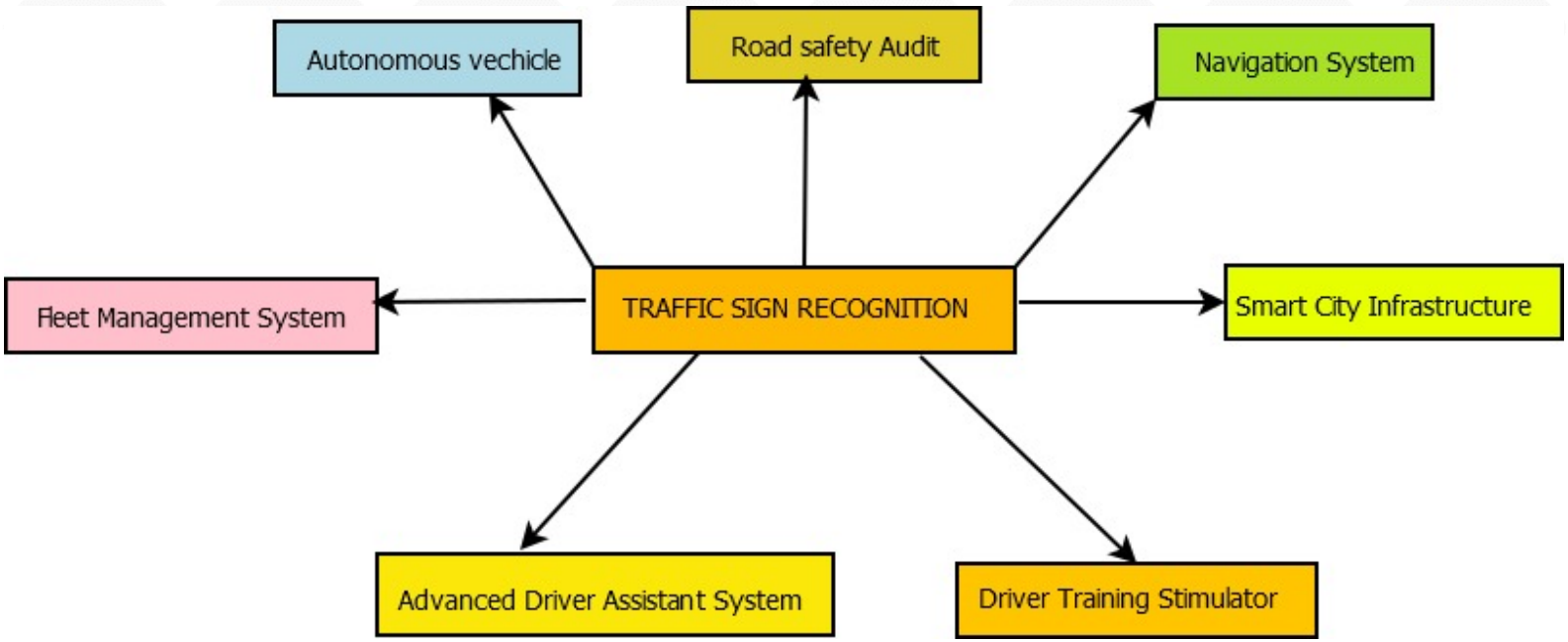


Figure 4 – Applications.

Future work / Scope

- Model Enhancement and Optimization
- Real-time Processing and Deployment
- Dataset Expansion and Diversification
- Integration with Other Systems
- User Interface and Experience Improvement
- Compliance and Safety Features
- Research and Development

Implementation & Results

project phase 2 (1).ipynb

File Edit View Insert Runtime Tools Help All changes saved

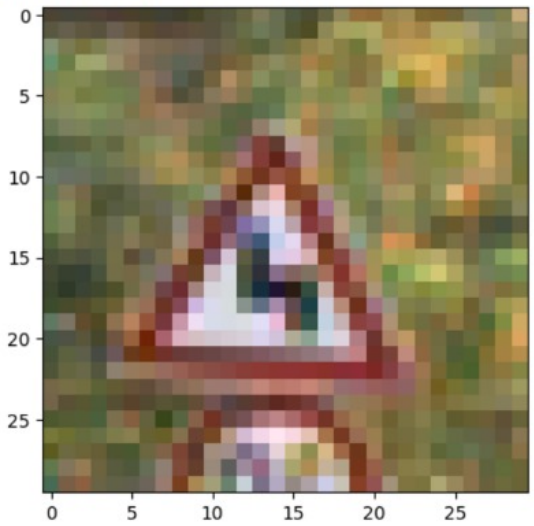
Share

+ Code + Text

Gemini

```
[44] plot,prediction = test_on_img("/content/traffic_sign_dataset/Test/00016.png")
     s = [str(i) for i in prediction]
     a = int("".join(s))
     print("Predicted traffic sign is: ", classes[a])
     plt.imshow(plot)
     plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
Predicted traffic sign is:  Double curve
```



0s completed at 4:30 PM

Figure 5 – Implementations & Results.

Implementation & Results

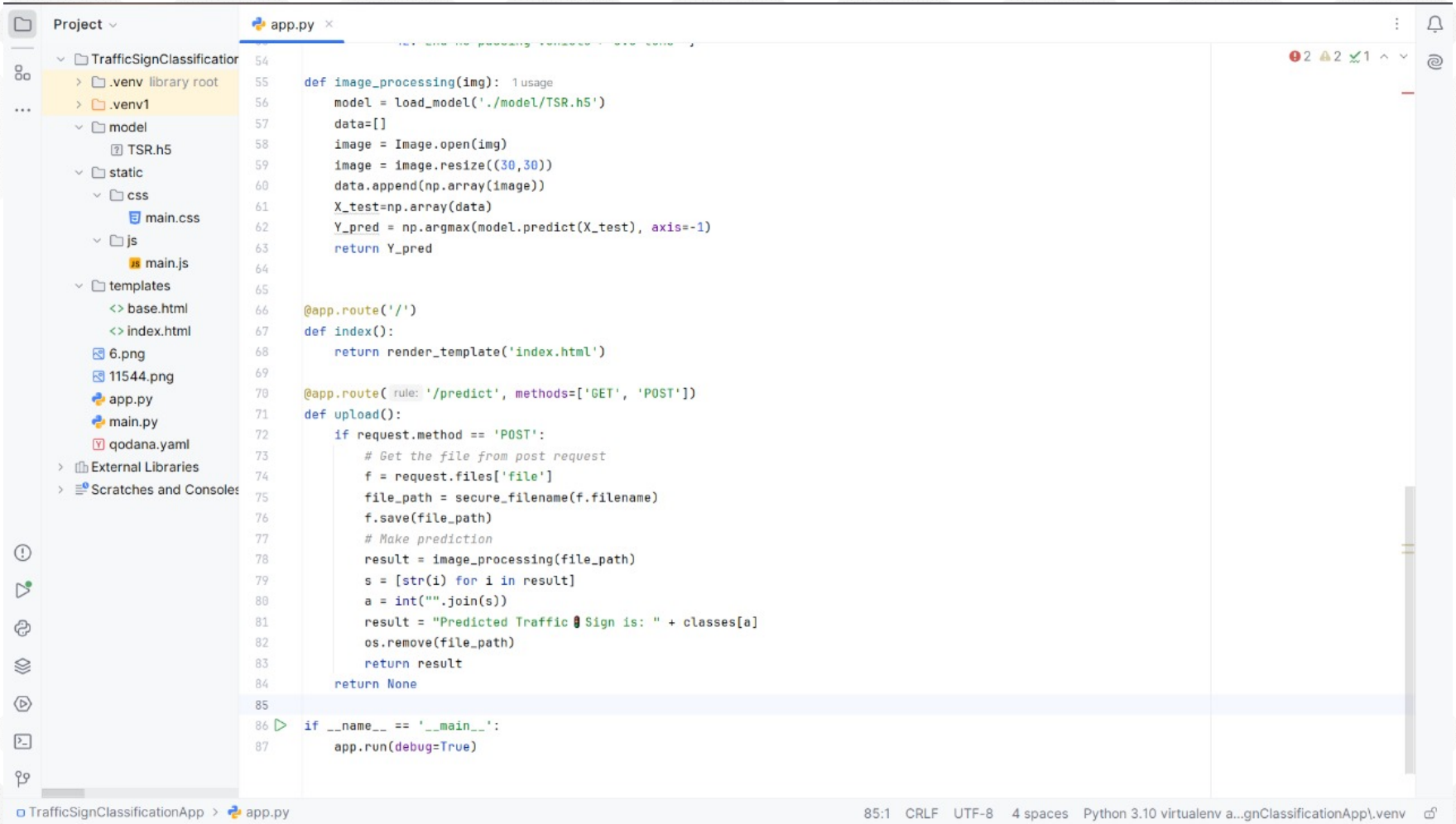


Figure 6 – Implementations & Resultsg.

Implementations & Results

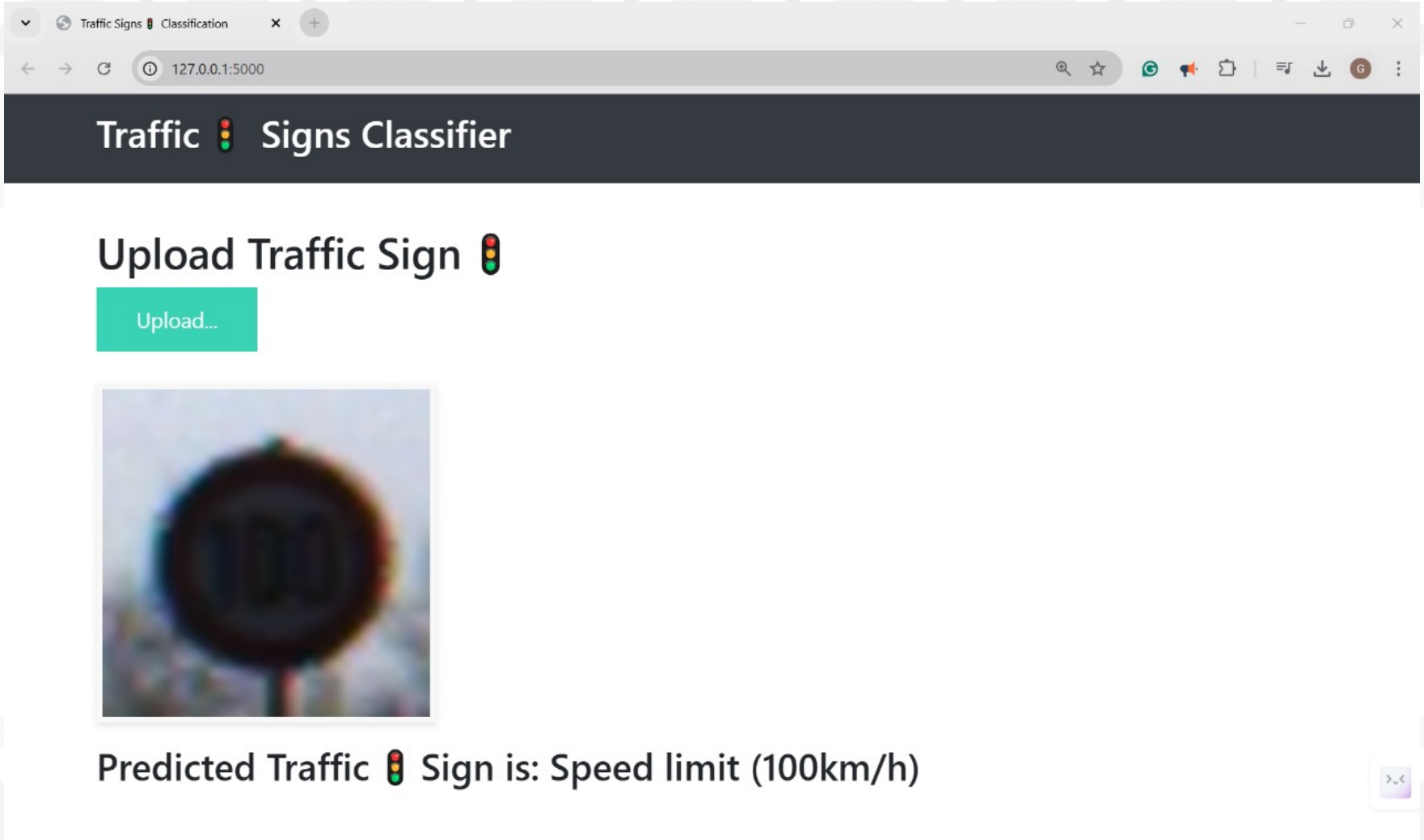


Figure 7 – Implementations & Results.

References

1] J. Stallkamp, M. Schlipsing, and J. Salmen, The German Traffic Sign Recognition Benchmark: A Multi-Class Classification Competition, 1st ed., Bochum, Germany: Institut für Neuroinformatik, Ruhr-Universität Bochum, 2011.
2] E. Miloš, A. Kolesau, and D. Šešok, "Traffic sign recognition using convolutional neural networks," in Science – Future of Lithuania, vol. 10, Vilnius: Vilnius Gediminas Technical University, 2018.
3] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark, 1st ed., Bochum, Germany: Institut für Neuroinformatik, Ruhr-Universität Bochum, 2013.

4] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, Flexible, High Performance Convolutional Neural Networks for Image Classification, 1st ed., Zurich, Switzerland: ETH Zurich, 2012.
5] B. Bayar and M. C. Stamm, Design Principles of Convolutional Neural Networks for Multimedia Forensics, 1st ed., New York, NY,
USA: Springer, 2019.
6] S. Jung, U. Lee, J. Jung, and D. H. Shim, Real-time Traffic Sign Recognition System with Deep Convolutional Neural Network, 1st ed., Seoul, South Korea: Seoul National University, 2020.

Thank You

Thank you.