UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Real-Time Traffic Sign Recognition Using OpenCV and Convolutional Neural Networks

Student : Salajan Denisa Patricia

Year/Series/Group : 3rd-EA-2031/3

Professors: Mircea Giurgiu, Ioana Maeva Carbunescu

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Mini-Project requirements

The theme of this mini-project is the design and implementation of a real-time traffic sign recognition system using image processing and machine learning techniques. The system detects traffic signs from a live camera stream and classifies them using a trained neural network model.

Contents

Mini-Project requirements 2
1. Introduction 3
Problem description 3
Proposed solution 3
Original aspects 3
2. Theoretical Background 4
2.1 Computer Vision 4
2.2 Convolutional Neural Networks 4
2.3 GTSRB Dataset 4
3. Implementation 4
3.1 Hardware 4
3.2 Software 4
3.3 CNN Training 5
3.4 Real-Time Recognition Pipeline 5
3.5 FPS (Frames Per Second) 5
4. Experimental Results 5
.... 6
5. Original contribution 6
6. Conclusions 6
7. Bibliography 7

1. Introduction

Traffic sign recognition represents an important research topic in the field of intelligent transportation systems and driver assistance technologies. Correct identification of traffic signs contributes to increased road safety and assists drivers by providing timely warnings and information.

The purpose of this mini-project is to develop a **real-time traffic sign recognition system** using a standard laptop webcam, without requiring specialized hardware. The system is capable of detecting traffic signs in a video stream, classifying them into predefined categories, and displaying the result together with a confidence score.

Problem description

The main challenges addressed in this project are:

- detecting traffic signs under varying lighting conditions,
- correctly classifying multiple sign types,
- maintaining stable predictions in real time,
- avoiding false classifications.

Proposed solution

To solve these challenges, the project combines:

- **OpenCV** for real-time image acquisition and sign detection,
- a **Convolutional Neural Network (CNN)** trained on the GTSRB dataset for classification,
- confidence thresholding and temporal smoothing to increase robustness.

Original aspects

Compared to a basic classification approach, the project introduces:

- an **UNKNOWN class** based on confidence thresholds,
- **temporal averaging** of predictions across frames,
- contrast normalization for webcam images,
- a modular design separating detection and recognition stages.

2. Theoretical Background

2.1 Computer Vision

Computer vision enables machines to interpret visual data. OpenCV is a widely used library that provides tools for image processing, video capture, and object detection. In this project, OpenCV is used to:

- capture video frames from a webcam,
- detect potential traffic sign regions,
- crop and visualize detected signs.

2.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning model specifically designed for image data. CNNs use convolutional layers to automatically extract features such as edges, shapes, and textures. The final layers perform classification using learned representations.

2.3 GTSRB Dataset

The German Traffic Sign Recognition Benchmark (GTSRB) dataset contains over 50,000 labeled images of traffic signs divided into 43 classes. It includes variations in illumination, scale, and orientation, making it suitable for training robust classifiers.

3. Implementation

3.1 Hardware

- Laptop with integrated webcam
- No external sensors or hardware accelerators

3.2 Software

- Python 3.11
- OpenCV
- TensorFlow / Keras
- NumPy

3.3 CNN Training

The CNN model was trained using the GTSRB dataset. Images were resized to 32×32 pixels and normalized. The network architecture includes convolutional layers, pooling layers, dropout for regularization, and dense layers for classification.

Training was performed for 10 epochs, achieving high validation accuracy.

3.4 Real-Time Recognition Pipeline

The real-time system follows these steps:

1. Capture frame from webcam
2. Detect sign candidate using OpenCV
3. Crop region of interest (ROI)
4. Apply contrast normalization
5. Resize and normalize input
6. Predict class probabilities using CNN
7. Apply temporal smoothing
8. Display label, confidence, and bounding box

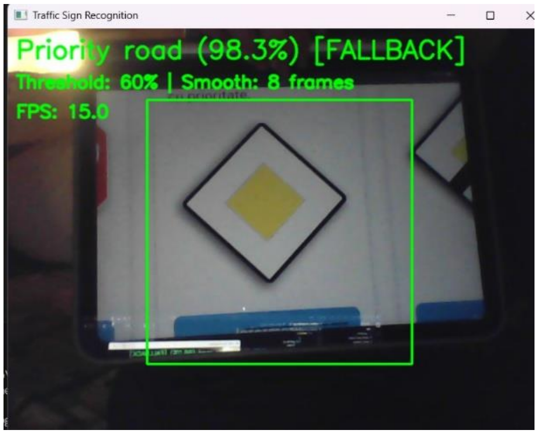
3.5 FPS (Frames Per Second)

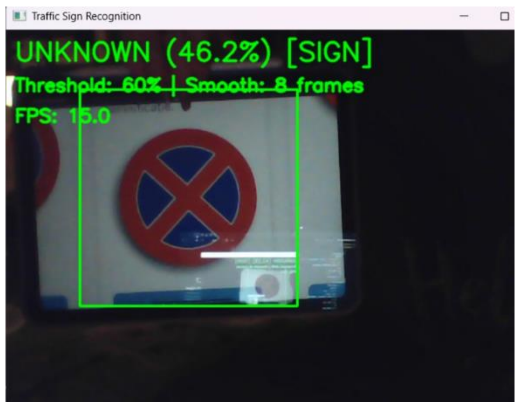
FPS indicates how many frames are processed per second. It measures the real-time performance of the system. A higher FPS means smoother and faster video processing. In this project, FPS is calculated dynamically and displayed on screen.

4. Experimental Results

- Training accuracy exceeded **98%**
- Validation accuracy reached approximately **99%**
- Real-time recognition works reliably for clear, well-lit signs
- Confidence threshold successfully prevents false positives
- Temporal smoothing reduces flickering predictions

The system performs well using only a standard webcam, demonstrating the feasibility of real-time traffic sign recognition on low-cost hardware.





5. Original contribution

The originality of the project lies in improving a standard traffic sign recognition system by introducing a confidence threshold that allows the system to output an **UNKNOWN** class when predictions are uncertain. In addition, prediction results are stabilized by averaging outputs over multiple consecutive frames, reducing fluctuations in real-time operation. Image preprocessing was adapted for webcam conditions to improve robustness against lighting variations. The region of interest is dynamically expanded to preserve important sign details. Finally, the modular structure of the application allows easy extension and further improvements.

6. Conclusions

This mini-project successfully demonstrates a complete traffic sign recognition system combining computer vision and deep learning techniques. The modular design allows easy extension and improvement. By introducing confidence-based decision making and temporal smoothing, the system achieves stable and reliable performance in real-time conditions.

7. Bibliography

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel,
"The German Traffic Sign Recognition Benchmark," *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, San Jose, CA, USA, 2011, pp. 1453–1460, doi: 10.1109/IJCNN.2011.6033395.

[2] Y. LeCun, Y. Bengio, and G. Hinton,
"Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton,
"ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012, pp. 1097–1105.

[4] I. Goodfellow, Y. Bengio, and A. Courville,
Deep Learning, Cambridge, MA, USA: MIT Press, 2016.

[5] R. Szeliski,
Computer Vision: Algorithms and Applications, New York, NY, USA: Springer, 2010.

[6] M. Abadi et al.,
"TensorFlow: A system for large-scale machine learning," *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Savannah, GA, USA, 2016, pp. 265–283.