PROJECT ABSTRACT

Traffic Sign Recognition using CNN

Convolutional neural network that classifies 43 German traffic sign types from the GTSRB benchmark, with augmentation, per-class F1 analysis and a demo UI — delivered as a built-to-order student project with code, notebook, report, PPT and viva kit.

01. Title

Traffic Sign Recognition using CNN

02. Introduction / Background

Driver-assistance systems and autonomous vehicles depend on reading road signs reliably. The task is simple for humans and hard for machines: the same sign appears blurred at speed, backlit, rain-streaked, tilted or partially occluded, and the dangerous confusions are between near-twins such as speed-limit 30 and 80. The German Traffic Sign Recognition Benchmark (GTSRB), introduced at IJCNN 2011, made this a rigorous academic task: 43 classes and over 50,000 cropped real-world images with genuine variation. Convolutional networks are the natural tool — they learn hierarchical visual features directly from pixels.

03. Problem Statement

Manual sign-reading does not scale to driver-assistance workloads, and template matching fails under real lighting and blur variation. The project addresses this by training a custom CNN on GTSRB with road-condition augmentation, evaluated with per-class F1 and a confusion matrix so that the dangerous near-twin confusions are studied openly rather than averaged away.

04. Objectives

- Build the GTSRB data pipeline: 39,209 training and 12,630 test images across 43 classes, resized to 48x48 with normalization.
- Design a custom CNN (convolution → batch-norm → ReLU → max-pool blocks, dropout, 43-way softmax) and train it with augmentation simulating road conditions.
- Reach a design target of ~95%+ test accuracy, with the notebook logging accuracy, per-class F1 and the confusion matrix on the held-out split.
- Build a demo UI that classifies uploaded sign photos with top-3 ranked predictions.
- Deliver the complete student kit: source code, notebook, exported weights, report, PPT and viva Q&A document.

05. Proposed Methodology

GTSRB crops are resized to 48x48 RGB and normalized. Training applies augmentation — random rotation, brightness jitter, blur and translations — that mimics road photography. The custom CNN (three convolutional blocks with batch normalization and dropout, then dense layers) trains with the Adam optimizer and categorical cross-entropy, with early stopping on validation accuracy. Evaluation on the held-out 12,630-image test split reports accuracy, per-class precision/recall/F1 and the confusion matrix; a misclassification gallery feeds the report's error analysis.

06. System Architecture / Working

GTSRB loader → resize/normalize → augmentation → CNN training with per-epoch logging → early-stopped best checkpoint → held-out evaluation (accuracy, per-class F1, confusion matrix) → SavedModel export → Flask demo: upload → identical preprocessing → top-3 ranked predictions with confidence bars. The notebook is the single source of truth for every reported metric.

07. Hardware / Software Requirements

- Software: Python 3.10, TensorFlow/Keras, OpenCV, NumPy, pandas, scikit-learn, Matplotlib, Seaborn, Flask, Jupyter.
- Hardware: training is feasible on a modern CPU; a GPU shortens it substantially.
- Dataset: GTSRB (public download, ~50,000 images).

08. Expected Outcomes

A trained 43-class sign classifier with notebook-measured test accuracy (design target ~95%+), per-class F1 table and confusion matrix; an augmentation study showing robustness to blur and lighting shifts; exported weights; and a demo that classifies sign photos in seconds. Every number in the report is measured on the student's own held-out split.

09. Applications

- Driver-assistance prototyping and ADAS coursework demonstrations.
- Teaching material for CNNs: convolutions, pooling, augmentation, confusion analysis.
- Reference build for other fine-grained traffic-object tasks.
- Starting point for the documented extension: detection/localization in full road scenes.

10. Future Scope

- Full-scene sign detection (YOLO/SSD) feeding crops to this classifier.

- Fine-tuning on Indian road-sign photography for local deployment.
- Video-rate inference optimization for dashboard-camera feeds.
- Comparison study against MobileNetV2 transfer learning.