

# Traffic Sign Classification using CNN

Road Sign Recognition for Autonomous Systems

Prathmesh Giram  
PG-DBDA, CDAC Kharghar

---

## Abstract

This project presents a Convolutional Neural Network (CNN)-based system for classifying German road signs using the GTSRB dataset. It also features a user-friendly web application for real-time prediction and supports applications in autonomous vehicles and traffic management systems.

## 1 Introduction

Traffic signs play a crucial role in road safety and autonomous driving. Recognizing these signs accurately is essential for developing intelligent transportation systems. This project leverages deep learning, particularly CNNs, to classify 43 types of traffic signs from images.

## 2 Dataset

**Dataset:** German Traffic Sign Recognition Benchmark (GTSRB)

- 43 traffic sign classes
- RGB images resized to 64x64 pixels
- Data normalized using ImageNet statistics

## 3 Model Architecture

**Model:** Custom CNN built using PyTorch

- Input: 64x64 RGB image
- Layers:
  - Conv2D (3→32) → ReLU → MaxPool
  - Conv2D (32→64) → ReLU → MaxPool
  - Conv2D (64→128) → ReLU → MaxPool

- Flatten  $\rightarrow$  Linear(8192 $\rightarrow$ 256)  $\rightarrow$  ReLU  $\rightarrow$  Dropout  $\rightarrow$  Linear(256 $\rightarrow$ 43)
- Loss Function: CrossEntropyLoss
- Optimizer: Adam

## 4 Environment Setup

**Conda Environment:** roadsign-env

Listing 1: Conda Environment Setup

```
name: roadsign-env
channels:
  - defaults
  - pytorch
  - nvidia
dependencies:
  - python=3.10
  - pytorch
  - torchvision
  - torchaudio
  - pytorch-cuda=11.8
  - pip
  - pip:
    - opencv-python
    - matplotlib
    - tqdm
    - pandas
    - scikit-learn
    - ultralytics
    - pyttsx3
```

## 5 Training and Evaluation

- Batch Size: 32
- Epochs: 5
- Learning Rate: 0.001
- Accuracy Achieved: (Insert % here after evaluation)
- Model Saved At: outputs/best\_model.pth

## 6 Web Application

**Built Using:** Streamlit

- Upload traffic sign images
- Real-time classification with confidence score
- Displays class name and prediction
- *(Note: UI screenshot not shown here)*

## 7 Project Structure

- `app.py` – Web application logic
- `labels.py` – Class names
- `outputs/best_model.pth` – Trained model
- `src/model.py` – CNN model definition
- `src/train.py` – Model training script
- `src/loaders.py` – Data loaders and preprocessing
- `src/eval.py` – Evaluation script for accuracy and loss
- `src/utils.py` – Utility functions (e.g., plotting, logging)

## 8 Future Work

- Add Grad-CAM visualization
- Enable real-time webcam input
- Enhance performance with data augmentation
- Deploy on cloud (AWS/HuggingFace)

## Acknowledgment

Special thanks to the GTSRB dataset creators and open-source communities of PyTorch and Streamlit.

*Built with by Prathmesh Giram*