Project Title

Traffic Sign Recognition : GTSRB Dataset

Project Overview

This project involves developing a traffic sign recognition system using deep learning. The system will assist in autonomous vehicle navigation and driver assistance by accurately classifying different types of traffic signs. Students will use a Convolutional Neural Network (CNN) model trained on the GTSRB dataset to achieve this task and will build an interactive user interface to enhance usability.

Objective

The main objective of this project is to create a system that can classify traffic signs from images. Students will build a complete pipeline, including data preprocessing, model training, and user interaction through a UI that allows users to upload traffic sign images and receive a classification result.

Tools and Libraries

- Python
- TensorFlow or PyTorch (for model building and training)
- OpenCV (for image processing)
- tkinter or Flask (for building the user interface)
- NumPy, Pandas, Matplotlib (for data handling and visualization)

Dataset

- **Dataset**: German Traffic Sign Recognition Benchmark (GTSRB) Dataset
- **Description**: The dataset contains images of German traffic signs, including more than 40 different classes of signs.

Classes

The dataset includes the following types of traffic signs:

- Speed limits (various limits such as 30 km/h, 50 km/h, etc.)
- Warning signs (e.g., curve ahead, children crossing)
- Prohibitory signs (e.g., no overtaking, stop)
- Informative signs (e.g., pedestrian crossings, roundabouts)

Project Components

1. **Data Preprocessing**:
   - Load the GTSRB dataset.
   - Resize images, normalize pixel values, and apply data augmentation (e.g., rotations, flips) to improve model generalization.

2. **Classification**:
    - Build a Convolutional Neural Network (CNN) model to classify the traffic signs.
    - Train the model on the preprocessed dataset and evaluate its accuracy.
    - Use transfer learning to improve performance if needed.
3. **User Interaction**:

    basic UI: button to upload from test dataset, visualization (+accuracy, predictions)

    - Develop a user-friendly interface using tkinter or Flask.
    - Allow users to upload images of traffic signs and display the classification result with confidence scores. (Optional)
4. **Documentation for Project**:

    -Abstract section

    - Include detailed documentation explaining the project overview, methodology, results, and future work.

    -Comment code for each method... (ex purpose of soft max layer,...)
    -Class diagram, UML diagram

Submission Guidelines

-Final: result - accuracy, how many data have trained

1. **Executable Code**:
    - Submit the entire project as executable python script.

    OOP:
    classes, methods, functions
2. **Documentation**:
    - Provide a comprehensive documentation of the project.
3. **Code Structure**:
    - Ensure the code is well-structured, with proper comments and modular functions.
4. **GUI**:
    - The GUI should be fully functional and demonstrate all key features.

    upload a random image from the internet to see