Traffic Sign Detection System

Introduction:

Using cutting-edge machine learning technology, the Traffic Sign Detection System is intended to recognize and categorize different types of traffic signs. The system examines and categorizes traffic signs in the provided photos using an advanced convolutional neural network (CNN).

Summary of the Week

This week, we made significant progress in several areas of our Traffic Sign Detection System project. Our team focused on model training, frontend development, and backend integration this time, setting the groundwork for further improvements and future work.

Task Progress:

Model Training: We kept adjusting parameters and streamlining training procedures to improve the Convolutional Neural Network (CNN) model. We got significant gains in performance metrics and accuracy due to our efforts.

Frontend Development: Much work has been done to improve the frontend

interface to give users a simple experience. Our UI modifications were guided by doing initial user testing and incorporating feedback. We got this from our classmates while discussing the project with each other.

Backend Development: The backend infrastructure was gradually enhanced to interact with the CNN model. Our top priorities were timely answer delivery, accurate model predictions, and compelling image upload management. As we could get more out of this project with accurate and swift results, it would be an excellent success for the time provided in this course.

Challenges Faced:

Resource Limitations: Despite constraints in time and computing resources, we maximized efficiency and productivity through streamlined workflows and focused efforts. It took much work to get new resources, and we are still focusing on new data types that can help us with the resources and proper results.

Integration Complexity: The complex interaction between frontend, backend, and CNN model components required careful coordination and thorough debugging, which produced a well-coordinated system design.

Progress Table:

Future Work:

Completing Model Training: To get the best possible accuracy and robus
tness, resolve any outstanding issues, maximize performance metrics, a
nd finish training the CNN model,

Model Evaluation: To verify the trained model's effectiveness in actua
l situations and guarantee reliable projections, thoroughly examine an
d test the model.

Frontend Refinements: Based on user feedback, implement more UI improv
ements emphasizing accessibility features, transparent feedback system
s, and straightforward navigation.

Backend Optimization: Optimise system performance and resource usage b
y fine-tuning backend operations for improved scalability, reliability
, and response time.

Integration and Testing: Thoroughly evaluate the system's operation, r
esolve compatibility problems, and fully integrate the frontend, backe
nd, and CNN model components.

Deployment Preparation: To ensure a seamless transition to operational
use, prepare the system for deployment by creating user manuals, docu
mentation, and deployment plans.

References:

Welcome to Flask — Flask Documentation (3.0.X). (n.d.-b). https://flask.palletsprojects.com/en/3.0.x/

Github Repo - https://github.com/shreyash5965/trafficsign