Traffic Sign Recognition System

Graduation Project Proposal

Digital Egypt Pioneers Initiative (DEPI)

Track: Artificial Intelligence / Machine Learning

Prepared by:

Abdelrahman Mohamed Abdelhaleem	(Team Leader)
Youssef Magdy Abd El-Aty	(ML Engineer)
Peter Ashraf Nathan	(ML Engineer)
Omar Mohamed El-Sayed	(ML Engineer)
Abdelrahman Mostafa Mohamed	(ML Engineer)
Adham Mansour	(ML Engineer)

1. Project Description

This project aims to develop an intelligent **Traffic Sign Recognition (TSR)** system that can automatically detect and classify traffic signs from images using computer vision and deep learning. The model will be trained on the **German Traffic Sign Recognition Benchmark (GTSRB)** dataset to identify various traffic signs such as speed limits, warnings, and prohibitions.

Applications include:

- Autonomous and semi-autonomous driving systems.
- Advanced driver assistance systems (ADAS).
- Road safety monitoring and automation.

The system enhances driving safety and reduces human error in traffic sign interpretation.

2. Group Members and Roles

Name	**Role**	**Responsibilities**
Abdelrahman Mohamed Abdelhaleem	Team Leader	Coordinates project tasks, manages Git repository, designs and trains CNN model, ensures integration between modules.
Youssef Magdy Abd El-Aty	ML Engineer	Handles data preprocessing, augmentation, and dataset cleaning.
Peter Ashraf Nathan	ML Engineer	Reviews papers, tunes hyperparameters, evaluates model performance, and implements YOLO/CNN.
Omar Mohamed El-Sayed	ML Engineer	Develops Flask/FastAPI API for model deployment and ensures scalability.
Abdelrahman Mostafa Mohamed	ML Engineer	Builds user interface for testing traffic sign images and prepares the final proposal.
Adham Mansour	ML Engineer	Prepares technical documentation, progress reports, and final presentation.

3. Objectives

- Preprocess and clean the GTSRB dataset for optimal model training.
- Build and train a CNN for accurate classification of traffic signs.
- Evaluate model performance using accuracy, precision, recall, and F1-score.
- Deploy the model using Flask or FastAPI with a graphical interface.
- Integrate YOLO for real-time detection and classification.
- Develop a UI/UX dashboard for testing images or videos.
- Optimize the system for scalability and low-latency prediction.

4. Tools and Technologies

- **Programming Language:** Python
- **Libraries & Frameworks:** TensorFlow, Keras, OpenCV, NumPy, Pandas, Matplotlib, Scikit-learn, YOLOv8
- **Deployment Tools:** Flask or FastAPI, Docker (optional)
- **UI/UX Tools:** Streamlit or Tkinter (desktop), HTML/CSS/JS (web)
- **Dataset:** German Traffic Sign Recognition Benchmark (GTSRB)
- **Version Control:** Git and GitHub
- **Development Environment:** Jupyter Notebook, Visual Studio Code
- **Hardware:** GPU-enabled (Google Colab or local GPU)

5. Milestones and Deadlines

Milestone	Description	Deadline
M1: Dataset Understanding & Setup	Download, explore, and split dataset.	Week 1
M2: Data Preprocessing	Cleaning, normalization, and augmentation.	Week 2
M3: Model Development	Design and train CNN architecture.	Week 3
M4: Model Evaluation	Analyze performance using validation and test data.	Week 4
M5: Deployment & YOLO Integration	Build web interface, API, and integrate YOLO model.	Week 5
M6: GUI Implementation & Testing	Develop GUI (Streamlit/Tkinter/Web), integrate and test system end-to-end.	Week 6
M7: YOLO Integration Completion	Train YOLO for detection and merge with CNN model.	Week 7

6. Key Performance Indicators (KPIs)

6.1. Data Quality

Metric	Target
Percentage of missing values handled	100%
Data accuracy after preprocessing	$\geq$98%
Dataset diversity (43 classes)	100% represented

6.2. Model Performance

Metric	Target
Model accuracy (F1-Score)	$\geq$ 96%
Prediction latency	$\leq$ 50 ms/image
Error rate (False Positive/False Negative)	$\leq$ 4%

6.3. Deployment and Scalability

Metric	**Target**
API uptime	$\geq 99\%$
Response time per request	≤ 200 ms
Real-time processing (video input)	≥ 20 FPS

6.4. Business Impact and Practical Use

Metric	**Target**
Reduction in manual sign detection effort	80%
Expected cost savings from automation	50%
User satisfaction (based on feedback)	$\geq 90\%$