Project Proposal

Project Title:

Night-Time Vehicle Detection Using YOLO-Based Deep Learning Models

Project Description:

This project aims to develop a robust vehicle detection system optimized for night-time and low-light conditions using the YOLOv8 architecture. The system will be trained on multiple datasets containing vehicle images captured under various lighting environments. Due to the scarcity of labeled night-time data, the project incorporates dataset merging, augmentation, and preprocessing strategies to improve generalization and accuracy.

The trained model will be evaluated on locally available test data and benchmarked across multiple key metrics such as accuracy, latency, and false detection rates. Ultimately, the goal is to build a scalable detection model suitable for real-world applications such as intelligent traffic surveillance, smart parking, and automated incident detection.

Group Members & Roles:

- Omar Mohamed Abdallah Ameen – Team Leader
- Seif Eldeen Hassan Mohamed Ahmed – Data collection
- Abdelrahman Mohamed Ramadan – Data Preprocessing and Augmentation
- Omar Mohamed Abdallah Ameen – Model Training and Optimization
- Ahmed Mohamed Abdelghany Abdelghafar – Evaluation and KPI Analysis
- Ahmed Yasser Ahmed Mohamed – Deployment and Integration

Team Leader:

Omar Mohamed Abdallah Ameen

Objectives:

1. Develop a YOLO-based vehicle detection model specifically optimized for night-time imagery.

2. Collect and preprocess diverse datasets to enhance representation of low-light scenarios.
3. Merge and label data effectively to overcome missing or incomplete annotations.
4. Train, validate, and fine-tune the YOLOv8 model for improved accuracy and reduced false detections.
5. Deploy the model as an inference API and test its real-time performance.
6. Evaluate model quality using data and performance KPIs to ensure practical applicability.

Tools & Technologies:

Python will be the main programming language used for this project. The deep learning model will be implemented with PyTorch using the YOLOv8 architecture provided by the Ultralytics framework. Data preprocessing will be conducted using OpenCV, NumPy, and Pandas, while visualization will rely on Matplotlib and Seaborn.

Annotation and label management will be handled using Roboflow or CVAT. GitHub will be used for version control. The deployment process will utilize FastAPI or Flask for creating a lightweight REST API, and optionally Docker for scalability. Model training will be conducted on GPU-enabled systems, either locally or through Google Colab.

Milestones & Deadlines:

1. **Dataset Collection and Cleaning:** Gather and preprocess daytime and night-time vehicle datasets, handle missing and inconsistent labels – *Week10*.
2. **Model Training:** Train the YOLOv8 model using the prepared dataset and fine-tune hyperparameters – *Week 13*.
3. **Model Evaluation:** Measure model accuracy, precision, recall, and F1-score – *Week 14*.
4. **Fine-tuning for Night Conditions:** Retrain or adapt the model using additional night-specific data – *Week 15*.
5. **Deployment and API Integration:** Develop and deploy an inference API to test model scalability – *Week 17*.
6. **Final Testing and KPI Evaluation:** Evaluate model performance against defined KPIs – *Week 18*.
7. **Report and Presentation Submission:** Submit final report and project presentation – *Week 20*.

KPIs (Key Performance Indicators):

Please specify the key metrics for measuring the success of your project based on the following aspects.

1. Data Quality

- Percentage of missing values handled: **98%**
- Data accuracy after preprocessing: **95%**
- Dataset diversity (representation of different categories): **85%**

2. Model Performance

- Model accuracy (Accuracy/F1-Score): **87%**
- Model prediction speed (Latency): **25 milliseconds**
- Error rate (False Positive/False Negative Rate): **13%**

3. Deployment & Scalability

- API uptime: **99%**
- Response time per request: **80 milliseconds**
- (If applicable) Real-time processing speed (e.g., FPS for video models): **20–25 FPS**

4. Business Impact & Practical Use

- Reduction in manual effort: **75%**
- Expected cost savings: **60%**
- User satisfaction: **85%**