Weekly Project: Traffic-Related Computer Vision Project

Presentation Day:

The project should be delivered and presented by the 5th of September at the latest..

Objective:

Students will solve a specific traffic-related problem using YOLO (You Only Look Once) for object detection or Optical Character Recognition (OCR) for text recognition. The project could involve detecting vehicles, recognizing license plates, identifying traffic signs, or other traffic-related tasks. By building YOLO for object detection or OCR for text extraction, students can develop a deep solution for improving traffic monitoring and management.

Project Overview:

Students can use YOLO architecture for real-time object detection or an OCR model for text recognition to solve a traffic-related computer vision problem or both. They will collect or use a traffic-related dataset, preprocess images or videos, and analyze the model's performance. The objective is to create a system that can detect and classify objects (e.g., vehicles, traffic signs) or recognize text (e.g., license plates) in real-time, providing valuable data for traffic management or enforcement.

Project Tasks:

Task 1: Problem Identification and Data Gathering

- Define a specific traffic-related problem that can be addressed using YOLO or OCR (e.g., vehicle detection, license plate recognition, traffic sign identification).
- Identify and collect a relevant image or video dataset containing objects of interest and text (e.g., license plates, street signs).
- Ensure the dataset is diverse and representative of real-world traffic scenarios.

Task 2: Exploratory Data Analysis (EDA)

- Preprocess images or videos (e.g., resizing, normalization, data augmentation) to make them suitable for YOLO or OCR.
- Visualize samples from the dataset to understand different classes and the quality of text for OCR or even the YOLO model.
- Split the dataset into training, validation, and test sets.

Task 3: Model Development and Training

- Implement the YOLO architecture for object detection or an OCR model for text recognition.
- Preprocess input data to ensure compatibility with YOLO (for object detection) or the OCR model (for text recognition).
- Train the models separately on their respective tasks and then integrate them into a unified pipeline (if chosen to develop both together).

Task 4: Model Evaluation and Selection

- Evaluate the performance of the YOLO model using standard evaluation techniques (e.g., mAP, precision, recall) for object detection tasks.
- Evaluate the OCR model's performance on text recognition tasks using metrics like accuracy, character error rate (CER), or word error rate (WER).
- Analyze the combined performance in the context of your chosen traffic-related problem (e.g., accuracy of detecting vehicles and reading license plates) if chosen to develop them both.

Task 5: Model Deployment (Optional but encouraged)

- Deploy the model (YOLO or OCR) in a simple web app or service (e.g., using Flask or FastAPI) to process traffic-related or videos.

Deliverable:

- A Jupyter Notebook documenting the entire workflow (data collection, preprocessing, model design, training, evaluation) focused on solving the traffic-related problem with YOLO or OCR.
- A presentation summarizing the problem, methodology, and results, including visual examples of the model's performance.

Team Collaboration and Task Assignment:

Each student should contribute to the project, and documentation should clearly indicate who was responsible for each task. Ensure that all tasks, from data gathering and preprocessing to model design and evaluation, are assigned and documented in the presentation.

Presentation Questions (These should help you, but they are not mandatory to answer. You can add if you need to clarify):

- What challenges did you face in developing the model, and how did you overcome them?
- Why did you choose the specific model for your issue?

- How does the integrated system perform compared to using each model independently? (if chosen to develop them both)
- What insights can be drawn from the system's misclassifications or incorrect text recognitions?
- How did deploying the model affect its usability in real applications?

Main Grading System:

Component	Description	Points
Project Idea Selection	Clarity, feasibility, and relevance of the chosen traffic-related problem using YOLO or OCR.	10
Data Relevance	How relevant is the data to solving the traffic-related problem with YOLO and OCR?	20
Data Collection & Preprocessing	Quality of data preprocessing for both object detection and text recognition tasks.	15
Model Selection & Implementation	Appropriateness of model choice, design, and implementation, including code quality and efficiency.	15
Model Training & Evaluation	Effectiveness of model training, tuning, and evaluation for the model.	10
Presentation	Quality of the presentation, including clear explanation of the problem, methodology, and results.	10
Team Collaboration & Contribution	Equal participation, clear task assignment, and documentation of who did each task.	10

Bonus Tasks for Extra Credit:

Bonus Task	Description	Bonus Points
Data Scraping or Manual Data Collection	If students scrape or collect data manually instead of using an existing dataset.	+5
Use of Single Shot Detector instead (SSD) or custom architecture instead of YOLO.	If students use SSD or customize the architecture.	+5
Custom OCR Implementation	If students implement a custom OCR model or significantly enhance an existing one.	+5
Deployment in a Cloud Environment	If students deploy the integrated model in a cloud environment or as a mobile application.	+10