Problem Statement

In the context of autonomous driving and advanced driver-assistance systems (ADAS), ensuring the safety of vehicles on the road is paramount. One critical aspect is the ability to detect and predict potential cut-ins, where a vehicle abruptly moves into the lane of the host vehicle, potentially leading to dangerous situations. This project aims to develop a system capable of real-time vehicle detection, distance estimation, and cut-in prediction using deep learning and computer vision techniques.

Unique Idea Brief (Solution)

Implement a comprehensive real-time system that leverages advanced deep learning and computer vision techniques to enhance the safety of autonomous and driver-assistance systems. This solution utilizes a pre-trained YOLO (You Only Look Once) model for robust and efficient detection of multiple vehicle types from input images. Once vehicles are detected, the system estimates their distances from the host vehicle using bounding box dimensions and known camera parameters.

To address potential safety hazards, the system analyzes consecutive frames to track vehicle movements and identify potential cut-ins. By calculating the Time-to-Collision (TTC) for vehicles within a critical distance threshold, the system can accurately predict imminent threats. This proactive approach allows for timely alerts and interventions, significantly enhancing the safety and responsiveness of autonomous driving and ADAS technologies.

Features Offered

1: Real-Time Vehicle Detection:

Utilizes a pre-trained YOLO (You Only Look Once) model to detect various vehicle types (cars, trucks, buses, motorbikes, bicycles) in real-time from input images.

2: Distance Estimation:

Accurately estimates the distance of detected vehicles from the host vehicle using bounding box dimensions and known camera parameters, enhancing situational awareness.

3: Cut-In Prediction:

Analyzes consecutive frames to track vehicle movements and identify potential cut-ins by calculating the Time-to-Collision (TTC) for vehicles within a critical distance threshold.

4: Time-to-Collision (TTC) Calculation:

Provides precise TTC calculations for vehicles posing an imminent threat, allowing for timely alerts and interventions.

5: Bounding Box Visualization:

Displays bounding boxes around detected vehicles, with color-coding to indicate potential hazards (e.g., red boxes for vehicles within a critical distance).

6: Scalability and Integration:

Designed for easy integration into existing autonomous driving and advanced driver-assistance systems (ADAS), with scalability to accommodate various sensor setups and configurations.

7: Robust Performance:

Ensures high accuracy and reliability in diverse driving scenarios, minimizing false positives and negatives to maintain system integrity and safety.

Processflow

1. Image Acquisition:

- Capture consecutive frames from a forward-facing camera mounted on the vehicle.

2. Preprocessing:

- Resize and normalize images for input into the YOLO model.
- Convert images to blob format suitable for deep learning inference.

3. Vehicle Detection:

- Feed preprocessed images into the YOLO model.
- Extract bounding boxes, class IDs, and confidence scores for detected vehicles.
- Apply non-max suppression to eliminate redundant bounding boxes.

4. Distance Estimation:

- Calculate the distance of each detected vehicle using the dimensions of the bounding boxes and the known camera parameters (focal length and real-world width of a vehicle).

5. Frame Analysis:

- Track detected vehicles across consecutive frames by matching bounding boxes based on Intersection over Union (IoU).
- Identify potential cut-ins by detecting significant lateral movements of vehicles relative to the host vehicle's trajectory.

6. Time-to-Collision (TTC) Calculation:

- Compute the TTC for each detected vehicle using the change in position across consecutive frames and the time interval between frames.
- Focus on vehicles within a critical distance threshold (e.g., 3.5 meters) for potential hazards.

7. Visualization and Alerting:

- Draw bounding boxes around detected vehicles, with color-coding to indicate potential threats (e.g., red for vehicles posing a cut-in risk).

- Display distance and TTC information above or around the bounding boxes.

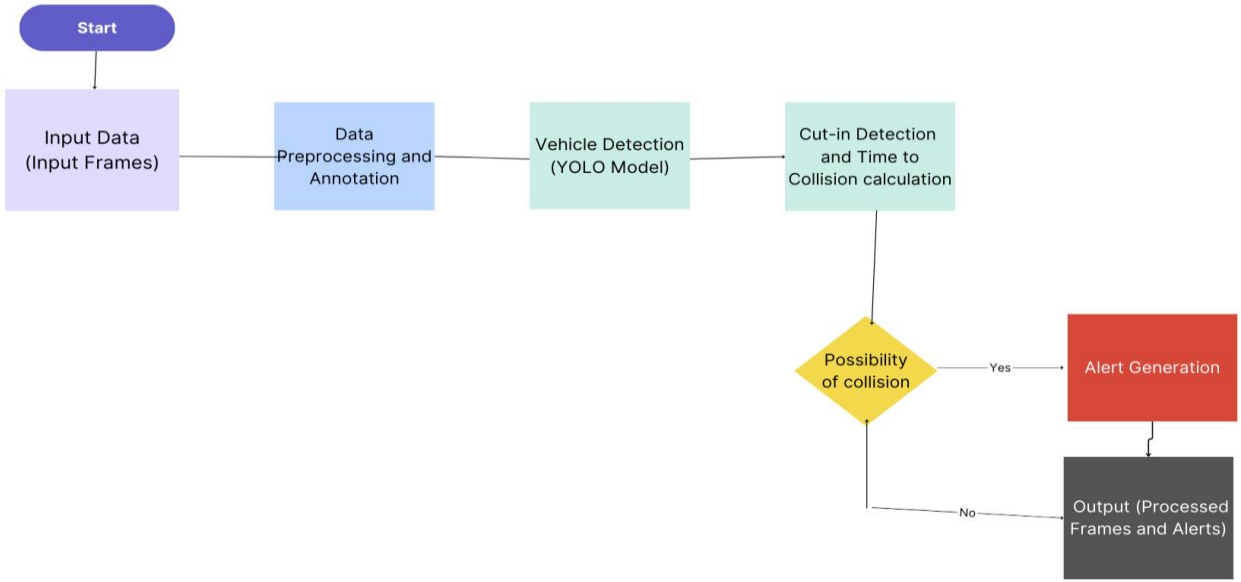
- Generate visual or auditory alerts if a vehicle poses an imminent collision threat based on TTC and distance.

8. Integration and Feedback:

- Integrate the processed information into the vehicle's autonomous driving or driver-assistance system.

- Continuously update and refine detection, tracking, and prediction algorithms based on real-world performance and feedback.

This ensures that the system operates in real-time, providing timely and accurate detection, distance estimation, and cut-in prediction to enhance the safety and responsiveness of autonomous and driver-assistance systems.



Architecture Diagram

Technologies used

1: YOLO (You Only Look Once)
2: OpenCV
3: Convolutional Neural Network (CNN)
4: Python
5: Matplotlib
6: Numpy
7: Pandas

Team members and contribution:

1: Sampreeth (Member 1):

- Develop algorithms to detect potential cut-ins using bounding box coordinates and IoU calculations.
- Implement logic to compute the time to collision (TTC) for detected vehicles.
- Integrate the cut-in detection module with the main detection pipeline.
- Implement YOLO model for vehicle detection.

2: Praneeth (Member 2):

- Ensure the system detects and flags only relevant cut-ins that pose a danger.
- Fine-tune and train the model using a relevant dataset.
- Optimize the model for real-time performance.
- Integrate the model with OpenCV for bounding box extraction and distance calculation.

3: Harshak(Member 3):

- Integrate all components (detection, cut-in detection, and visualization) into a cohesive system.
- Develop and execute unit tests to ensure each module functions correctly.
- Conduct comprehensive system validation tests under various conditions to ensure reliability.

4: Shrish(Member 4):

- Develop scripts to read, process, and annotate image frames.
- Implement visualization tools using Matplotlib to display detection results, bounding boxes, distances, and TTC.
- Assist in validating the system against real-world scenarios.

Conclusion

This project successfully integrates advanced computer vision techniques with real-time vehicle detection and safety prediction mechanisms to address potential road safety issues. By leveraging the YOLO model for efficient vehicle detection and incorporating algorithms for cut-in detection and time-to-collision (TTC) calculations, the system provides a robust solution for identifying and predicting hazardous driving scenarios. The collaborative effort ensured a comprehensive approach to data processing, model optimization, and system integration, resulting in a functional and reliable prototype. This work demonstrates the potential of AI-driven solutions in enhancing vehicular safety and offers a foundation for further development and deployment in real-world applications.