23CSE209:PYTHON PROGRAMMING

VISION-BASED LANE DETECTION AND OBSTACLE-AWARE DECISION MAKING IN AUTONOMOUS DRIVING

FINAL PRESENTATION

MAMMILAPALLII SHNAMUKHA VINAYAK(ARE24023)

AGRAHARAM RAKESH(ARE24001)

PARDHU SURYA ABHIRAM N(ARE24031)

SIVA SAI INDEEVER P(ARE24061)

INTRODUCTION

- Vehicles today rely on Computer Vision to identify lanes and obstacles.
- Lane detection helps maintain safe driving paths.
- Object detection identifies cars, pedestrians, and cyclists to prevent collisions Implemented using OpenCV (for lane) and YOLOv8 (for object detection).

Problem Definition

- Safe navigation is critical in autonomous driving.
- Existing depth camera (LiDAR, depth cameras) provide accuracy but are costly and less viable for small-scale systems.
- computer vision offers a cost-effective way to estimate depth and detect obstacles.

OBJECTIVES

- Detect lanes which helps in lane keeping
- deets vehicles and pedastrians detect video feeds.
- Implement a real-time decision engine to guide safe driving actions.
- voice based assistance

LITERATURE REVIEW

LANE DETECTION:

- POLYNOMIAL FITTING AND CNN-BASED METHODS (SCNN, LANENET) IMPROVED ACCURACY.
- HOWEVER, DEEP MODELS NEED LARGE DATASETS AND GPUS, LIMITING CLASSROOM DEMOS.

OBJECT DETECTION:

- TRADITIONAL: HOG + SVM (SLOW, LESS ROBUST).
- MODERN: YOLO, SSD, FASTER R-CNN ACHIEVE REAL-TIME DETECTION.
- YOLOV8 OFFERS BEST BALANCE OF SPEED, ACCURACY, AND SIMPLICITY.

OUR APPROACH:

- A LIGHTWEIGHT FUSION OF OPENCV-BASED LANE DETECTION AND YOLOV8 FOR OBJECTS — IDEAL FOR REAL-TIME ACADEMIC ADAS DEMONSTRATION.

METHODOLOGY (PYTHON & TOOLS)

INPUT:
FRONT-REAR CAMERA ON

PROCESSING FLOW:

- LANE & ROAD BOUNDARY DETECTION.
- OBJECT DETECTION & CLASSIFICATION (STATIC/DYNAMIC)
- DEPTH ESTIMATION USING COMPUTER VISION.
- RULE-BASED DECISION LOGIC → "SAFE LANE" / BRAKING

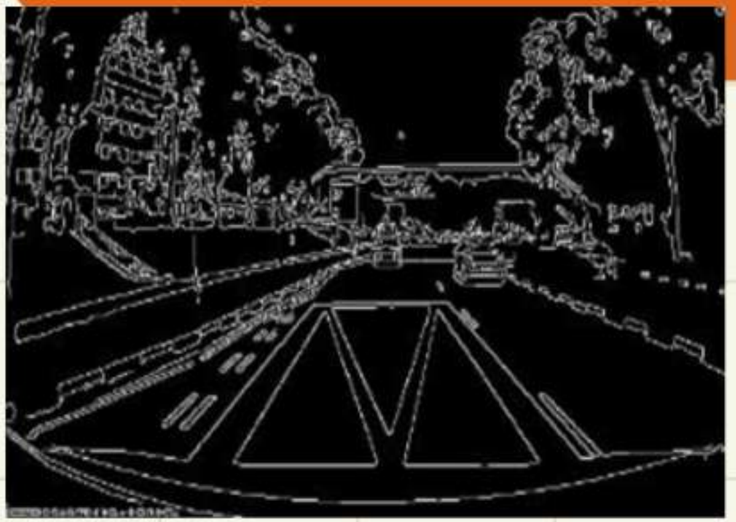
PYTHON PACKAGES:

OUTPUT:
DETECTED OBSTACLES AND LSNES

PACKAGES

- MoviePy: Video processing and editing.
- Ultralytics YOLO: Object detection model (YOLOv8).
- Numpy: Matrix and array operations.
- OpenCV: Image processing for lane and object detection.
- Matplotlib: Visualization (plots and dashboards).
- Pandas: Logging and statistical analysis.
- Scipy: Signal smoothing for confidence metrics.

ALGORITHMS/MODELS



Theory Behind Code:

1. Convert to grayscale and blur to remove noise.
2. Use Canny edge detection to highlight edges.
3. Apply Region of Interest (ROI) mask to focus on road area.
4. Use Hough Transform to detect straight lines.
5. Separate left & right lanes by slope and average them.
6. Draw lanes and compute confidence score

ALGORITHMS/MODELS

Theory Behind Code:

1. Run YOLOv8 model inference → returns bounding boxes.
2. Extract class, confidence, and coordinates.
3. Filter relevant classes (vehicles/pedestrians).
4. Classify threat levels (HIGH/MED/LOW) based on vertical position (y2).
5. Draw bounding boxes in color-coded levels.

RESULTS OVERVIEW

Level 1 – Lane Detection:

- Given a road image, the system detects lane markings using Canny + Hough Transform and overlays detected lane lines.

Level 2 – Object Detection:

- Given the same or a different image, YOLOv8 identifies vehicles, pedestrians, and other road objects with bounding boxes.

Level 3 – Integrated ADAS View:

- Combines both — lanes and detected objects — into a single annotated image with overlay commands for driver assistance.

LEVEL1

LEVEL2

Original

Objects: 3

LEVEL 3

Original

Complete ADAS

VISUAL OUTPUT

DESCRIPTION:

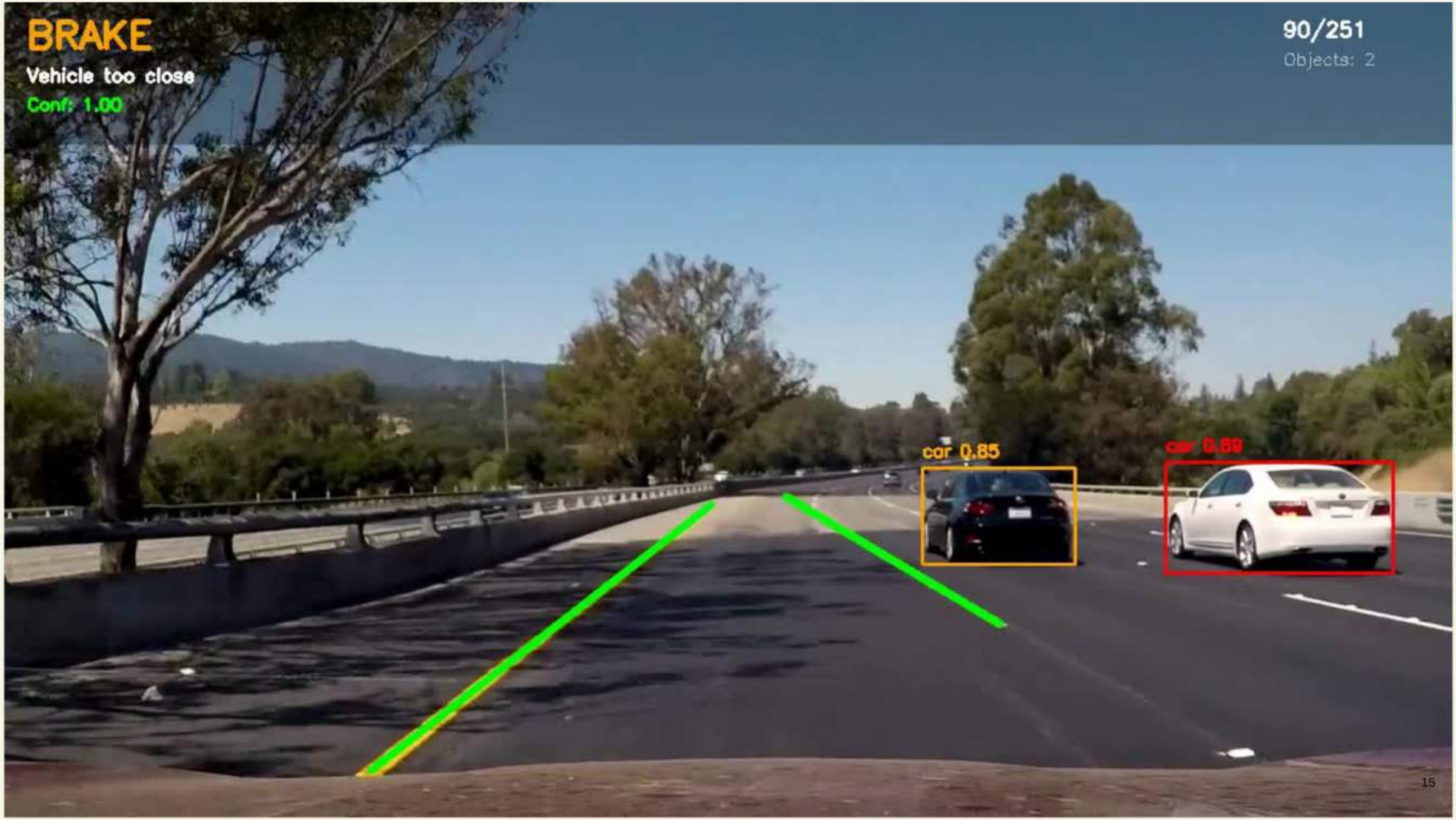
- EXTENDS THE IMAGE-BASED ADAS SYSTEM TO REAL-TIME VIDEO PROCESSING.
- EACH VIDEO FRAME UNDERGOES LANE DETECTION AND OBJECT RECOGNITION SIMULTANEOUSLY.

FEATURES:

- PROCESSES CONTINUOUS FRAMES WITH LANE + OBJECT OVERLAYS.
- REAL-TIME ALERT SYSTEM USING VOICE FEEDBACK FOR CRITICAL EVENTS (E.G., OBSTACLE WARNING).
- DEMONSTRATES END-TO-END ADAS FUNCTIONALITY.

OUTPUT:

- ANNOTATED VIDEO SHOWING DYNAMIC LANE GUIDANCE AND OBJECT TRACKING.


BRAKE
Vehicle too close
Conf: 1.00
90/251
Objects: 2
car 0.85
car 0.89

SIMPLE DASHBORD

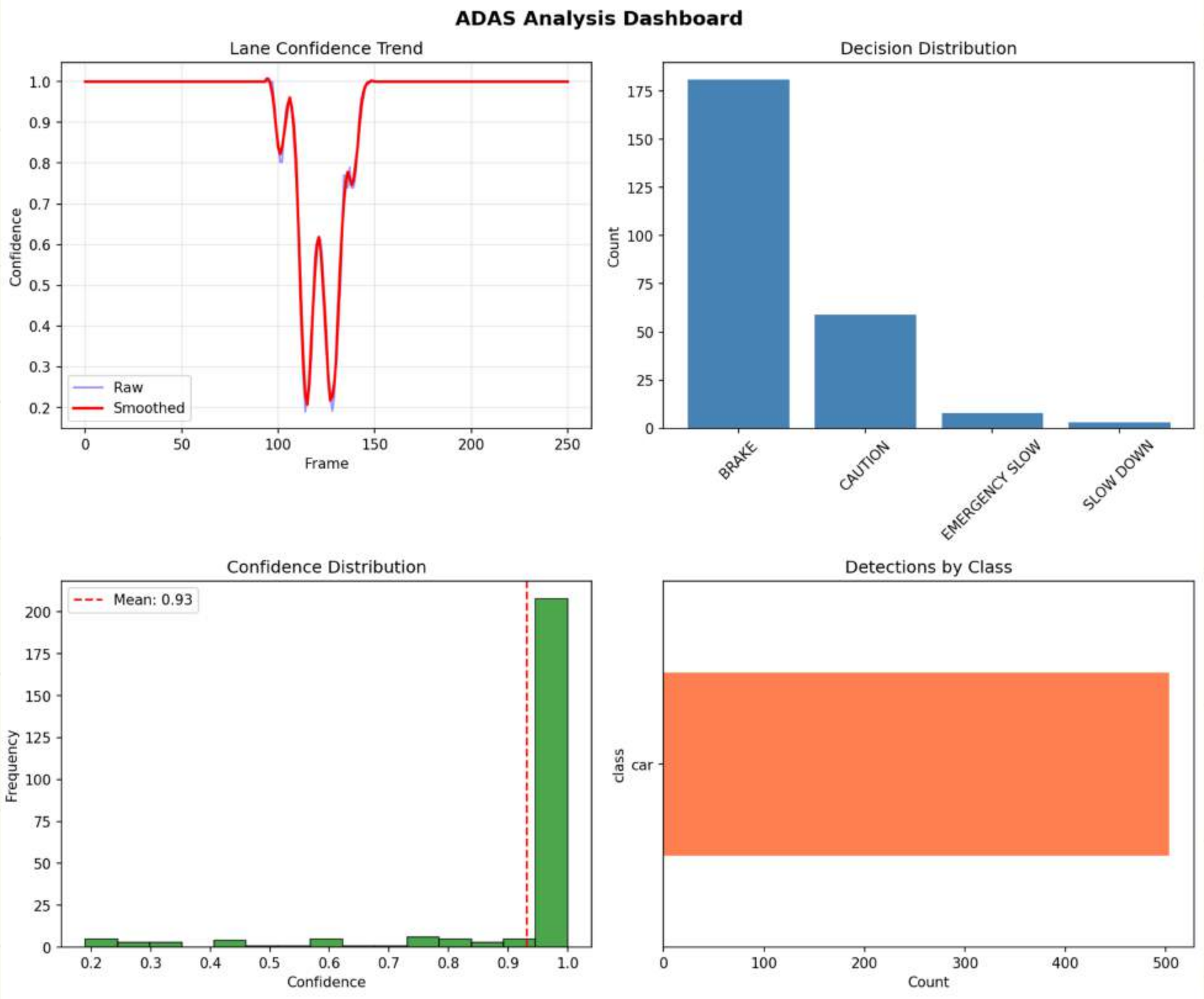
- PURPOSE:
- TO VISUALIZE ALL THE PERFORMANCE DATA IN ONE PLACE USING MATPLOTLIB.
- IT GENERATES A DASHBOARD WITH 4 CHARTS THAT GIVE A FULL SUMMARY OF SYSTEM PERFORMANCE.

1) LANE CONFIDENCE TREND (TOP-LEFT GRAPH)
   - PLOTS CONFIDENCE VS. FRAME NUMBER.
   - HELPS VISUALIZE HOW STABLE THE LANE DETECTION WAS DURING THE VIDEO.

(2) DECISION DISTRIBUTION (TOP-RIGHT GRAPH)
   - COUNTS HOW MANY TIMES EACH DECISION WAS MADE (E.G., “STOP”, “BRAKE”, “SLOW DOWN”).
   - PLOTTED AS A BAR CHART.

(3) CONFIDENCE HISTOGRAM (BOTTOM-LEFT GRAPH)
   - DISPLAYS HOW THE CONFIDENCE VALUES ARE SPREAD OVERALL.
   - IT USES A HISTOGRAM (FREQUENCY GRAPH).
   - DRAWS A RED DASHED LINE TO SHOW THE MEAN CONFIDENCE VALUE.

(4) DETECTIONS BY CLASS (BOTTOM-RIGHT GRAPH)
   - USES THE DETECTION DATAFRAME AGAIN.
   - COUNTS HOW MANY DETECTIONS PER CLASS (CAR, PERSON, BUS, ETC.).
   - PLOTS A HORIZONTAL BAR CHART USING COLOR “CORAL”.


ADAS Analysis Dashboard
Lane Confidence Trend
Confidence
Frame
Raw
Smoothed
Decision Distribution
Count
BRAKE
CAUTION
EMERGENCY SLOW
SLOW DOWN
Confidence Distribution
Mean: 0.93
Frequency
Confidence
Detections by Class
class
car
Count

MAJOR CONCLUSIONS

- ADAS system can be implemented in Python using open-source tools and pre-trained models for robust road scene analysis.
- Hybrid lane and object detection, coupled with simple rule-based decision making, achieves high accuracy and interpretability for academic demonstration.
- The framework is modular, adaptable for extending to more advanced models, and suitable for integration into robotics or vehicle safety research.

Performance

- The system runs smoothly and can be used for research or testing projects.
- It makes safe decisions — stopping immediately for pedestrians and showing caution for vehicles.
- It also creates data logs and dashboard charts to review the system's performance.

REFERENCES

- Ultralytics YOLO documentation: https://docs.ultralytics.com/
- OpenCV library docs: https://opencv.org/
- MoviePy library docs: https://zulko.github.io/moviepy/
- Scipy documentation: https://scipy.org/
- Matplotlib library docs: https://matplotlib.org/
- Relevant academic works on ADAS, lane detection, and YOLO (include any paper citations as needed for your version).


CONTINUE
Path clear
Conf: 1.00
78/1199
Objects: 0

THANK YOU