Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 02, February-2025

Road Lane and Object Detection in the Video Footage using Open CV and Python

Mukul R Kulkarni
Department of Mechanical Engineering
RV College of Engineering
Bengaluru, India

Karthik M A
Department of Mechanical Engineering
RV College of Engineering
Bengaluru, India

Mehul Abhishua
Department of Mechanical Engineering
RV College of Engineering
Bengaluru, India

Dr. Nagesh S
Department of Mechanical Engineering
RV College of Engineering
Bengaluru, India

***Abstract*— This paper presents an innovative framework for road lane and object detection in video footage using OpenCV and Python. The lane detection component employs preprocessing techniques such as edge detection and perspective transformation to accurately identify lanes. Object detection is achieved through advanced methods, enabling the identification of vehicles, pedestrians, and obstacles. The integration of these functionalities provides a robust system for autonomous driving and driver assistance applications. Experimental results demonstrate the system's effectiveness under varying road and lighting conditions, showcasing its potential to enhance road safety and vehicular technology.**

***Keywords*— Road Lane Detection; Object detection; OpenCV; YOLO;**

I. INTRODUCTION

The rapid advancement of autonomous vehicles and Advanced Driver-Assistance Systems (ADAS) has created a growing demand for accurate road lane and object detection technologies. Ensuring vehicles remain within their designated lanes and detecting obstacles such as other vehicles and pedestrians are critical functionalities for enhancing road safety and developing reliable autonomous navigation systems.

Improper lane discipline is a significant factor in road accidents, resulting in substantial fatalities and injuries each year. According to the Transport Research Wing under the *Ministry of Road Transport & Highways*, in 2019, over 37,259 out of 4,61,312 road accident deaths in India were attributed to overtaking and lane-changing incidents. This highlights the urgent need for effective measures to improve lane discipline and prevent accidents.

Lane detection ensures that vehicles stay within their designated lanes, while object detection identifies and classifies obstacles. This dual functionality is critical for creating safe and reliable systems that can adapt to diverse environments.

This paper presents a comprehensive approach to road lane and object detection using OpenCV, Python, and YOLOv8. The proposed system employs computer vision techniques to analyze recorded video footage, accurately identifying road lanes and various objects. YOLOv8, the latest version of the YOLO series, is utilized for object detection, enabling the identification of vehicles, pedestrians, and obstacles with high accuracy and efficiency. By integrating these functionalities, the system provides a robust solution for enhancing road safety and vehicular technology.

The importance of lane detection and object detection systems lies in their ability to adapt to diverse environments and ensure safe navigation. As mechanical engineers, understanding and developing these systems are crucial for addressing real-world challenges and improving vehicle safety standards. This project demonstrates the critical role of engineering in enhancing road safety measures and reducing the incidence of road accidents caused by improper lane discipline.

II. LITERATURE REVIEW

1) Hough Transform for Lane Detection Zhang, Y., et al. (2016). "Real-time lane detection using Hough Transform."[1] Journal of Transportation Engineering. This paper discusses the application of the Hough Transform for real-time lane detection in video streams. The authors highlight the efficiency of the algorithm in detecting straight lane markings, even under varying environmental conditions.

2) H. Shafique, et al. (2018). "Lane detection using HSV color space."[2] International Journal of Computer Applications. This study investigates the use of the HSV color space for lane detection. The authors demonstrate that transforming the image into HSV allows for better separation of lane markings from the road surface, improving detection accuracy across different lighting conditions.

3) Chen, L., et al. (2019). "Lane detection using convolutional neural networks."[3] IEEE Transactions on Intelligent Transportation Systems. This paper presents a CNN-based approach for lane detection. The authors show that their model outperforms traditional methods, especially in complex driving scenarios, by capturing lane features more effectively.

4) Tiwari, S. G., & Gupta, S. (2020). "Object detection in urban environments using OpenCV."

[4] Journal of Computer Vision and Image Processing. This research explores various background subtraction techniques implemented in OpenCV for detecting moving objects in urban settings. The authors provide a comparative analysis of different algorithms, highlighting their effectiveness in real-time applications.

5) Redmon, J., et al. (2016). "You Only Look Once: Unified Real-Time Object Detection.". [5] This foundational paper introduces the YOLO framework, which revolutionizes object detection by treating it as a single regression problem. The authors demonstrate the model's ability to detect multiple objects in real-time, making it highly suitable for applications in autonomous driving.

6) K. M. Althoff, et al. (2021). "Multi-task learning for lane and object detection." [6] IEEE Transactions on Intelligent Vehicles. This study explores the integration of lane and object detection using a multi-task learning framework. The authors demonstrate that simultaneous detection improves overall performance and situational awareness for autonomous vehicles.

7) "An end-to-end lane and vehicle detection system." [7] IEEE Transactions on Intelligent Transportation Systems. This paper presents an end-to-end framework that combines lane and vehicle detection. The authors emphasize the importance of integrating these tasks for enhanced decision-making in autonomous driving scenarios.

III. METHODOLOGY

The methodology part includes two key factors which is Lane detection and object recognition are the two main goals of the technique, which aims to improve situational awareness and driving safety. In order to identify lane markings, the image is first processed using methods like perspective transformation and edge detection. Vehicles, people, and other barriers in the scene are simultaneously identified using an object detection model (YOLOv8). A thorough picture of the road environment is then produced by integrating the outcomes of the two tasks and superimposing lane markers and identified objects onto the original frame. This cohesive strategy provides notable enhancements in comprehending and maneuvering through intricate road situations.

Image Preprocessing:

- Conversion to HLS Color Space: This step involves converting the input image to the HLS color space to enhance the visibility of lane markings. The HLS color space separates the image into hue, lightness, and saturation components, making it easier to distinguish lane lines under different lighting conditions.
- Gaussian Blur: A Gaussian blur is applied to the image to reduce noise and smooth out any irregularities. This step is crucial for minimizing false edges that might interfere with accurate lane detection.
- Edge Detection: The Canny edge detection algorithm is used to identify the edges within the image. This algorithm highlights the boundaries of lane markings, making them more prominent for further processing.

Region of Interest (ROI) Extraction:

- Masking: To focus on the road and ignore irrelevant areas, a mask is applied to the image. This mask isolates the region of interest (ROI) where lane markings are most likely to be found, such as the lower portion of the frame where the road is visible.

Perspective Transformation:

- Warping: The image is transformed into a bird's-eye view perspective, which provides a top-down view of the road. This transformation simplifies lane detection by making the lane lines appear parallel and straight.
- Points Definition: Source and destination points are defined for the perspective transformation. The source points correspond to the corners of the trapezoidal region where the lane lines converge, while the destination points form a rectangle representing the bird's-eye view.

Lane Detection and Tracking:

- Histogram-Based Localization: A histogram of the lower half of the bird's-eye view image is used to locate the lane lines. Peaks in the histogram indicate the positions of the lane markings.
- Sliding Window Technique: Starting from the identified positions, the sliding window technique is used to trace the lane lines. This involves placing a series of windows along the lane lines and identifying the non-zero pixels within each window.
- Polynomial Fitting: A polynomial curve is fitted to the identified lane points, creating a smooth representation of the lane lines. This polynomial fitting helps in accurately modelling the curves of the lane.

A. Object Detection System

Installation and Setup:

- Install the necessary Ultralytics package and set up the YOLOv8 model for use.

Model Loading:

- Load the YOLOv8 model with pre-trained weights. This model is capable of detecting a wide range of objects with high accuracy, making it suitable for identifying vehicles, pedestrians, and other obstacles in the video footage.

Frame Processing:

- Pass each frame of the video footage through the YOLOv8 model. The model divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell.

- Classify detected objects into categories such as vehicles, pedestrians, and obstacles. Each object is annotated with a bounding box that highlights its position within the frame.

Detection Details:

- Each detected object is enclosed within a bounding box, which is a rectangular frame surrounding the object.
- Each bounding box is associated with a class label indicating the type of object detected (e.g., person, car, potted plant).
- The model provides a confidence score for each detected object, indicating the likelihood that the detection is accurate.

Integration with Lane Detection:

- Overlay the detected lanes and objects on the original video frame. This comprehensive visualization allows for a better understanding of the road environment and enhances the system's functionality.

Video Processing:

- Process each frame of the video and display the detected objects with their bounding boxes and class labels.

Saving the Processed Video:

- Save the final processed video for further analysis and review.

B. System Integration

The system successfully integrated the outputs of lane detection and object detection, creating a synchronized and unified visualization of lane boundaries and object annotations.

Synchronization:

- The lane detection and object detection processes were meticulously synchronized to operate on the same video frames. This step was vital to ensure that the detected lane boundaries and objects corresponded to the exact moment in the video, allowing for coherent and consistent analysis. The synchronization ensured that both detections were temporally aligned, providing an accurate representation of the road environment at any given time.

Overlaying Results:

- The combined results from both lane detection and object detection were then seamlessly overlaid onto the original video frame. Lane boundaries were highlighted with clearly defined curves, making them easily distinguishable. Concurrently, objects detected by the YOLOv8 model were annotated with bounding boxes and class labels. Each bounding box was labelled with the object's type (e.g., person, car, potted plant) and accompanied by a confidence score, indicating the likelihood of accurate detection. This overlay process provided a comprehensive visual representation of the road scenario, integrating both lane and object information into a single frame.

Display:

- The final output was displayed, presenting a thorough and detailed view of the road environment. This visualization was instrumental in aiding drivers or autonomous systems to navigate safely by providing a complete understanding of their surroundings. The simultaneous display of lane boundaries and object annotations ensured that all crucial information was readily available in one view, enhancing situational awareness. This integrated approach not only improved safety but also offered a novel solution to the challenge of real- time road analysis, making it a significant contribution to the field of autonomous driving and advanced driver-assistance systems (ADAS).

IV. FLOWCHART

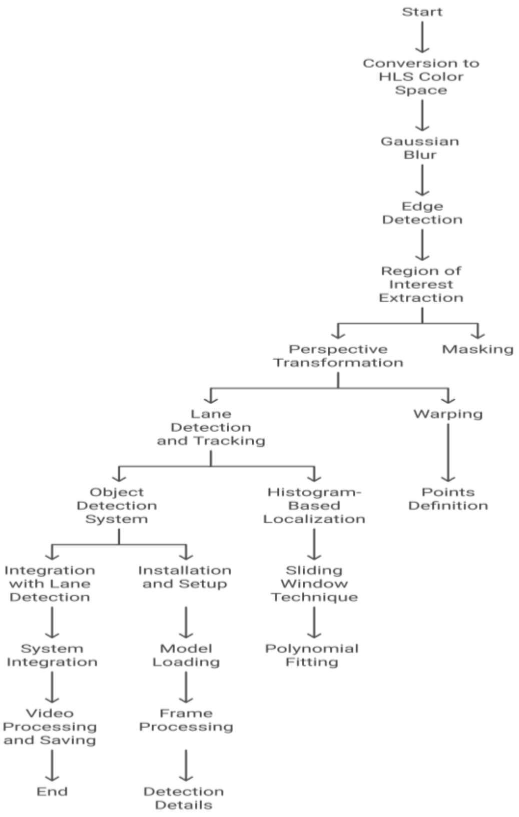


Fig. 1. Flowchart of Lane and Object Detection System

The flowchart highlights the essential elements of a road lane and object detection system and shows its layers. There are three main stages to the system's structure. Lane Boundary

Detection, the initial stage, is essential to autonomous driving and lane departure warning systems since it recognizes road markers and borders to guarantee that cars stay in their lanes. In order to improve road safety through real-time hazard detection, the second stage, object detection and classification, focuses on identifying and classifying items including cars, pedestrians, and obstructions. By confirming lane detection and object categorization and improving algorithms in response to test findings, the last step, System Validation and Testing, guarantees the correctness and dependability of the detection models. The diagram's pyramidal structure represents a methodical process in which each layer develops upon the previous one creating a robust and efficient road detection system.

V. IMPLEMENTATION

The implementation process involves processing video frames to identify lanes and objects, which is part of the implementation process. The frames are taken first, and then it is improved by applying a blur and converting them to grayscale. Lane markers are detected using edge detection, and for increased accuracy, a particular area is targeted. The Hough Line Transform is used to detect lane lines, while pre- trained algorithms such as YOLO are used to identify objects. For safer travel, this combination helps to recognize the path and objects. Video Capture: The first step in the process involves capturing video frames from a source, such as a pre-recorded video file or a live camera feed. This provides the raw input data for further processing.

Preprocessing: To enhance the quality of the input frames, the captured video frames are converted to grayscale. This reduces computational complexity by eliminating color information, focusing solely on intensity. Additionally, a Gaussian blur is applied to the grayscale frames to reduce noise and smoothen the image, which helps in edge detection.

Canny Edge Detection: The processed frames undergo edge detection using the Canny algorithm. This algorithm identifies areas of high intensity change, which typically correspond to edges of objects within the frame. Edge detection is a crucial step in identifying lane markings and other structural elements in the video.

Region of Interest (ROI): A region of interest is defined to focus the analysis on a specific part of the frame where lane markings are expected to be found, typically the lower half of the frame. By applying a mask to the frame, we restrict the subsequent processing steps to this region, improving efficiency and accuracy.

Hough Line Transform: The Hough Line Transform is used to detect straight lines within the defined region of interest. This method is particularly effective for identifying lane markings, which are often represented as continuous or dashed lines in the frame. The detected lines provide the structural information necessary for lane detection.

Object Detection: To identify objects within the video frames, a pre-trained deep learning model, such as YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), or Faster R-CNN, is employed. These models are trained on large datasets and can accurately detect and classify various objects in real-time. The model processes each frame, producing bounding boxes and class labels for detected objects.

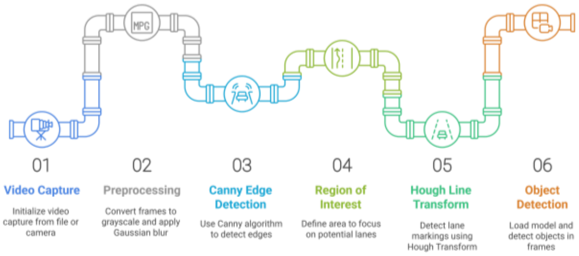

MPG
01
Video Capture
Initialize video capture from file or camera
02
Preprocessing
Convert frames to grayscale and apply Gaussian blur
03
Canny Edge Detection
Use Canny algorithm to detect edges
04
Region of Interest
Define area to focus on potential lanes
05
Hough Line Transform
Detect lane markings using Hough Transform
06
Object Detection
Load model and detect objects in frames

Fig. 2. Steps for the Implementation to identify the Road Lanes

YOLO Object Detection: The figure below offers a methodical summary of the YOLO (You Only Look Once) object identification procedure, dissecting it into crucial elements required for its execution. The process is shown in detail in the figure on the left, which begins with configuring the environment with required libraries like OpenCV and NumPy, choosing the right YOLO model (such as YOLOv3, YOLOv4, and YOLOv5), and using pre-trained models based on datasets like COCO. It goes on to describe important steps including recording video from cameras or files, converting images into blobs, using model inference to get bounding box coordinates, class IDs, and confidence scores, and using non- max suppression to hone detections. Real-time bounding box sketching is another step in the process, and a continuous live feed is used to show the finished product. The elements which are shown in the right diagram are the interconnected keys, highlighting their function in opening a productive YOLO-based object detection system. When taken as a whole, these examples demonstrate the methodical methodology needed to apply YOLO for real-time object identification with the best possible accuracy and speed.

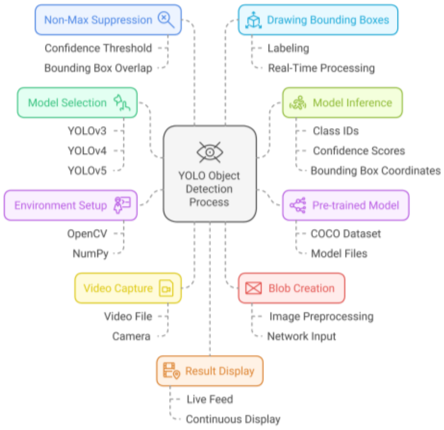


Fig. 3. YOLO Object Detection Process

Object Detection Process: Below figure shows the steps of a road lane and object detection system are depicted in this image. The approach starts by converting a road scene's raw original color image to grayscale for easier processing. The next step is edge detection, which involves first identifying possible boundaries using the Canny edge detection technique and then converting this to a binary edge picture for more distinct delineation. These phases help to identify the object and road lane.

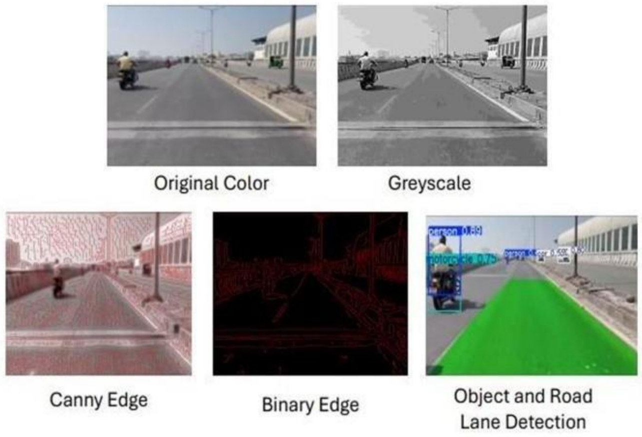


Fig. 4. A Step by Step Breakdown from Raw Image to Object Detection

Image Processing for Object Detection:

Below picture depicts the complete pipeline for the object detection on the road lane.

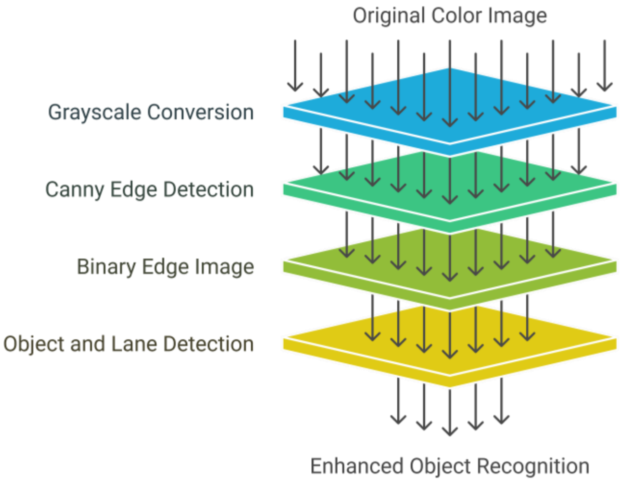


Fig. 5. Image Processing Pipeline for Object Detection Mobile

Camera Route Visualization:

Locations: Collection from diverse locations including maps and various urban and suburban areas. - Extensive footage captured in Bengaluru, highlighting busy streets, intersections, and residential areas. - Inclusion of varied settings such as highways, city centers, and local roads to ensure comprehensive data coverage.

Equipment: Utilized mobile cameras for capturing footage. Mobile cameras chosen for their ease of use and flexibility in different environments. - ensures high-resolution settings for clarity and detail in the collected footage (1080p; 30 fps). Below picture depicts the Mobile Camera Route Mapping. The Key Components of an Effective Footage Collection Strategy is shown below in figure 6

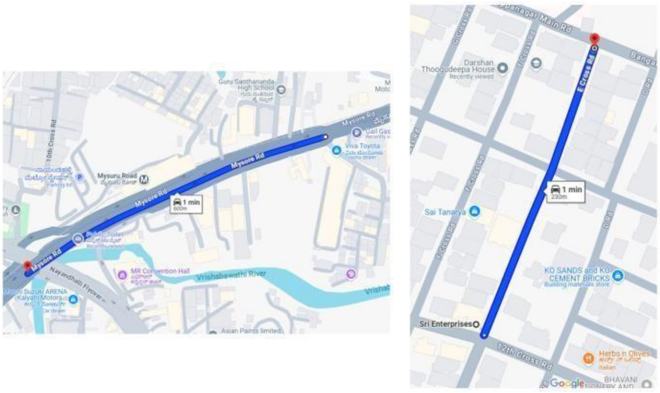


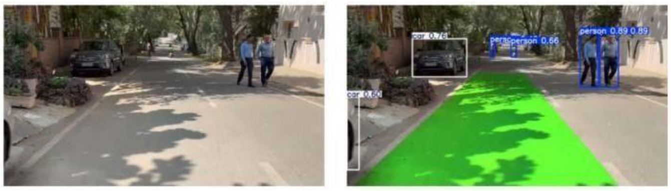
Fig. 6. Video capturing in Bengaluru Locations using Mobile Cameras for Clear High Resolution Footage (1080p, 30 fps)

Breakdown of Footage Collection Strategy
Equipment
Mobile cameras used for flexibility and clarity
Locations
Diverse urban and suburban areas covered

Fig. 7. Mobile Camera Data Collection

Visual Scene Analysis of both Road Lane(Green) and Object Detection Respectively:

This picture below demonstrates the process by which a computer vision system recognizes lanes and objects in a road scene. The original camera image is displayed on the left. Once the image has been analysed by the system, it appears on the right side. The car and passengers are surrounded by boxes that the system has drawn, labelled, and assigned a confidence score to. Additionally, it shows the green driving lane. This illustrates the system’s comprehension of what it “sees” in the picture, which is essential for autonomous vehicles and other related technologies.



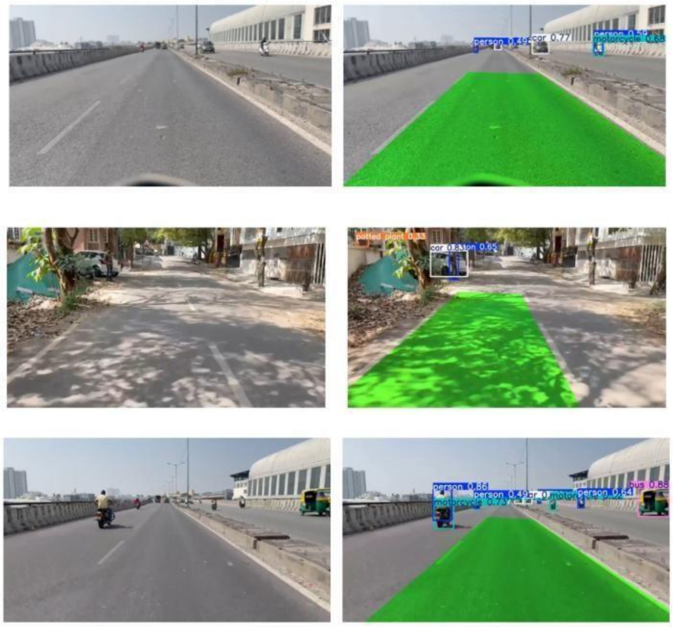


Fig. 8. Road Lane and Object Detection using OpenCV and Python

Figures and Tables

From taking high-resolution video in Bengaluru using smartphone cameras to processing the photos for improved object recognition, this set of pictures illustrates the road lane and object detection process. The procedures involve object detection, picture conversion, and the use of Python and OpenCV for the end result. The flowchart shows the road lane detection process, emphasizing the main goals and techniques needed to get precise detection results.

Sl. no	Title of the Image	Description
1	Flowchart of Lane and Object Detection System	The flowchart illustrates the sequential steps involved in creating and testing a system, beginning with implementation and progressing through object detection and classification, lane border detection, system validation, and testing.
2	Steps for the Implementation to identify the Road Lanes	A video processing pipeline's processes are shown in this figure, which begins with video capture and moves on to Preprocessing, edge identification, region of interest selection, line transformation, and object detection.
3	YOLO Object Detection Process	The phases from video input and Preprocessing to model selection, inference, and finally the presented findings are all depicted in this figure, which demonstrates the YOLO object identification process.
4	A Step by Step Breakdown from Raw Image to Object Detection	From the original color image to grayscale, edge detection, and eventually object and lane marker identification, this figure illustrates the steps involved in image processing for object and lane recognition.
5	Image Processing Pipeline for Object Detection	The image processing processes for object detection are depicted in this graphic, starting with the raw input image and continuing through grayscale conversion, edge detection, and object and lane recognition.
6	Image capturing of Bengaluru Locations using Mobile Cameras for Clear High Resolution Footage (1080p,30 fps)	Route maps from mobile camera data.
7	Mobile Camera Data Collection	Data collection strategy using mobile cameras in diverse locations.
8	Road Lane and Object Detection using OpenCV and Python	Result of road lane and object detection on left and right side respectively.

Table. 1. Understanding of Each Figure in detail

VI. RESULTS AND FUTURE WORK

The implementation process for the "Road Lane and Object Detection in the Video Footage Using OpenCV and Python" project involves several crucial steps, including video capture, preprocessing, edge detection, region of interest selection, Hough Line Transform for lane detection, and the use of pre-trained models like YOLO for object detection. This comprehensive pipeline ensures accurate identification of lanes and objects in video footage, contributing to safer travel. Currently, the model is applied to recorded footage, and the time required to process each frame is relatively longer. This limitation highlights the need for real-time processing capabilities to enhance the system's practicality and usability. Future work will focus on optimizing the model to achieve real-time performance, enabling immediate detection and response. Moreover, integrating warning and driving assistance features will be a key objective. By introducing real-time alerts for detected objects and potential hazards, the system can significantly improve road safety. These enhancements will ensure that the technology not only identifies lanes and objects accurately but also provides actionable assistance to drivers, ultimately contributing to a safer driving experience.

In summary, while the current model demonstrates promising results in lane and object detection from recorded footage, future advancements aim to achieve real-time processing and introduce driving assistance features to further enhance road safety. This continued development will pave the way for more practical and effective applications of the technology in real-world scenarios.

ACKNOWLEDGMENT

We, a team of three mechanical engineering students, express our sincere gratitude to RV College of Engineering for providing us with the opportunity to undertake the minor project titled "Road Lane and Object Detection in the Video Footage Using OpenCV and Python." We are deeply grateful to our guide, Dr. Nagesh S, whose invaluable guidance and support were instrumental in the timely and successful completion of this project. We also extend our heartfelt gratitude to our Head of Department, Dr. Shanmukha Nagaraj, for his academic leadership and unwavering encouragement, which greatly facilitated our research and project work.

REFERENCES

[1] Y. Zhang, et al., "Real-time lane detection using Hough Transform," Journal of Transportation Engineering, 2016.
[2] H. Shafique, et al., "Lane detection using HSV color space," International Journal of Computer Applications, 2018.
[3] L. Chen, et al., "Lane detection using convolutional neural networks," IEEE Transactions on Intelligent Transportation Systems, 2019.
[4] S. G. Tiwari and S. Gupta, "Object detection in urban environments using OpenCV," Journal of Computer Vision and Image Processing, 2020.
[5] J. Redmon, et al., “You Only Look Once: Unified Real- Time Object Detection” IEEE Transactions on Pattern Analysis and Machine Learning, 2019.
[6] K. M. Althoff, et al., "Multi-task learning for lane and object detection," IEEE Transactions on Intelligent Vehicles, 2021.
[7] An end-to-end lane and vehicle detection system." IEEE Transactions on Intelligent Transportation System.