A PROJECT SYNOPSIS

ON

"LANE DETECTION SYSTEM"

Submitted in the partial fulfillment of the requirements for

The degree of

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

1)ATHARV BHILARE
2) NAUSHAD QUADRI
3) SHAIKH AMIR HAMZA
4) ADITYA UTEKAR

UNDER THE GUIDANCE OF

Prof. Hemalata Gosavi

Department of Computer Engineering
Saraswati College of Engineering, Kharghar, Navi Mumbai
University of Mumbai
2023-24

Saraswati College of Engineering

Vision

To be universally accepted as autonomous center of learning in engineering education and research.

Mission

- To educate Students to become responsible & quality technocrats to fulfill society and industry needs.
- To nurture student's creativity and skills for taking up challenges in all facets of life.

Department of Computer Engineering

Vision

To be among renowned institution in computer engineering education and research by developing globally competent graduates.

Mission

- To produce eminent Engineering graduates by imparting quality teaching learning and providing hands on experience in programming to fulfill society and industry needs.
- To encourage students to pursue research and learn new technologies in Computer Engineering and across interdisciplinary areas so that they participate in research, innovation and entrepreneurship.
- To provide stimulating learning ambience to enhance innovative ideas, problem solving ability, leadership qualities, team-spirit and ethical responsibilities.

SARASWATI Education Society's
SARASWATI College of Engineering
Learn Live Achieve and Contribute
Kharghar, Navi Mumbai - 410 210.

(Approved by AICTE, recg. By Maharashtra Govt. DTE ,Affiliated to Mumbai University)

PLOT NO. 46/46A, SECTOR NO 5, BEHIND MSEB SUBSTATION, KHARGHAR, NAVI MUMBAI-410210
Tel. : 022-27743706 to 11 * Fax : 022-27743712 * Website: www.sce.edu.in

CERTIFICATE

This is to certify that the requirements for the synopsis report entitled "Lane Detection System" have been successfully completed by the following students:

Roll numbers	Name
10	Atharv Bhilare
61	Naushad Quadri
65	Shaikh Amir Hamza
75	Aditya Utekar

In partial fulfillment of Sem –VII , **Bachelor of Engineering of Mumbai University in Computer Engineering** of Saraswati college of Engineering , Kharghar during the academic year 2023-24.

Internal Guide
Prof. Hemalata Gosavi

External Examiner

Project Co-ordinator
Dr. Anjali Dadhich

Head of Department
Prof. Sujata Bhairnallykar

Principal
Dr. Manjusha Deshmukh

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVE's

1) To embed a strong foundation of Computer Engineering fundamentals to identify, solve, analyze & design real time Engineering problem as a professional or an entrepreneur for the benefit of society.
2) To motivate & prepare students for lifelong learning & research to manifest global competitiveness.
3) To equip students with communication, team work & leadership skills to accept challenges in all facets of life ethically.

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAM OUTCOMES

1. Apply the knowledge of Mathematics, Science and Engineering Fundamentals to solve complex Computer Engineering Problems.
2. Identify, formulate and analyze Computer Engineering Problems and derive conclusion using First Principle of Mathematics, Engineering Science and Computer Science.
3. Investigate Complex Computer engineering problems to find appropriate solution leading to valid conclusion.
4. Design a software System, components, Process to meet specified needs with appropriate attention to health and Safety Standards, Environmental and Societal Considerations.
5. Create, select and apply appropriate techniques, resources and advance engineering software to analyze tools and design for Computer Engineering Problems.
6. Understand the Impact of Computer Engineering solution on society and environment for Sustainable development.
7. Understand Societal, health, Safety, cultural, Legal issues and Responsibilities relevant to Engineering Profession.
8. Apply Professional ethics, accountability and equity in Engineering Profession.
9. Work effectively as a member and leader in multidisciplinary team for a common goal.
10. Communicate effectively within a Profession and Society at large.
11. Appropriately incorporate principles of Management and Finance in one's own Work.
12. Identify educational needs and engage in lifelong learning in a Changing World of Technology.

DEPARTMENT OF COMPUTER ENGINEERING

PROGRAMME SPECIFIC OUTCOME

1) Formulate and analyze complex engineering problems in computer engineering (Networking/Big data/ Intelligent Systems/Cloud Computing/Real time systems).
2) Plan and develop efficient, reliable, secure and customized application software using cost effective emerging software tools ethically.

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included. I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. Atharv Bhilare
2. Naushad Quadri
3. Shaikh Amir Hamza
4. Aditya Utekar

Date:

ACKNOWLEDGEMENT

After the completion of this work, words are not enough to express feelings about all those who helped us to reach goal.

It's a great pleasure and moment of immense satisfaction for us to express my profound gratitude to **Project Guide**, **Prof. Hemalata Gosavi** , whose constant encouragement enabled us to work enthusiastically. His perpetual motivation, patience and excellent expertise in discussion during progress of the project work have benefited us to an extent, which is beyond expression.

We would also like to give our sincere thanks to **Prof. Sujata Bhairnallykar, Head of Department**, and **Dr. Anjali Dadhich**, Project **Co-ordinator** from Department of Computer Engineering, Saraswati college of Engineering, Kharghar, Navi Mumbai, for their guidance, encouragement, and support during a project.

I am thankful to **Dr. Manjusha Deshmukh, Principal,** Saraswati College of Engineering, Kharghar, Navi Mumbai for providing an outstanding academic environment, also for providing the adequate facilities.

Last but not the least we would also like to thank all the staffs of Saraswati college of Engineering (Computer Engineering Department) for their valuable guidance with their interest and valuable suggestions brightened us.

1. Atharv Bhilare
2. Naushad Quadri
3. Shaikh Amir Hamza
4. Aditya Utekar

ABSTRACT

Lane detection is a critical component of autonomous driving systems, and in this project, we present a comprehensive pipeline for lane detection using computer vision techniques. The pipeline encompasses several key stages. It begins with camera calibration, where we compute the camera calibration matrix and distortion coefficients using a set of chessboard images. These parameters are subsequently employed for distortion correction, ensuring that raw images are free from lens distortion that could impact the accuracy of lane detection.

Our method then proceeds to create a threshold binary image by applying a combination of color transforms, gradients, and other techniques. This binary image is then subjected to a perspective transform to achieve a "bird's-eye view" perspective, simplifying the task of detecting lane pixels.

The detected lane boundaries are then warped back onto the original image, providing a visual representation of the lane markings.

INDEX

LIST OF FIGURES

CHAPTER 1
INTRODUCTON

1.1 GENERAL

In the realm of autonomous driving, the ability to accurately detect and track lane boundaries stands as a cornerstone of safety and precision. As the world moves steadily towards autonomous vehicles, it is imperative that we develop advanced lane detection systems that are robust, reliable, and adaptable to the complexities of real-world driving scenarios. This project presents a comprehensive lane detection pipeline that leverages computer vision techniques to fulfill this essential goal.Motivation:The motivation behind this project is deeply rooted in the transformative potential of self-driving technology. With autonomous vehicles poised to reshape transportation and revolutionize industries, the need for cutting-edge lane detection systems has never been more pronounced. These systems are vital for ensuring the safety of passengers and pedestrians while enhancing the overall efficiency and effectiveness of self-driving cars. Our motivation lies in contributing to the advancement of this technology and making roads safer for all.

Lane Detection in Computer Vision:

Lane detection is a crucial component of self-driving systems. It involves the identification and tracking of lane boundaries on the road. Traditional methods primarily rely on computer vision techniques, making it a discipline at the intersection of image processing and machine learning. The aim is to develop algorithms capable of processing image data to locate lane markings accurately.

1.2 OBJECTIVE AND PROBLEM STATEMENT

1.2.1 OBJECTIVE

The objective of the Advanced Driver Assistance System (ADAS) Lane Detection System is to enhance road safety and driver assistance by developing a robust and efficient system capable of accurately detecting and tracking lane markings on roads. Our aim is to offer not only a functional lane detection system but also a clear and insightful demonstration of the underlying computer vision techniques, making it a practical learning resource for those interested in understanding the intricacies of this critical aspect of autonomous driving technology.

The primary objectives of this project are two-fold.

- First, we aim to develop a robust and reliable lane detection system that can accurately identify and track lane boundaries under various conditions, including conditions. The pipeline encompasses camera calibration, distortion correction, thresholding, perspective transformation, lane detection, and curvature calculation.
- Second, we endeavor to provide a well-documented and accessible framework that can serve as a valuable resource for researchers, engineers, and enthusiasts in the field of autonomous vehicle

1.2.2 PROBLEM STATEMENT

The problem addressed by the ADAS Lane Detection System is the need to accurately and reliably detect lane markings on various types of roads and under diverse environmental conditions. Lane detection is a critical component of advanced driver assistance systems, and it poses several challenges.

Develop an advanced perspective lane detection system using computer vision techniques. The system must accurately identify lane boundaries in complex road scenarios, including sharp curves and changing perspectives, to enhance autonomous driving capabilities, improve road safety, and reduce accidents.

Addressing the Challenge of Lighting Conditions: The system must effectively handle variations in lighting conditions, including shadows, glares, and abrupt changes in brightness. It should be capable of adapting to these conditions to ensure reliable lane detection.

Robust Lane Detection: The objective is to create a robust lane detection system that can perform consistently under diverse lighting scenarios, providing a safer and more reliable solution for autonomous vehicles and advanced driver-assistance systems.

CHAPTER 2
LITERATURE REVIEW

Single Image Super- Resolution Method Using OpenCV Networks

A perspective mapping , then transforms semantic lanes from camera's view to top-down view space
to which quadratic polynomials apply for fitting parallel
lane lines by using perspective transformation [1].
Authors : Seonjae Kim , Dongsan Jun, Byung-Gyu Kim , Hunjoo Lee, Eunjun Rhee :

Robustness of Lane Detection Models

Raw images, captured by the vehicle's camera, are subject to lens distortions that can negatively impact lane detection accuracy. To address this issue, we apply distortion correction to the images using OpenCV's undistort() function. By utilizing the calibration matrix and distortion coefficients, we obtain undistorted images that more accurately represent the real-world scene.[2]
Authors: Takami Sato, Qi Alfred Chen

Efficient End-toEnd Accurate Real-Time Lane Detection.

The primary goal is to highlight pixels that are likely part of the lane lines. To achieve this, we employ a combination of thresholding techniques.[3]
Authors: Annika Meyer, Philipp Skudlik, Jan-Hendrik Pauls, Christoph Stiller

Lane Detection with Deep Hough Transform

The curvature of the lane lines is essential for understanding the road's geometry. We employ polynomial fits to both the left and right lane lines to determine the radius of curvature.[4]

Authors: Jia-Qi Zhang, Haoqi Duan

CHAPTER 3

PROBLEM STATEMENT & SCOPE

3.1 PROBLEM STATEMENT:

- The problem addressed by the Advanced Driver Assistance System (ADAS) Lane Detection System is the need for an efficient, reliable, and robust solution for detecting and tracking lane markings on various road types and under diverse driving conditions.

3.2 SCOPE:

1. Sensor Integration: Integration of sensors such as cameras, lidar, radar, and ultrasonic sensors to enhance lane detection accuracy and reliability. This involves optimizing sensor placement and fusion techniques.

2. Environmental Adaptability: Creating algorithms that can adapt to various environmental conditions, such as inclement weather (rain, snow, fog), low-light situations, and varying road surface conditions.

3. Lane Geometry Recognition: Extending the system's capabilities to recognize complex road geometries, including intersections, roundabouts, merges, and diverges. The scope may include multi-lane recognition and lane change detection.

4. Human-Machine Interface (HMI): Developing a user-friendly HMI that communicates lane detection information to the driver effectively, providing visual and auditory alerts or assistance.

CHAPTER 4

PROPOSED SYSTEM

4.1 ALGORITHM

Algorithm for Lane Detection with OpenCV:

1. Camera Calibration:

   - Read a set of calibration images of a chessboard.
   - Apply camera calibration to undistort images using the calibration data.

2. Distortion Correction:

   - Capture or load test images or video frames.
   - Apply distortion correction using the calibration data to remove lens distortion.

3. Thresholding:

   - Apply color and gradient thresholding to enhance lane line features.
   - Experiment with different color spaces and thresholding techniques to optimize results.

4. Perspective Transformation:

   - Define source and destination points for perspective transformation.
   - Warp the thresholded image to a top-down view of the road (bird's-eye view).

5. Lane Detection:

- Implement a sliding window or polynomial fitting approach to identify lane line pixels.
- Visualize the detected lane lines on the original image.

6. Video Processing Loop (for testing):

- Open a video file for input or use real-time video capture.
- For each frame in the video:
  - Apply distortion correction to the frame.
  - Apply thresholding.
  - Perform perspective transformation.
  - Detect the lane lines.
  - Draw the lane lines on the frame.
  - Output the frame to an output video.

7. Save Output Video:

- Save the processed frames into an output video file.

8. Optimize and Fine-Tune:

- Experiment with different parameter settings and techniques to optimize lane detection.
- Handle challenging scenarios, such as shadows, road markings, and changing lighting conditions.

PSEUDOCODE :

```
while video is available:

    frame = read_next_frame(video)  # Read the next frame from the video feed

    undistorted_frame = undistort(frame)  # Apply distortion correction

    # Apply perspective transformation

    thresholded_frame = apply_thresholding(undistorted_frame)

    warped_frame = warp_perspective(thresholded_frame)

    left_lane, right_lane = detect_lane_lines(warped_frame)  # Detect lane lines

    draw_lane_lines(frame, left_lane, right_lane)  # Draw detected lane lines

    # Test the Model

    output_video.write(frame)  # Write the frame to the output video

output_video.release()  # Close the output video file
```

4.2 DESIGN DETAILS

Designing a lane detection project involves breaking down the project into various components and detailing how each component will be implemented. Here are design details for the lane detection project using OpenCV :

Project Design Details: Lane Detection with OpenCV

1. Camera Calibration:

- Collect a set of calibration images of a chessboard with known dimensions.
- Implement a calibration process to compute the camera matrix and distortion coefficients using OpenCV's camera calibration functions.
- Save the calibration data for distortion correction.

2. Distortion Correction:

- Apply the camera matrix and distortion coefficients to undistort images or video frames.
- Implement a function for distortion correction, using the calibration data.

3. Thresholding:

- Implement a thresholding pipeline that combines color and gradient thresholds.
- Experiment with various color spaces (e.g., RGB, HLS, LAB) and gradient techniques (e.g., Sobel operators).
- Fine-tune thresholding parameters for optimal lane detection.

4. Perspective Transformation:

- Define source and destination points for perspective transformation. These points should form a trapezoid shape in the original and transformed views.

- Implement a perspective transformation function to warp the image to a top-down view of the road.

- Ensure that the perspective transformation is applied consistently to all frames.

5. Lane Detection:

- Implement a lane detection algorithm using the top-down perspective view.

- Consider techniques like sliding windows or polynomial fitting to locate and track lane lines.

- Create functions to identify lane pixels, fit curves, and calculate lane curvature.

6. Video Processing Loop (for testing):

- Design a video processing loop to handle video input for testing.

- Open the input video file or use real-time video capture as needed.

- Process each frame through the pipeline (undistortion, thresholding, perspective transformation, lane detection, curvature calculation).

- Continuously update and draw the detected lane lines on the frames.

7. Lane Visualization:

- Create functions to overlay lane markings, lane curvature, and vehicle position on the original frames.

- Design a graphical user interface (GUI) for displaying the processed video with lane detection results.

8. Output Video Creation:

- Save the processed frames into an output video file to generate the final result.

- Utilize OpenCV functions to create and write the output video.

9. Optimization and Fine-Tuning (Next Sem):

- Experiment with different parameters, techniques, and configurations to optimize lane detection accuracy and reliability.

- Address challenges such as shadows, varying road conditions, and changing lighting.

10. Error Handling (Next Sem):

- Implement error handling to handle unexpected scenarios, such as missing calibration data or corrupted video input.

- Include informative error messages and logs for debugging.

11. Documentation:

- Creating detailed documentation that includes the project's purpose, components, parameters, and usage instructions.

- Documenting the code, explaining the purpose of each function and component.

12. Testing and Validation:

- Testing the project extensively using a variety of video clips with different road conditions and scenarios.

- Validating the results against ground truth data, such as annotated lane positions.

13. User Interface (Next Sem):

- Building a user-friendly interface for users to interact with the lane detection system.

- Including options for video selection, parameter adjustment, and result visualization.

IMPLEMENTATION DETAILS AND RESULTS

IMPLEMENTATION:

In this chapter, we provide a comprehensive overview of the technical aspects of implementing the lane detection pipeline. We delve into the software and library dependencies, the data used for calibration and testing, the structure of the project's code, and the key modules that facilitate the lane detection process.

Software and Library Dependencies:

Our implementation relies on several software and library dependencies, making it essential to ensure compatibility and availability. The primary dependencies include Python 3.5, NumPy, OpenCV-Python, Matplotlib, and Pickle. Python 3.5 serves as the programming language, providing a robust and versatile environment for image processing and computer vision tasks. NumPy, a fundamental numerical library, aids in efficient array manipulation and mathematical operations. OpenCV-Python is a key component, offering a rich suite of computer vision tools, while Matplotlib facilitates data visualization. Pickle, a Python module, is employed for serializing and deserializing objects, enabling the storage and retrieval of essential calibration data.

Data Used:

Calibration is a pivotal component of the lane detection process. We utilize a set of chessboard images for camera calibration, which are found in the 'camera_cal' directory. These images are captured under various angles and positions to ensure accurate calibration matrices and distortion coefficients. Additionally, we employ test videos, including 'project_video.mp4,' to validate and demonstrate the lane detection pipeline. The final annotated video output, 'out.mp4,' is generated during the process.

The ability to use and process video data enables the lane detection system to be applied in real-world driving scenarios.

Code Structure and Modules:

The project's code is organized into distinct modules and directories to maintain a structured and manageable implementation. The primary code modules include 'calibrate_camera.py,' 'combined_thresh.py,' 'perspective_transform.py,' 'line_fit.py,' and 'Line.py.'

'calibrate_camera.py' is responsible for camera calibration, as it computes the calibration matrices and distortion coefficients based on chessboard images.

'combined_thresh.py' focuses on image preprocessing and thresholding techniques, creating thresholded binary images for further processing.

'perspective_transform.py' handles the perspective transformation, warping the binary image to achieve a top-down view.

'line_fit.py' is the core of the lane detection process, employing sliding window search and polynomial curve fitting.

'Line.py' serves as a helper class for line detection, enabling the management of historical lane data, smoothing of lane parameters, and calculations of lane curvature and vehicle offset.

By presenting these implementation details, we provide a comprehensive understanding of the technical framework that supports our lane detection pipeline.

RESULTS:

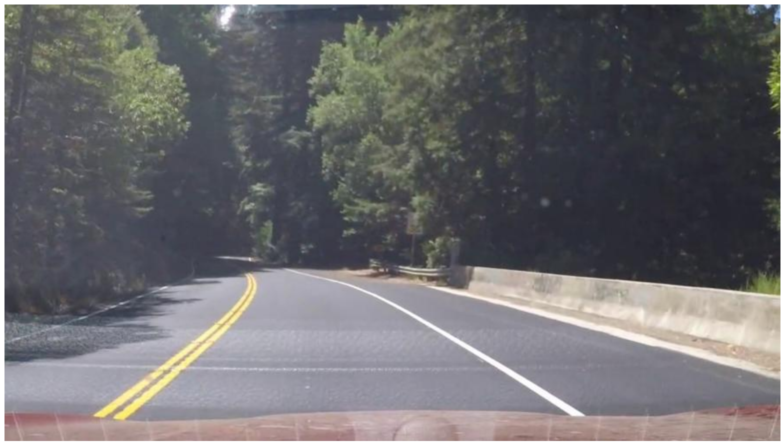

Fig 4.2.1 – Input Video Frame

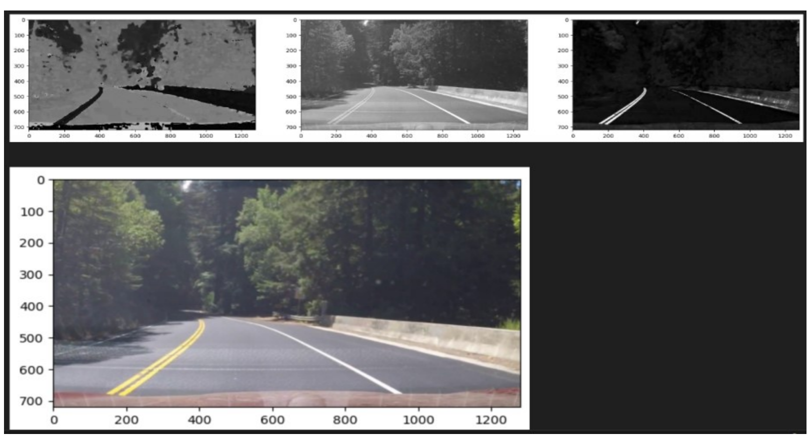

Fig 4.2.2 – Gaussian Filters

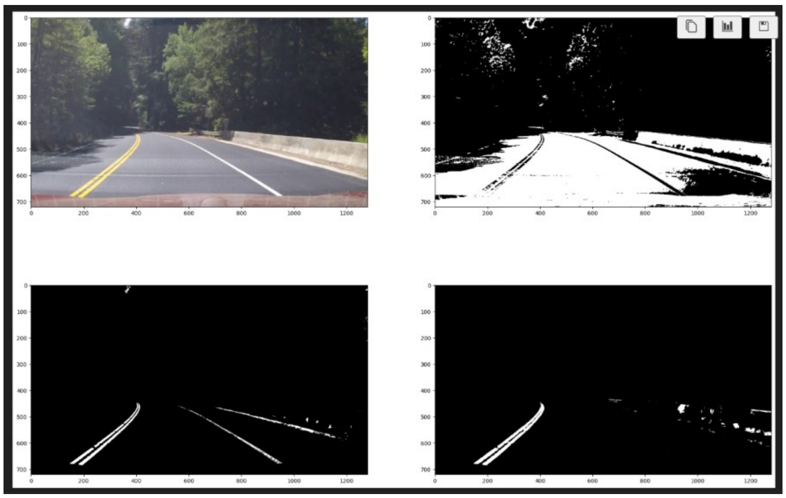

Fig 4.2.3 – Thresholding Layers

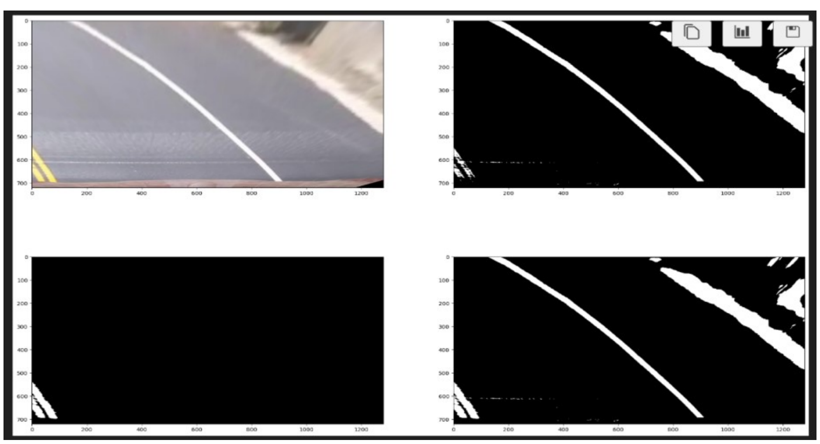

Fig 4.2.4 Perspective Transformation

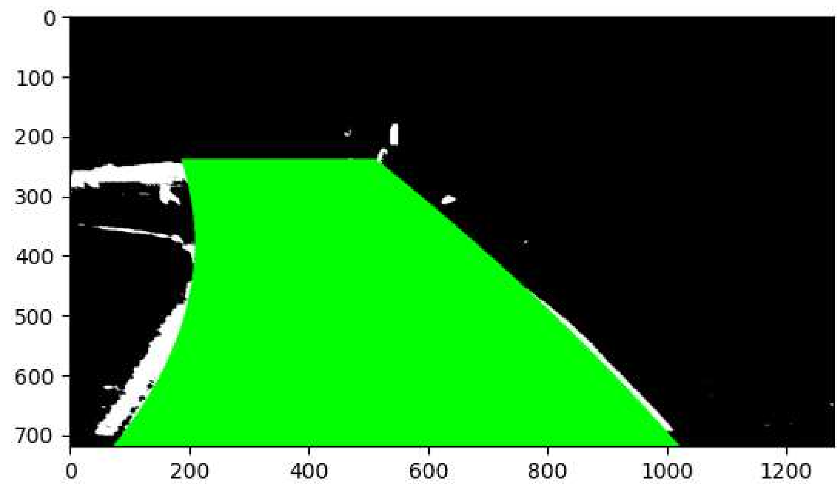

Fig 4.2.5 Lane Line Detection

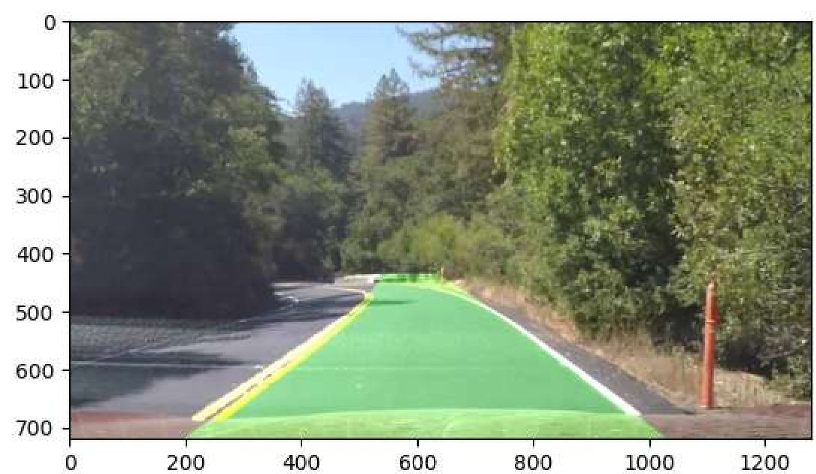

Fig 4.2.6 Final Output Frame

4.3 FLOWCHART

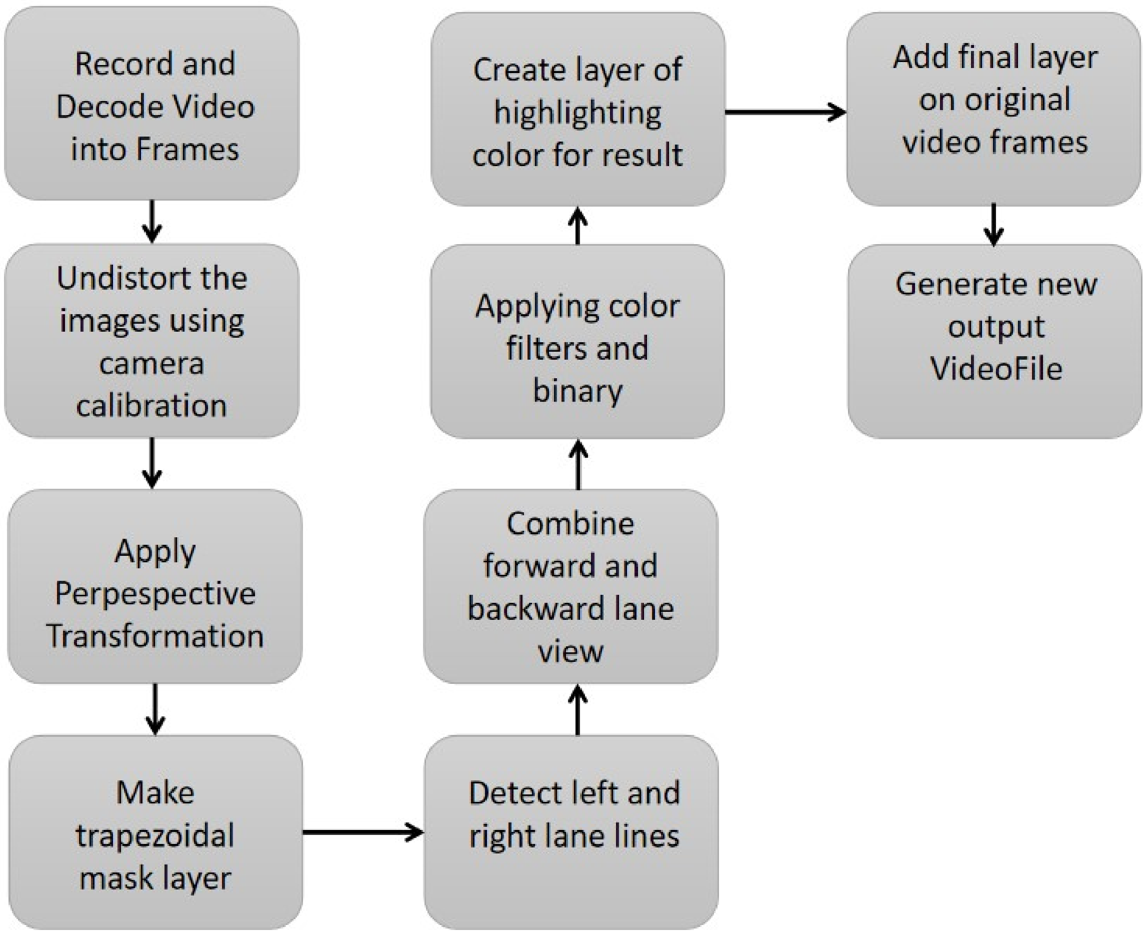

Fig 4.3.1 Flowchart for Lane Detection process

CHAPTER 5

REQUIREMENTS

5.1 HARDWARE & SOFTWARE

HARDWARE:

1. RAM 4Gb
2. Intel i3+Processor
3. Portable Camera 20MP+
4. Storage 5GB+

SOFTWARE:

1. Web Browsrer
2. VS code
3. Python 3.5

Libraries Used:

1. Numpy
2. OpenCV2
3. Python 3.5 IDE
4. Matplotlib

CHAPTER 6

IMPLEMENTATION PLAN FOR NEXT SEM

Graphical User Interface (GUI):

Develop a user-friendly GUI to interact with the system.

Include features for video selection, start/stop control, parameter adjustment sliders, and display options for visualizing lane detection results in real-time.

Allow users to adjust key parameters such as color thresholds, perspective transformation points, and sensitivity to adapt to different road and lighting conditions.

Nighttime Lane Detection:

Implement adaptive thresholding that can automatically adjust to changing lighting conditions, such as transitioning from day to night.

Dynamic Region of Interest (ROI):

Implement an algorithm that dynamically adjusts the region of interest based on the road's curvature and vehicle speed.

Shadow and Glare Handling:

Develop algorithms to handle shadows and glares more effectively, including strategies for shadow removal and glare reduction in the image.

Real-time Performance Optimization:

Investigate optimization techniques to achieve real-time performance, such as GPU acceleration, parallel processing, and code optimization.

CHAPTER 7
CONCLUSION

In conclusion, the development and implementation of an Advanced Driver Assistance System (ADAS) Lane Detection System represent a significant advancement in automotive safety and autonomous driving technology. This system plays a pivotal role in enhancing road safety, providing real-time driver assistance, and contributing to the vision of self-driving vehicles.

In essence, ADAS Lane Detection Systems are a vital component of the evolving automotive landscape. They offer a glimpse into the future of safer, more efficient, and increasingly autonomous driving. As technology advances and the automotive industry continues to innovate, these systems will continue to evolve, pushing the boundaries of what is possible in terms of driver assistance and road safety.

REFERENCES

In the reference chapter, we cite a selection of research papers and documentations that have been instrumental in the development and understanding of the lane detection project. These resources have been pivotal in shaping the methodologies, algorithms, and best practices applied in our project. Below are some key references:

1. https://www.semanticscholar.org/paper/End-to-end-deep-learning-of-lane-detection-and-path-Lee-Liu/d6233efc96f66b420bae07086aa21048baf36606

2. https://arxiv.org/pdf/2107.02488.pdf

3. https://pubs.acs.org/doi/abs/10.1021/ac034984k

4. https://www.semanticscholar.org/paper/HoughLaneNet%3A-Lane-Detection-with-Deep-Hough-and-Zhang-Duan/da807d87c96b2ff3b724abe7fba429a3f20837ac

5. https://arxiv.org/abs/2206.10692

6. https://arxiv.org/abs/2007.12147

These references have played a significant role in shaping the methodologies and techniques employed in our lane detection project. They serve as foundational resources for understanding and implementing the core components of our pipeline.

/

CHAPTER 8

CERTIFICATION AND PUBLICATION