PROJECT REPORT

ON

"LANE DETECTION"

Project Report submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

(2021-2025)

IN

Computer Science & Engineering

By

PIYUSH ANAND **(CS21B1019)**

NEEHARIKA TELU **(CS21B1029)**

Under the guidance of

Dr. GYANESHWAR DUBACHERLA, Ph.D

Assistant Professor

ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು

भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर

Indian Institute of Information Technology Raichur

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY RAICHUR

Transit in Government Engineering College (GEC) Yermarus

KARNATAKA,INDIA - 584135

CANDIDATE'S DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated or falsified any idea/data/fact/source in our submission. We 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 that have thus not been properly cited or from whom proper permission has not been taken when needed.

PLACE: RAICHUR

DATE: 30/04/2024

PIYUSH ANAND (CS21B1019)

NEEHARIKA TELU (CS21B1029)

ACKNOWLEDGEMENT

We wish to express our sincerest gratitude to Gyaneshwar Dubacherla for the continuous guidance and mentorship that he provided us during the project. He showed us the path to achieve our targets by explaining all the tasks to be done and explaining to us the importance of this project as well as its industrial relevance. He was always ready to help us and clear our doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

PLACE: RAICHUR

DATE: 30/04/2024

PIYUSH ANAND (CS21B1019)

NEEHARIKA TELU (CS21B1029)

INDEX

AIM **5**
OBJECTIVE **5**
METHODOLOGY **5**
MOTIVATION **6**
EXPECTED RESULTS **7**
CHALLENGES **9**
CURVED ROAD DETECTION **11**
ALGORITHM 11
FINAL RESULT 13
1. Improved Curved Road Detection 13
2. Directional Guidance Implementation 13
ADDITIONAL CHALLENGES **13**
1. Foggy Weather 14
2. The camera angle is not so good 15
3. Sand (No proper road) 17
4. Too much gap between lane lines 18

AIM

This project aims to develop a road lane detection system using Python and computer vision algorithms.

OBJECTIVE

The objective of this project is to implement a road lane detection system capable of accurately identifying lane markings under various conditions, including night, curved roads, ring roads, zebra crossings, foggy and rainy conditions, and sandy roads. The system will process video input and provide real-time lane detection, contributing to the advancement of autonomous driving technology.

METHODOLOGY

The project is implemented in Python using the OpenCV library for image processing.

The following steps outline the implementation process:

1. Video Capture and Decoding: Video files are captured and decoded using the VideoFileClip object from the moviepy library.
2. Grayscale Conversion: RGB frames are converted to grayscale to simplify processing.
3. Noise Reduction: Gaussian blur is applied to reduce noise in the frames.
4. Edge Detection: The Canny edge detection algorithm is used to detect edges in the frames.
5. Region of Interest: A mask is applied to focus only on the region covering the road lanes.
6. Hough Line Transform: The Probabilistic Hough line transform is applied to detect straight lines representing lane boundaries.
7. Drawing Lane Lines: Detected lane lines are drawn on the original frames.

8. Real-time Processing: The lane detection algorithm is applied to process video frames in real-time.
9. Evaluation: The performance of the system is evaluated based on the accuracy of lane detection and its suitability for autonomous driving applications.

MOTIVATION

1. **Autonomous Driving:** The road lane detection system enables vehicles to navigate autonomously within lanes, contributing to the development of self-driving cars.

2. **Advanced Driver Assistance Systems (ADAS):** The technology can be integrated into ADAS to provide lane departure warnings and lane-keeping assistance to drivers.

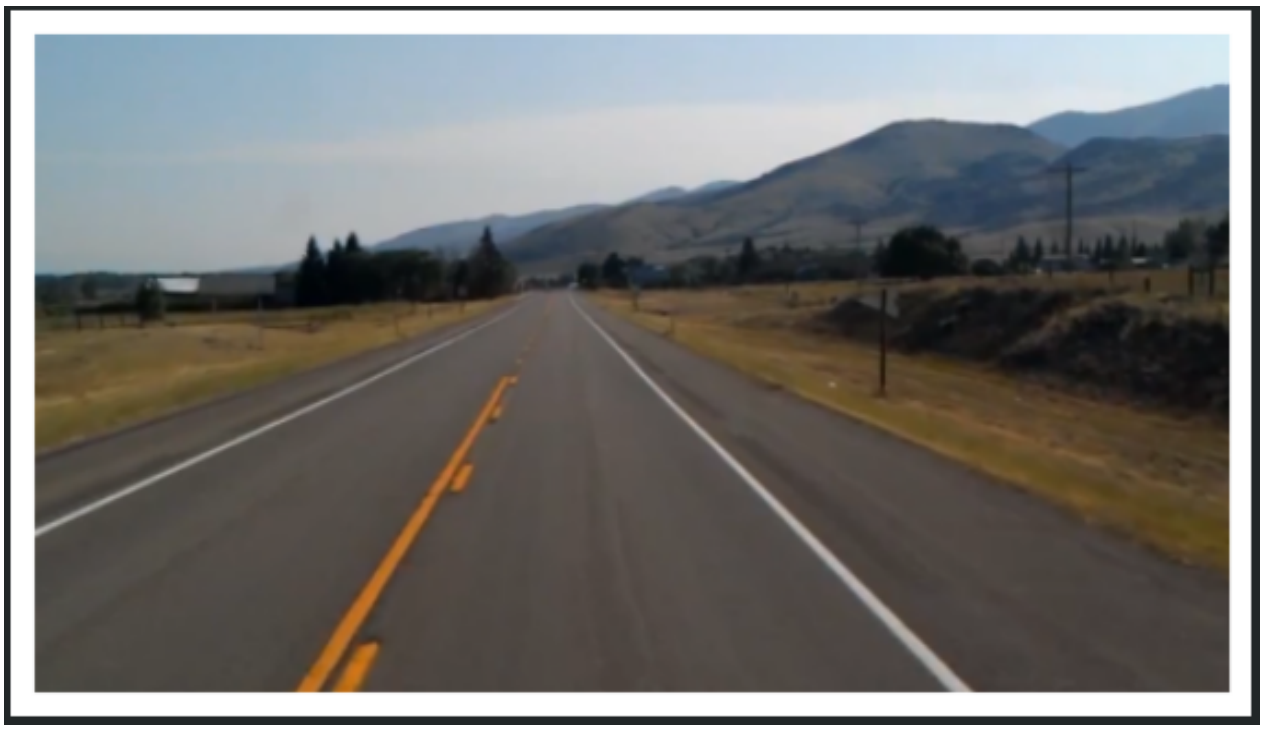
3. **Safety Enhancement:** Accurate lane detection helps improve road safety by reducing the risk of lane departures and collisions.

4. **Research and Development:** The project serves as a foundation for further research and development in the field of computer vision and autonomous driving.

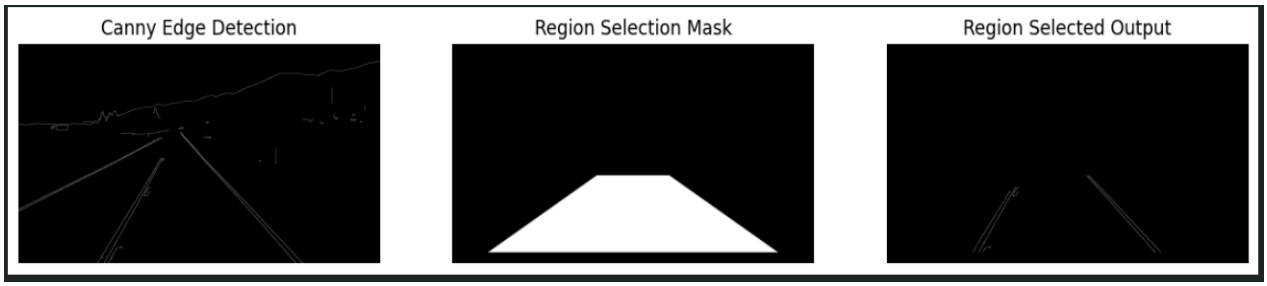
EXPECTED RESULTS

A) The core objective of our project is to create a lane detection system that possesses the capability to process not only static images but also dynamic video footage. Our focus will initially be directed towards perfecting the algorithms necessary for image processing. This approach allows us to thoroughly refine and optimize the techniques employed in detecting lane markings before transitioning to the more complex task of processing video streams.

Input Image



Output Image



1. Grayscale Image :
The original input image is converted to grayscale. This simplifies the image by reducing it to a single channel, which can help in detecting edges and other features.

2. Gaussian Blur :
Gaussian blur is applied to the grayscale image. This step helps to smooth out noise and minor variations in intensity, which can improve the performance of edge detection algorithms.

3. Canny Edge Detection :
Canny edge detection is performed on the blurred grayscale image. This algorithm detects edges by looking for significant changes in intensity and highlighting them as white lines against a black background.

4. Canny Edge Detection (Zoomed) :
This is a zoomed-in view of the Canny edge detection output. It provides a closer look at the detected edges, which are crucial for identifying lane markings.

5. Region Selection Mask :
A region of interest (ROI) mask is applied to the edge-detected image. This mask selects the portion of the image where lane markings are typically located and masks out irrelevant areas.

6. Region Selected Output :
The final output of the lane detection process, showing only the lane markings within the selected region of interest. This output is obtained by applying the region selection mask to the edge-detected image.

B) For the **second evaluation, our primary objectives** are:

During our first evaluation, we encountered challenges with accurately detecting lanes on curved roads due to distortion in camera perspective. This challenge affected the project's ability to identify and track lane markings effectively, leading to less reliable lane detection results.

1. **Improved Curved Road Detection**: We will refine our lane detection algorithms to accurately identify and track lane markings on curved roads. This entails enhancing our approach to handle the curvature of lanes effectively, ensuring precise detection even in challenging scenarios.

2. **Directional Guidance Implementation**: We will integrate functionality to determine the appropriate direction (right or left) based on the curvature and design of the road. This involves analyzing the curvature of detected lanes and providing real-time guidance to the vehicle, enabling it to navigate safely through curves by indicating whether to turn right or left.

By focusing on enhancing lane detection accuracy on curved roads, we aim to make the project more robust and suitable for real-world driving scenarios. This aligns with our objective of developing an effective lane detection system to advance autonomous driving technology.

CHALLENGES

1. Curved Road Detection: Lane detection algorithms that we used in the project are often optimized for straight road scenarios. **When encountering curved roads, the lanes may not follow a straight path**, making it difficult to accurately detect and represent them. As a result, the system fails to identify the correct curvature of the lanes, leading to inaccurate lane predictions.

1. **Lane Boundary Ambiguity**: In curved roads, the boundaries between lanes can become ambiguous, especially in areas where lanes merge or split. This ambiguity makes it challenging to distinguish between different lanes and accurately determine their boundaries.

2. **Varying Lane Marking Patterns**: Lane markings on curved roads may exhibit varying patterns due to factors such as road curvature, road width, and lane merging. These variations can confuse the detection algorithm, as it may not be able to generalize lane marking patterns effectively across different road geometries.

3. **Limited Field of View**: Curved roads often require a wider field of view to detect lane markings effectively. However, the system has limitations in terms of the camera angle or resolution, which can impact its ability to detect lanes accurately, especially in curved sections of the road.

CURVED ROAD DETECTION

ALGORITHM

1. **Calibration and Undistortion**: The script starts by importing calibration parameters and functions for undistorting images from external modules. This step ensures that distortions caused by the camera lens are corrected, providing accurate representations of the road scene.

2. **Image Thresholding**: It defines functions for applying Sobel edge detection, calculating gradient magnitude and direction, and thresholding HLS color channels. These operations help in isolating lane lines from the rest of the image.

3. **Lane Detection Pipeline**: The main execution block contains a pipeline for processing images or frames from a video feed. This pipeline includes:

- Correcting distortion in the input image or frame.
- Combining gradient and color thresholding results to identify potential lane line pixels.
- Warping the perspective of the image to obtain a bird's-eye view of the road.
- Applying a sliding window algorithm to detect lane lines.
- Drawing detected lane boundaries on the original image or frame.
- Providing visualizations of road status and lane detection results.

Input Handling: The script provides options to process either images or videos, and it includes commented-out lines for specifying the input type and name.

Visualization and User Interaction: It uses OpenCV to display intermediate and final results, allowing users to interact with the system through keyboard commands (e.g., pausing the video feed or quitting the application).
Output: If the input type is a video, the script processes frames sequentially, applying the lane detection pipeline to each frame and displaying the results in real-time. If the input type is an image, the script processes the image once and displays the final result.

Input Video Snap

Output

FINAL RESULT

1. **Improved Curved Road Detection**

   We have significantly enhanced our lane detection algorithms to address the challenges encountered during the evaluation, particularly in accurately identifying and tracking lane markings on curved roads. By refining our approach, we have achieved more precise detection even in challenging scenarios characterized by distortion in camera perspective. Our improvements ensure that the system can effectively handle the curvature of lanes, resulting in more reliable lane detection results across various road conditions.

2. **Directional Guidance Implementation**

   Integration of directional guidance functionality marks a significant advancement in our project. By analyzing the curvature of detected lanes in real-time, our system can now provide actionable guidance to the vehicle,

enabling it to navigate safely through curves. This implementation considers factors such as the curvature and design of the road, allowing the system to determine the appropriate direction (right or left) for the vehicle. This feature enhances the overall autonomy and safety of the vehicle by assisting the driver in making informed navigation decisions.

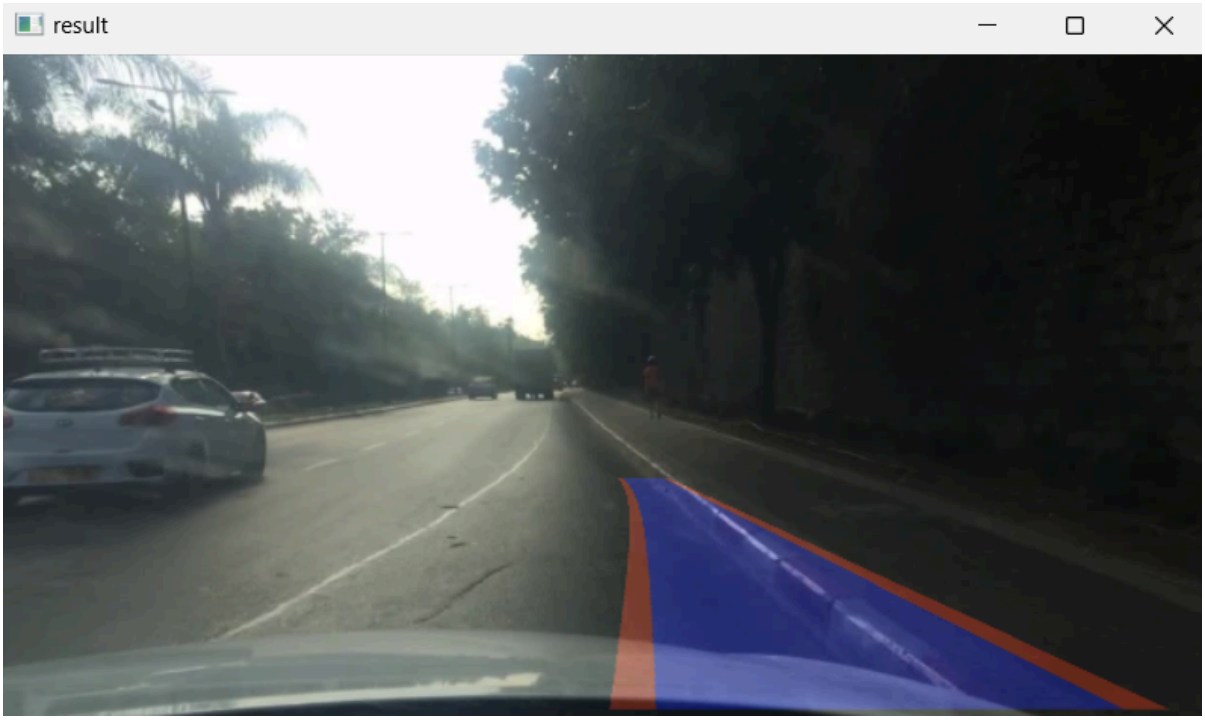
ADDITIONAL CHALLENGES

Further Areas for Improvement:
While our project has made significant progress, there are still areas where our lane detection system requires improvement:

1. Foggy Weather

Input

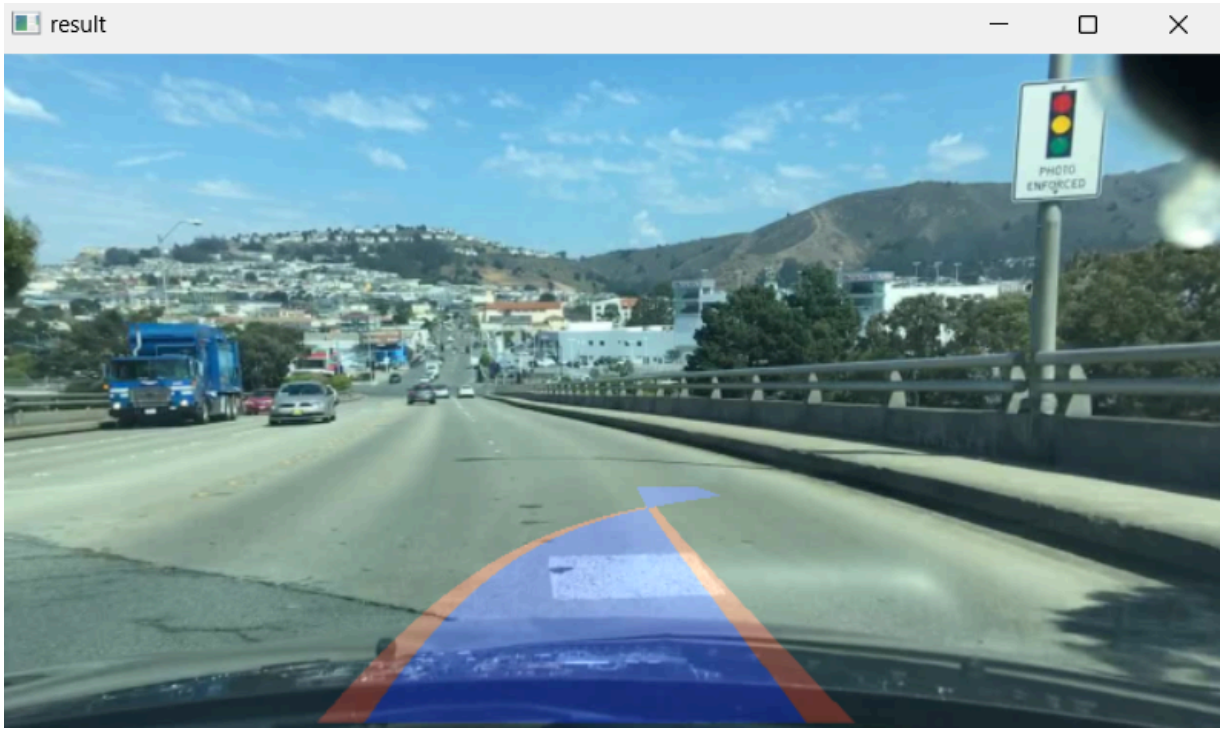
Output



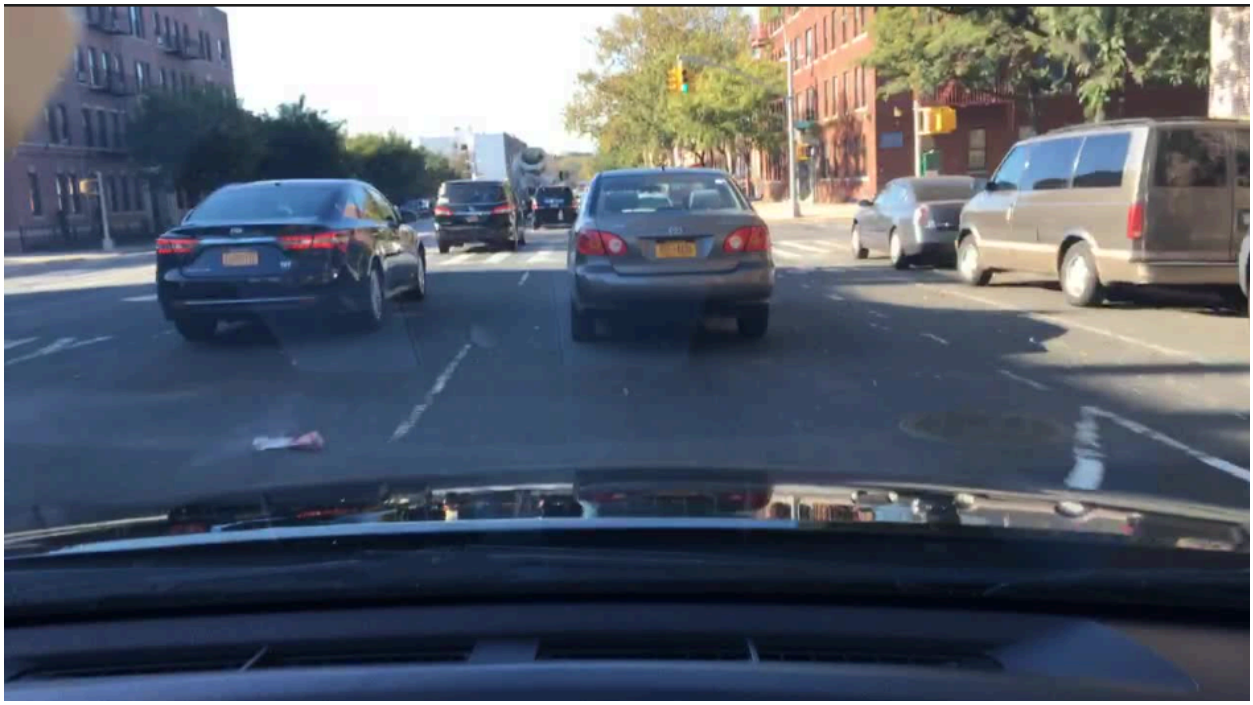
2. The camera angle is not so good

Input 1

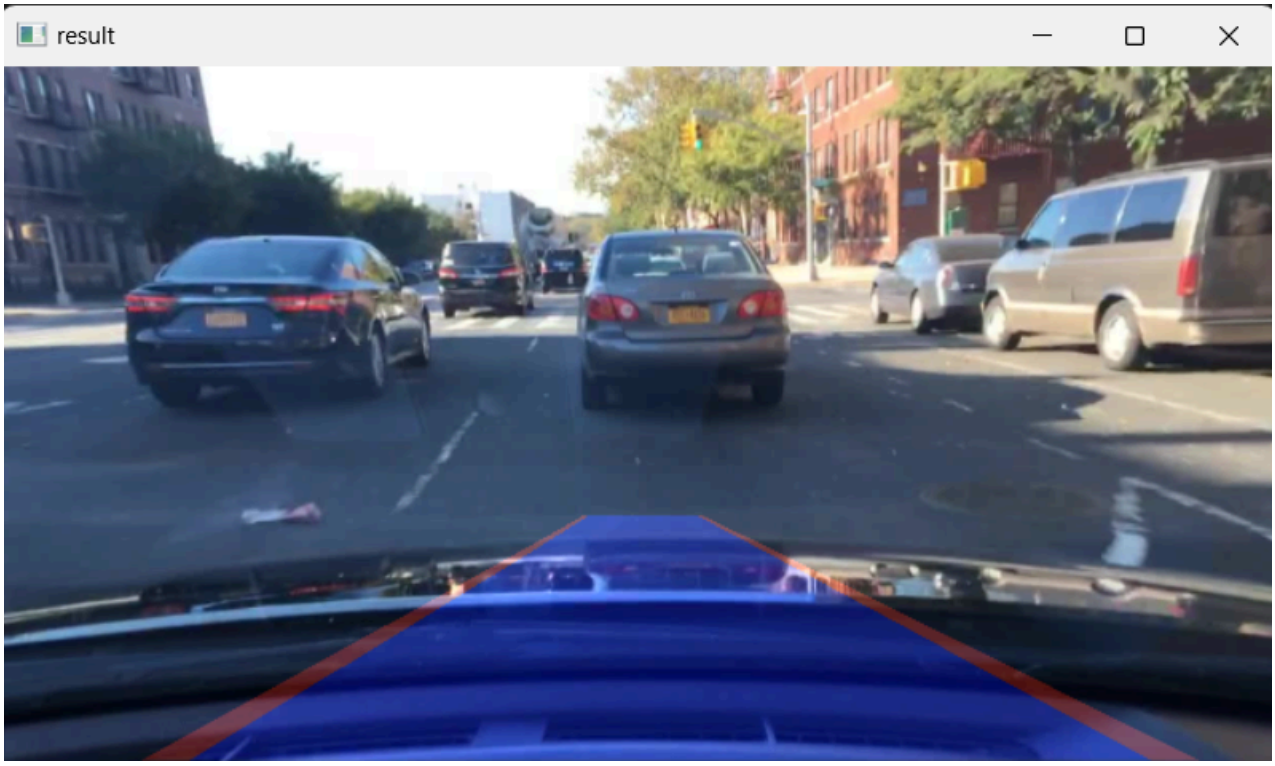
Output1



Input2



Output2



3. Sand (No proper road)

Input

Output

4. Too much gap between lane lines

Input

Output