International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

DRIVING AUTOMATION USING LANE DETECTION ALGORITHM

P. Mansa Devi[1], Riddhi Chakrabarti [2], Vineet Mathireddi[3], Alok Kumar[4] and Ravi Kiran[5]

[1] Assistant Professor, Dept. of CSE, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India.
[2345]Student, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India.

---***---

Abstract - *The project "Driving Automation Using Lane Detection Algorithms" aims to revolutionize the automotive industry by harnessing the power of computer vision and machine learning. With the proliferation of autonomous vehicles, lane detection plays a pivotal role in ensuring safe and reliable self-driving capabilities. Our research focuses on developing state-of-the-art lane detection algorithms that can accurately identify and track road lanes under diverse and challenging real-world conditions. By leveraging advanced image processing techniques and deep learning models, we seek to provide vehicles with the ability to interpret complex road environments, including highways, urban streets, and adverse weather scenarios. This abstract explores the integration of Canny Edge Detection for lane detection in the context of driving automation. Lane detection is a critical component of autonomous vehicles and advanced driver assistance systems (ADAS), enabling precise vehicle positioning within road lanes. Canny Edge Detection, a classical computer vision technique, is employed to identify lane boundaries, offering a computationally efficient and robust solution. This abstract provides an overview of the approach's potential to contribute to safer and more efficient autonomous driving systems by leveraging the strengths of Canny Edge Detection for comprehensive scene understanding and accurate lane detection.*

***Key Words*: Driving Automation, Lane detection Algorithm, Autonomous Vehicles, Canny Edge Detection, Advance Driver Assistance Systems (ADAS), Road Lane Detection**

1. INTRODUCTION

Using computer vision and machine learning, the groundbreaking project "Driving Automation Using Lane Detection Algorithms" aims to revolutionize the field of autonomous driving. To drive safely on roads, humans depend on their ability to see and obey lane markings. Similarly, autonomous cars need to be able recognize and follow these cues with unparalleled accuracy. There are currently two accepted approaches: the model-based approach and the feature-based method. There are currently two well-established techniques for using video to perform lane recognition: the feature-based method and the model-based method. Our project's mainstay, canny edge detection, is essential for correctly extracting lane markings from video streams. When combined with other image processing and computer vision algorithms, this method aids in helping cars recognize where they are in relation to lane boundaries and make judgment calls to stay in their designated lanes. The captured video frames were processed using the Canny edge detection algorithm to look for edges. After that, additional processing was done on these edges to identify lane markings. By implementing this plan, we hope to develop a lane-detection system that is dependable and advances the field of autonomous driving technology.

2. LITERATURE REVIEW

Vision-Based Robust Lane Detection and Tracking in Challenging Conditions: The first paper proposes a method using three key technologies to detect road lane markings under difficult conditions such as changing illumination. First, to handle different types of lane edges, edge detection is improved by the Comprehensive Intensity Threshold Range (CITR). Second, the Hough Transform and geometric constraints are used by the Two-Step Lane Verification Technique to confirm the lane characteristics. Finally, even in situations where markings are partially or completely invisible, the Novel Lane Tracking Method forecasts lane positions by using historical frames. Testing on various datasets demonstrates detection rates of 97.55% and processing times of 22.33 ms per frame, which are faster than state-of-the-art techniques. The algorithm performs well in real-world lane marking detection scenarios, especially when there is noise present.

Vehicle Lane Change Prediction on Highways Using Efficient Environment Representation and Deep Learning: The second paper introduces a novel lane detection method using light field (LF) technology to improve prediction accuracy and robustness in intelligent transportation systems. Deep convolutional neural networks (CNNs) are promising for lane detection, but they have difficulty generalizing to different types of road conditions. The LF-based method offers better performance and robustness by making use of additional angular information that LF captures. Experimental results demonstrate the effectiveness of the proposed method in improving lane line point prediction accuracy and robustness against adverse conditions, outperforming traditional image-based lane detection methods. All things considered, the LF-based lane detection approach holds promise for resolving issues with conventional techniques and enhancing prediction robustness and accuracy.

Real-Time Road Curb and Lane Detection for Autonomous Driving Using LiDAR Point Clouds: The third paper introduces a real-time method for detecting lane markings using LiDAR point clouds to enhance autonomous driving safety and efficiency. The method includes ROI selection using constrained RANSAC, road curb detection based on segment point density, and lane marking identification using an adaptive threshold selection method. Evaluation on five datasets shows high Precision, Recall, Dice, and Jaccard Index scores (82.82% to 94.49%), indicating accurate lane marking detection. The method achieves an average detection range per frame of 33.01m to 47.14m and a computation time of less than 50 milliseconds per frame, demonstrating its effectiveness and feasibility for real-time operation. This approach offers efficient and reliable lane marking detection using cost-effective 16-beam LiDAR technology, benefiting autonomous driving systems.

3. SYSTEM METHODOLOGY

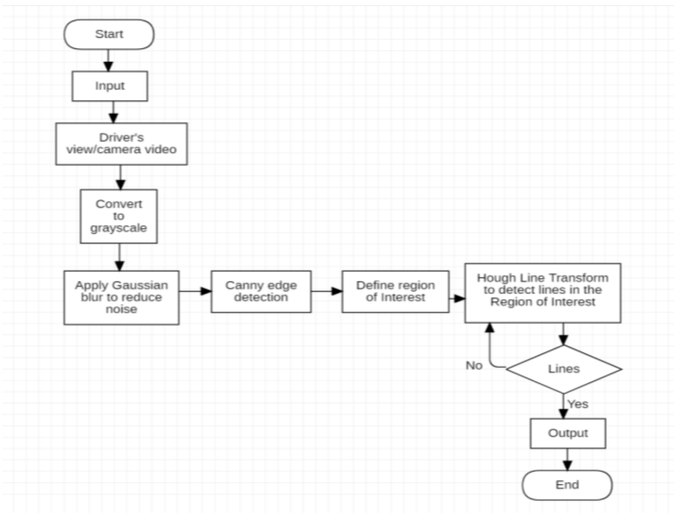


Figure-1: System Architecture

1. **Create a dark grayscale version of the original image**: Convert a grayscale version of the original image. In order to reduce the image to a single channel (intensity) and preserve crucial features for edge detection, this step is crucial.

2. **Apply Gaussian Blur**: Give the grayscale image a Gaussian blur. By lowering false positives, this step helps to smooth out the image and reduce noise, which can enhance the effectiveness of the edge detection algorithm.

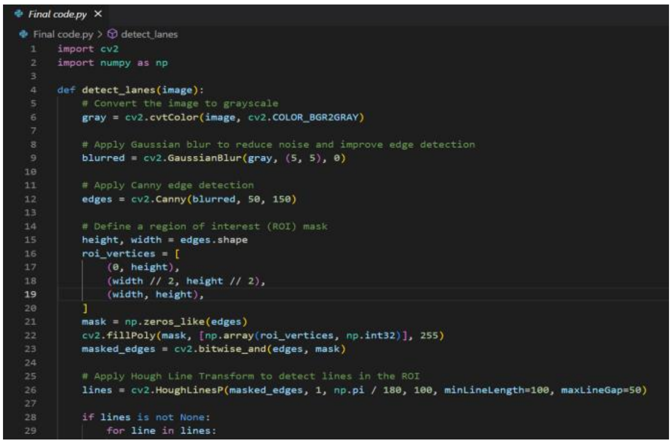
3. **Use Canny Edge Detection to obtain edges**: On applying the Canny edge detection algorithm to the blurred grayscale image. Canny edge detection is a multi-step process that involves detecting strong edge gradients, suppressing weak edges, and tracking along strong edges to produce a binary edge map.

4. **Determine the Interest Region:** In the image, identify the region of interest (ROI) where lane lines should appear. This aids in removing unwanted edges, such as those from surrounding objects or the sky, that the edge detector might have detected outside of this area.

5. **Recover the Hough lines**: On the edge-detected image, apply the Hough transform to find lines that might be lane markings. One method for identifying straight lines in an image is the Hough transform, which is especially helpful for identifying lane markings.

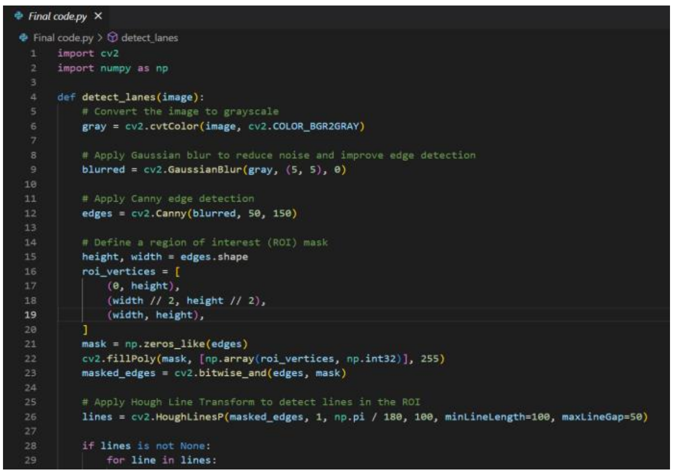
6. **Draw the Hough lines onto the original image**: Once the Hough lines are detected, consolidate and extend the Hough lines to create full lane lines after they have been identified. Then, draw these lines onto the original image to visualize the detected lanes.

4. IMPLEMENTATION



```python
import cv2
import numpy as np

def detect_lanes(image):
    # Convert the image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur to reduce noise and improve edge detection
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)

    # Apply Canny edge detection
    edges = cv2.Canny(blurred, 50, 150)

    # Define a region of interest (ROI) mask
    height, width = edges.shape
    roi_vertices = [
        (0, height),
        (width // 2, height // 2),
        (width, height),
    ]
    mask = np.zeros_like(edges)
    cv2.fillPoly(mask, [np.array(roi_vertices, np.int32)], 255)
    masked_edges = cv2.bitwise_and(edges, mask)

    # Apply Hough Line Transform to detect lines in the ROI
    lines = cv2.HoughLinesP(masked_edges, 1, np.pi / 180, 100, minLineLength=100, maxLineGap=50)

    if lines is not None:
        for line in lines:
```

Figure-2: Screenshot 1 of code



```python
import cv2
import numpy as np

def detect_lanes(image):
    # Convert the image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur to reduce noise and improve edge detection
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)

    # Apply Canny edge detection
    edges = cv2.Canny(blurred, 50, 150)

    # Define a region of interest (ROI) mask
    height, width = edges.shape
    roi_vertices = [
        (0, height),
        (width // 2, height // 2),
        (width, height),
    ]
    mask = np.zeros_like(edges)
    cv2.fillPoly(mask, [np.array(roi_vertices, np.int32)], 255)
    masked_edges = cv2.bitwise_and(edges, mask)

    # Apply Hough Line Transform to detect lines in the ROI
    lines = cv2.HoughLinesP(masked_edges, 1, np.pi / 180, 100, minLineLength=100, maxLineGap=50)

    if lines is not None:
        for line in lines:
```

Figure-3: Screenshot 2 of code

1. The code reads the input video frame by frame with cv2.VideoCapture.

2. It initialises variables for averaging lines and determines an average factor for smoothing.

3. Within the loop, it reads each frame of video.

4. It uses the detect_lanes function to detect lanes in the current frame.

5. **detect_lanes Function:** This function accepts an image as input and follows the steps below to detect lanes:

- Convert the image to grayscale.
- Use gaussian blur to minimise noise and improve edge detection.
- Use Canny edge detection to identify edges.
- Create a region of interest (ROI) mask that focuses on the area where lanes are predicted.
- Use the mask on the edge-detected image.
- Use the Hough Line Transform to detect lines within the region of interest.
- Draw the identified lines on the original image.
- Return the image with the discovered lanes.

4.1 Testing

Figure-4: Output Image 1 of video 1

Figure-5: Output Image 2 of video 1

Figure-6: Output Image of video 2

Figure-7: Output Image of video 3

5. RESULTS AND DISCUSSIONS

Our deep learning models have demonstrated exceptional recall rates and precision in lane marking detection. This accomplishment is essential to guaranteeing the dependability and safety of autonomous vehicles on the road. Moreover, our algorithms have proven to be resilient in a variety of road conditions, such as busy intersections, highways, and city streets. Since road conditions can vary greatly in the real world, this adaptability is crucial for the deployment of autonomous vehicles. Another key aspect of our project is to focus on addressing challenges posed by adverse weather conditions, such as rain and snow. Future research could focus on improving these algorithms to improve adaptability to adverse weather conditions and incorporating them into autonomous driving systems for real-world applications.

6. CONCLUSION

The project "Driving Automation Using Lane Detection Algorithms," driven by Python, NumPy, and OpenCV, has made significant strides toward advancing the field of autonomous driving. Our work has addressed critical challenges related to road safety, complex environments, adverse weather conditions, and real-time processing. Through the development and implementation of state-of-the-art lane detection algorithms, we have laid the foundation for safer and more reliable autonomous vehicles.

Our achievements include the design and training of deep learning models capable of accurately identifying and tracking lanes in diverse scenarios. The integration of sensor data from cameras and other sources has further improved the robustness and redundancy of our lane detection system. Moreover, our real-time processing capabilities ensure timely lane information for swift and precise decision-making by autonomous vehicles. Our algorithms have demonstrated promising results in maintaining lane detection performance even in low-visibility scenarios.

7. FUTURE SCOPE

To implement the system, start by setting up a camera in the front of the vehicle to record the road ahead. Activate the camera using a manual switch or an automated system connected to the ignition of the vehicle to turn on the camera. Use OpenCV or a similar library to process the video stream from the camera. Apply your lane detection algorithm to each frame to detect lane markings on the road. Additionally, we can use the YOLO (You Only Look Once) algorithm to detect objects within the lanes, providing bounding boxes and class labels for each object. Combine these procedures into a single pipeline to process every video stream frame in real time, making sure that processing is done effectively to satisfy the car's computational limitations.

REFERENCES

[1] Y. Lei, S. L. Phung, A. Bouzerdoum, H. Thanh Le and K. Luu, "Pedestrian Lane Detection for Assistive Navigation of Vision-Impaired People: Survey and Experimental Evaluation," in *IEEE Access,* vol. 10, pp. 101071-101089, 2022.

[2] R. Izquierdo et al., "Vehicle Lane Change Prediction on Highways Using Efficient Environment Representation and Deep Learning," in IEEE Access, vol. 9, pp. 119454-119465, 2021.

[3] S. Li, X. Wu and Z. Wu, "Efficient Multi-Lane Detection Based on Large-Kernel Convolution and Location," in IEEE Access, vol. 11, pp. 58125-58135, 2023.

[4] N. J. Zakaria, M. I. Shapiai, R. A. Ghani, M. N. M. Yassin, M. Z. Ibrahim and N. Wahid, "Lane Detection in Autonomous Vehicles: A Systematic Review," in *IEEE Access*, vol. 11, pp. 3729-3765, 2023

[5] S. Sultana, B. Ahmed, M. Paul, M. R. Islam and S. Ahmad, "Vision-Based Robust Lane Detection and Tracking in Challenging Conditions," in IEEE Access, vol. 11, pp. 67938-67955, 2023.