Lane Detection based on edge detection and Hough

Transform feature recognition algorithm.

Abdelrahman Safwat

194637

Ahmad Essameldin   
192687

Evon Samy  
191365  
Omar Ashraf   
190265

Ziad Hegazi  
195407

*Abstract*— Images can be processed to extract various pieces of information from them. One of the processes that are usually applied to images is the process of edge detection or extraction, this combined with the use of the Hough transform for feature recognition constitutes the road lane detection algorithm which will be created in this project. Lane detection is the building block of many modern technologies in the automotive field such as lane keeping assist and lane departure warning. This paper will discuss the steps taken in order to develop a lane detection algorithm using traditional image processing techniques, namely edge detection and the Hough Transform using MATLAB.

Keywords—edge, edge detection, Hough Transform, lane, image processing, MATLAB

# Introduction

Autonomous vehicles are the next big step in the transportation field due to their efﬁciency and safety beneﬁts. A self-driven car needs to be aware of its surroundings so it can navigate its way in streets with minimal human assistance. Driver and passenger safety systems are one of the rising topics in automotive development. Lane keeping and Steering assist are some of the systems that allow vehicles to detect lanes and help in keeping the car in an ideal position with respect to the road and assist in turning. lane marking detection using video sensors is among the essential functions of an intelligent vehicle because it has many applications of driver assistance as lane departure and lane keeping assistance. the aim of the project is to use image processing techniques to perform road lane detection.

# Literature Survey



Figure 1: IRAODs Dataset

Extracting edges from images is a common process for finding and highlighting features for use in various applications. One of these applications is the main topic of this paper; that is road lane detection. Many methods are used for lane detection applications, some of these methods include traditional image processing techniques such as the ones that will be used in this project using MATLAB, similar to that in [1]. Other techniques revolve around more modern technology such as computer vision and artificial intelligence.

[2] discusses machine learning approach to detecting road lanes using what is referred to as a convolution neural network (CNN), the results of which are excellent in terms of accuracy and efficiency. This is the approach used in autonomous vehicles due to the speed of its operation and its low failure rate.

[3] applies a deep learning-based approach to isolate road surfaces from other objects within the image frame. Semantic segmentation and a training data set was used. The output is an image that is color coded to recognize and isolate different object classes which includes people, vehicles, and of course, the road.

Another approach is implemented by [5]. The method is called a Recurrent Feature-Shift Aggregator (RESA) which uses a CNN but with the addition of features that allow for more accurate and consistent detection even when the lanes are partially or even completely invisible.

Finally, [6] conducted road lane detection using a conditional convolution framework (CondLaneNet), in addition to the implementation of a Recurrent Instance Module (RIM) which provides predictive properties. The results are a robust yet efficient lane detection system that is able to overcome topological issues such as lines with different densities or forked lines.

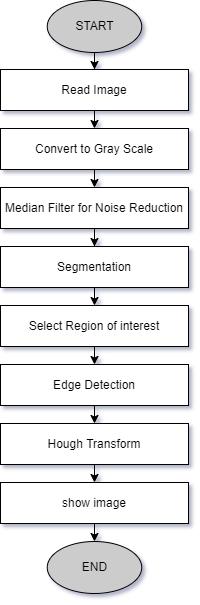
# Methodology

## Dataset Description

In order to compare the results of the algorithm with other papers, we needed to work with the same dataset that is iROADS. It’s separated into seven categories in different weather conditions; Daylight, Night, Rainy Day, Rainy Night, Snowy Day, Sun Stroke, and Tunnel. Each category includes JPEG Images of the road at the same weather conditions as shown in the figure.

## The used Algorithm

The flowchart of the algorithm is shown as follows:

The program starts by reading the image then convert that image into gray then median filter is used for noise reduction and then to binarize the image, we used global thresholding technique. Now we have a black and white image, and we need to select a region of interest, for this we drew a trapezoid and used the function poly2mask which transform the drawn trapezoid to a mask, that mask is then multiplied by the image. Then Sobel edge detection technique is applied on the masked image. After that we need to identify and draw lines inside the image for this, we used Hough transform, finally the lines are dawn on top of the original image.

## Comparison between the two algorithms:

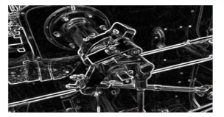
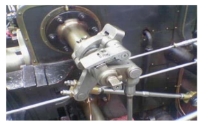
In this paper, **median filter was used** in order to reduce noise from the image; however,on the other hand, the compared paper is using **gaussian filter.**  The Median filter is a non-linear filter that is most commonly used as a simple way to reduce noise in an image. It's claimed to fame (over Gaussian for noise reduction) that it removes noise while keeping edges relatively sharp. **Median filter** was used because itpreserves edges and is not affected by the noise value while **gaussian filter** blurs edges and is affected by the noise value itself, and this project needs to preserve the edges to be able to detect rod lanes. The one advantage a Gaussian filter has over a median filter is that it's faster because multiplying and adding is probably faster than sorting. The second main change in this paper is the use of **Sobel edge detection technique** was used in order to extract the sharp edges from the image instead of the **canny edge detection technique**, used in the compared paper. a brief comparison between the two operators is shown in the following table:

Figure 2: Canny and Sobel difference

Canny

|  |  |  |
| --- | --- | --- |
| Parameter | Sobel | Canny |
| Computation | Simple and time consuming | Complex and time consuming |
| Number of objects in image | Suitable for simple images | Suitable for simple as well as complex images. |
| Strength | It is simple and detects edges. | Removes noise through smoothing effect. Good localization and response. |
| Weakness | Inaccurate and sensitive to noise. | Time consuming. |

Sobel



Each operator is useful in its own domain but the parameters like accuracy, complexity and efficiency creates a difference between these two. Basically, Sobel edge detector is used for high data transfer as Sobel gives fast performance results and its computation ability is also high. For this paper Sobel was used in order to accomplish faster computation time.

# Results

The output of each step in the algorithm is explained as follows:

## The Original Image

A car driving on a road

Description automatically generated with medium confidence

Figure 3:The Original Image

## Converting to gray

A car driving on a road

Description automatically generated with medium confidence

Figure 4: Converting to Gray

## Noise Reduction

A car driving on a road

Description automatically generated with medium confidence

Figure 5: Noise Reduction

## Global Thresholding

A car driving on a road

Description automatically generated with medium confidence

Figure 6: Global Thresholding

## Creat A mask

Shape

Description automatically generated

Figure 7:Create A mask

## Apply Mask

A picture containing dark, night sky

Description automatically generated

Figure 8: Apply Mask

## Sobel Edge Detction

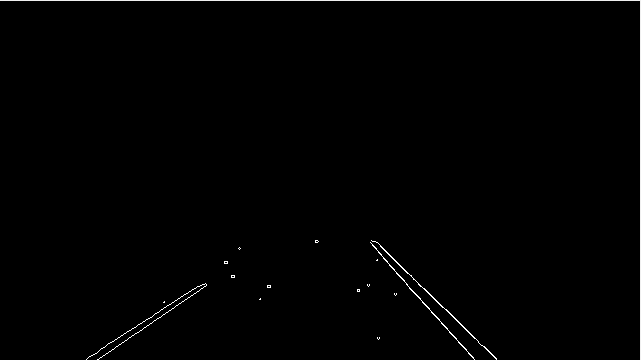


Figure 9: Sobel Edge Detection

## Hough Transform

A car driving on a road

Description automatically generated with low confidence

Figure 10: Hough Transform

# EValuation

In order to test the proposed algorithm, IROADS dataset is used which as mentioned consists of 7 collections with a total of 4700 images, in this paper only 3 categories were evaluated. The evaluation is simple, the results are divided into two groups, correct detection, and false detection groups. The correct detection includes the images that has two correct lines on both sides of the car; on the other hand, the false detection group includes the images with an incorrect identification of the lanes. However, how to decide whether the lane is identified correctly or not, here the ground truth of the IROADs dataset comes. This dataset includes an annotation file that includes labelling code for each image in this dataset. The annotation file is used as a ground truth and the decision was made based on this ground truth. Also, a python code was developed in order to make sure of that ground truth, the results were compared with both ways and the results were quite the same.



The proposed evaluation metric is the identification rate where the number of correct images is divided by the total number of images in this collection all times hundred to get a percentage.

The results are shown in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| IROADS Dataset | Total Number of frames | Number of false detection frames | Identification Rate % |
| Daylight | 903 | 51 | 94.3 |
| Tunnel | 307 | 19 | 93.75 |
| Rainy day | 1049 | 40 | 96.1 |

Comparison with the other algorithms:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithms | Daylight | Tunnel | Rainy day |
| Hu’s [4] | 96.23 | 94.46 | 94.75 |
| Fallah’s | 96.45 | 93.16 | 99.3 |
| Proposed | 94.3 | 93.75 | 96.1 |

As shown, the results of this algorithm did not exceed any of the compared papers; however, the results are acceptable. In order to justify that, this same algorithm was used in another image processing-based Lane keeping assist system project and the results are shown in the next Section.

# Implementation

As mentioned, this algorithm is used in another project which is, Computer-Vision Based Lane Keeping Mobile Robot. Basically, the project idea is to control the steering of the robot based on the measured parameters from the frame. The simulation and the actual results are shown in the following figures:

Graphical user interface, website

Description automatically generated

Figure 11: V-rep and MATLAB Simulation

Figure 12: Python Created Ground Truth

Shape

Description automatically generated

Figure 13: Applying Algorithm

A picture containing chart

Description automatically generated

Figure 14: Final Image

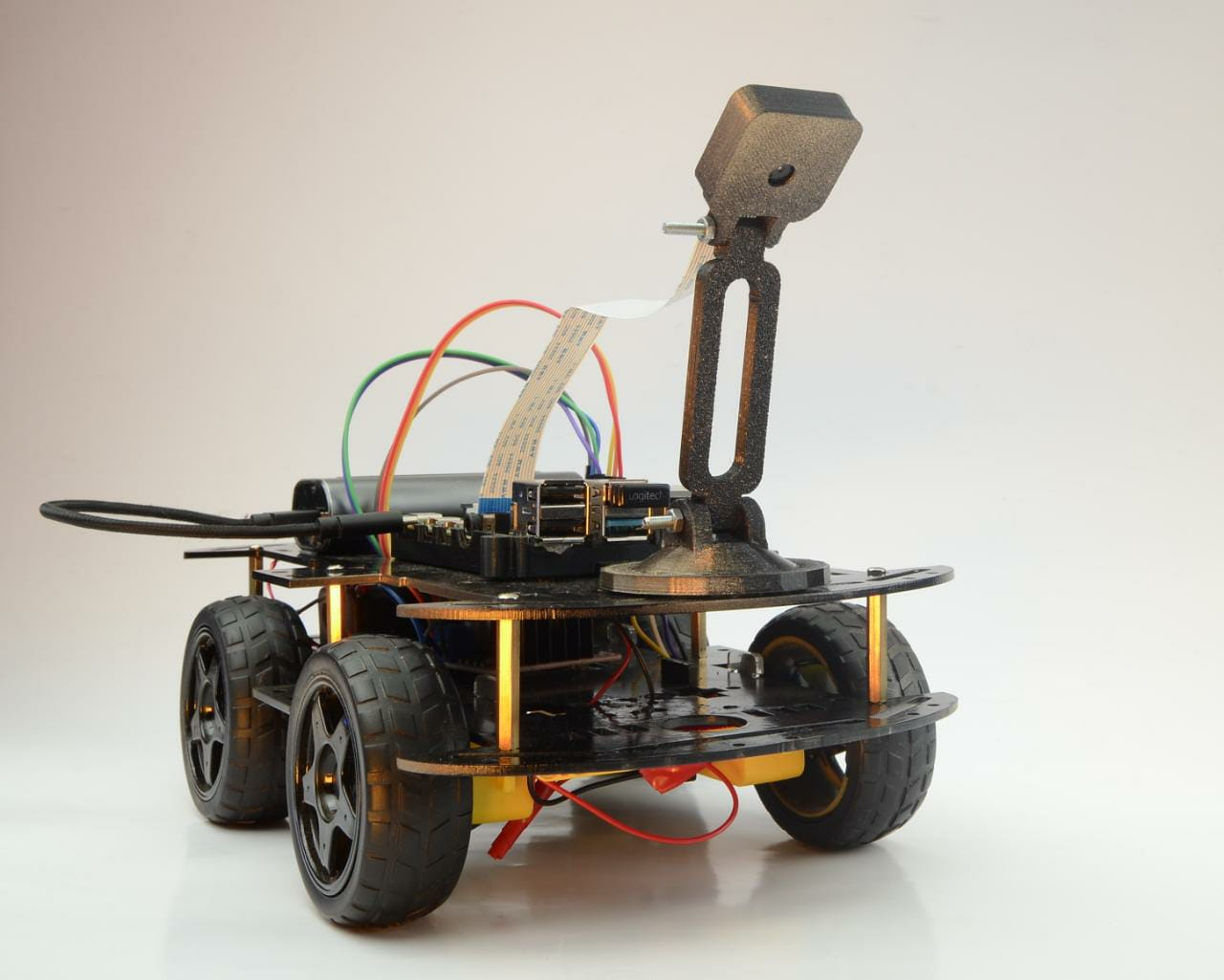


Figure 15: Hardware

Graphical user interface

Description automatically generated

Figure 16: Project Final Results

# Conclusion and future work

Automation is huge part of the upcoming future, and as technologies progress rapidly through time, it is clear that soon almost everything will be able to operate on its own including land vehicles. The project presented is a demonstration of how a vehicle would align itself in street lanes and curves by comparing the angle of deviation as well as the difference in distance between the center of the car and the center of the lane. With the use of computer vision and PID controller we built a prototype of the foundation of lane tracking system which is used in some applications and is being developed even more for further practices. Our next step is to modify the code to make it less time consuming as well as to enhance the results.

# References

1. Fallah, A., Soliemani, A., Khosravi, H. (2021). Real-time Lane Detection Based on Image Edge Feature and Hough Transform. Journal of Electrical and Computer Engineering Innovations (JECEI), 9(2), 193-202. doi: 10.22061/jecei.2021.7659.418
2. Satti, Satish Kumar & Devi, K. & Dhar, Prasenjit & Srinivasan, Padmanabhan. (2020). A machine learning approach for detecting and tracking road boundary lanes. ICT Express. 7. 10.1016/j.icte.2020.07.007.*)*
3. Goyal, A., Singh, M., & Srivastava, A. (2019). Lane Detection on Roads using Computer Vision. *International Journal of Engineering and Advanced Technology*.
4. Hu J, Xiong S, Sun Y, Zha J, Fu C. Research on Lane Detection Based on Global Search of Dynamic Region of Interest (DROI). *Applied Sciences*. 2020; 10(7):2543. https://doi.org/10.3390/app10072543
5. Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., & Cai, D. (2021). RESA: Recurrent Feature-Shift Aggregator for Lane Detection. AAAI.
6. Liu, L., Chen, X., Zhu, S., & Tan, P. (2021). CondLaneNet: A top-to-down lane detection framework based on conditional convolution. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv48922.2021.00375