order number: . . . . .

Thesis submitted to the

UNIVERSITY OF MOHAMED BOUDIAF – MSILA

FACULTY OF MATHMATICS AND COMPUTER SCIENCE

DEPARTEMENT OF COMPUTER SCIENCE

In partial fulfillment of the requirements for the degree of

Master in **Computer science**

By

FETAYAH, Ahmed

DJELLALI, Ali

Title of the thesis

Lane-line detection system

Under the supervision of

Hemza LOUCIF

Composition of the jury

Hemza LOUCIF	University of Msila	President
Abderrahim GUERNA	University of Msila	Reporter
Abdelouahab KHETTAF	University of Msila	Examiner

June, 2022

Dedications

In the name of God the merciful and the merciful praise to ALLAH the Almighty.

I dedicate this modest work as a sign of respect, recognition and thanks:

To my parents, Without their support and advice, my accomplishments would not have taken place, they have been behind me in each of my steps throughout my life, my deepest gratitude has been expressed to them, no words could describe the esteem that I have for them nor the good that they have done me, bring and give. To all my friends jamel, khaled, faical, abdelatif, elhachemi, Mohamed, nasserdine , idriss, imad and all those who are dear to me.

Ahmed,ali

Acknowledgments

First of all, I would like to thank allah for giving me the strength to survive and the courage to overcome all difficulties.

I would like to express my heartfelt thanks to my dissertation advisor, Mr. Loucif hemza. I thank him for framing me, guiding me, and helping me. For a long time, I have benefited from the knowledge and know-how that I was able to use in many discussions. I would also like to thank him for the autonomy and invaluable advice he gave me, enabling me to carry out this work. I would like to thank the jury of the faculty of Computer Science at University of mohamed boudiaf, where I express my heartfelt thanks for the kind evaluation of this work.

I thank all the speakers and all who have guided my thinking through their words, their contributions to this work, their writings, their advice and criticism, and agree to answer my questions in the course of my research question.

I could not have completed this project without expressing my sincere gratitude to all the professors in our college, their dedication and support throughout my education. I thank my very dear parents who have always been by my side: "You sacrificed everything for your children to be healthy and hardworking. You have given me a great example of hard work and perseverance. I owe me something to be proud of educate.

Finally, a special thanks to my family for their encouragement. I express my thanks, respect and gratitude to all these speakers.

Contents

List of figures

General introduction

With the rapid development of the global automobile industry, it has become one of the largest industries in the world, and the problem of road or lane perception is a crucial enabler for advanced driver assistance systems . At present, it has become the main industry in the national economic development of Germany, Japan, South Korea, United States and other countries.

The automobile industry has a strong role in promoting and leading the upgrading of industrial structure and the development of other related industries in each developed country. At the same time, many countries also appear with the contradiction between the rapid growth of automobile sales and the lack of basic road construction.

Some adverse phenomena such as automobile traffic accidents, energy waste, road congestion and environmental pollution have become more and more frequent and serious in many countries . According to the statistics, there are about 1.3 million people died in traffic accidents every year in the world .

Advanced driver assistance systems need Lane Line detection system. This system relies on a camera mounted on the front of the vehicle, which provides a visualization of the road environment, particularly the mark on the ground. Next, the boundaries or edges of the road are revealed in the images provided by the on-board camera. This detection makes it possible to estimate the lateral position of the vehicle in the lane.

The end of study project report for the Master degree in computer science in the M&C faculty of the university of M'sila consists of the following parts:

- chapter 1: we present the principle of lane line detection system and their phases.
- chapter 2 : we present a state of the art of lane detection.
- chapter 3: we discuss our approach adopted for lane detection.
- chapter 4: we present our developed application which detects lane line

Chapter1 : lane line detection

1. Introduction

Traffic safety remains one of the biggest challenges facing automakers today. The aim is to reduce the risk of accidents while maintaining the driver's driving comfort. However, vehicle collisions remain the leading cause of accidents resulting in fatalities and injuries in most crowded countries such as the United Kingdom, the United States, and Asian countries [1]. For this reason, automakers are integrating more and more driver aids into the design of new vehicles. Often referred to as ADAS (Advanced Driver Assistance Systems), these devices support drivers in their driving tasks.

ADAS is a system that identifies potentially dangerous situations and warns drivers to avoid accidents. One of the main technologies of the system is computer vision, a powerful tool for perceiving the road environment.

Computer vision has been widely used in intelligent transport applications such as obstacle detection, and lane departure warning [2]. Figure 1 illustrates these two ADAS. The first device detects the vehicles present in the driver's environment and keeps the latter informed in order to be able to avoid any collision. As for the second device, it detects the driving lane in order to maintain an ideal lateral trajectory for the vehicle and signals, to the driver, any involuntary lane departure.

This second type of ADAS perceives the road scene via an on-board camera on the front of the vehicle. Subsequently, the camera data is processed in order to detect the limits of the roadway. If the vehicle deviates from its ideal trajectory, the system warns the driver [3].

Figure 1:*Vehicle detection and lane detection systems*

2. Lane line detection

Marking lines on the ground are a safety device put in place by road managers to avoid leaving the road and frontal collisions. The system detects by image processing the lines of markings on the ground which delimit the rolling lane (the edges of the lane)

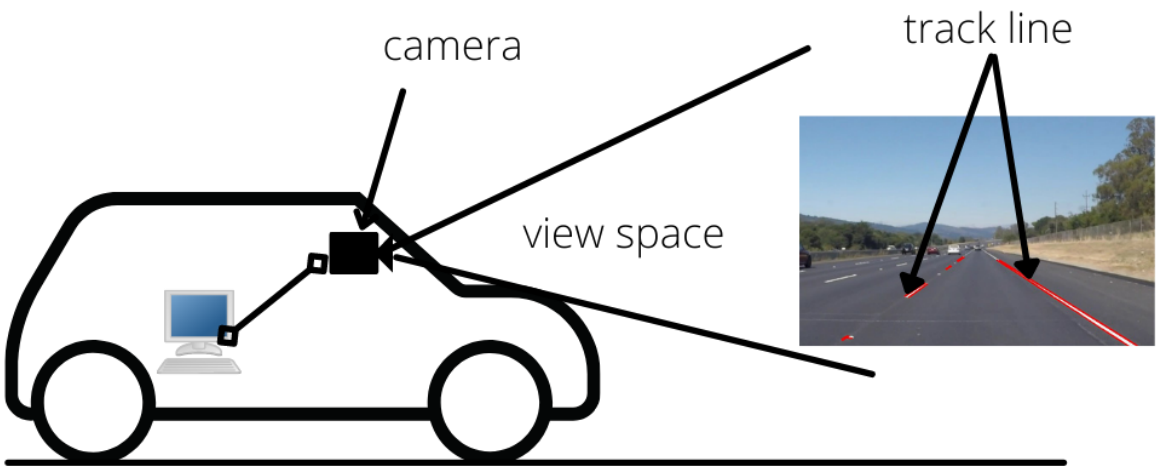

Figure 2: lane line detection

Lane detection requires three operations to perform: acquisition, lane detection and output image .

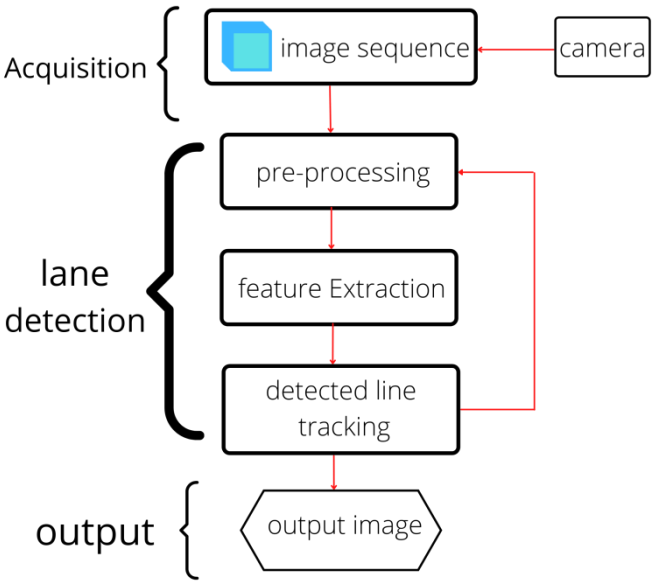

Figure 3:lane line detection architecture

2.1. Acquisition

The camera is installed at the front of the vehicle to perceive the road environment. The acquisition process provides a stream of images processed through the road detection process.

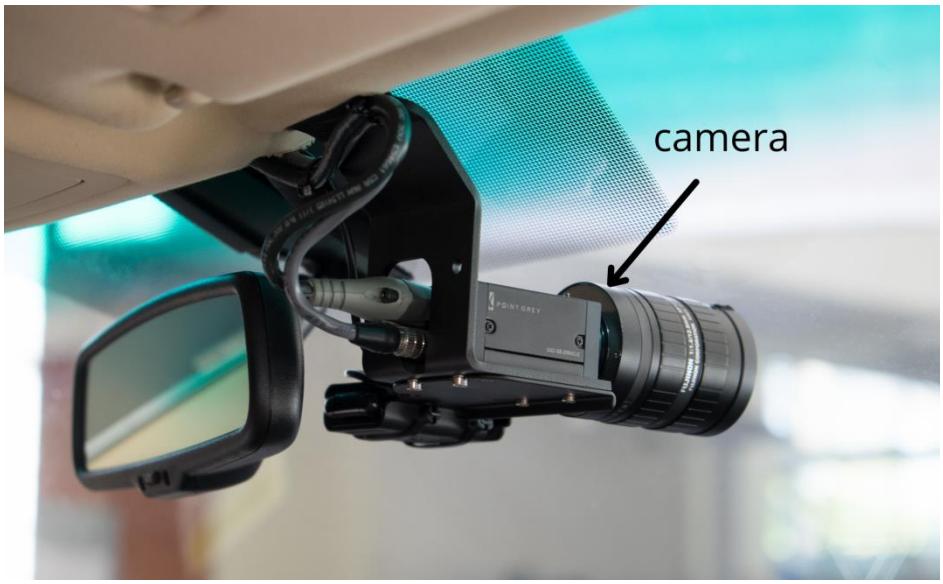

Figure 4:Front camera on board a vehicle

2.2. Lane detection

lane detection is based on three fundamental steps: pre-processing ,feature extraction and detection line tracking.

Different algorithms have been proposed for lane detection and each of them uses specific parameters [3]:

- Type of marking lines (continuous, discontinuous).
- Track model (linear, parabolic).
- Camera type (color, monochrome).
- Different methods (Hough transform, neural network, edge detection, segmentation).

2.3. Output image

In the process, the input images are displayed and marked with lines of different colors to determine the route and the cars on that road

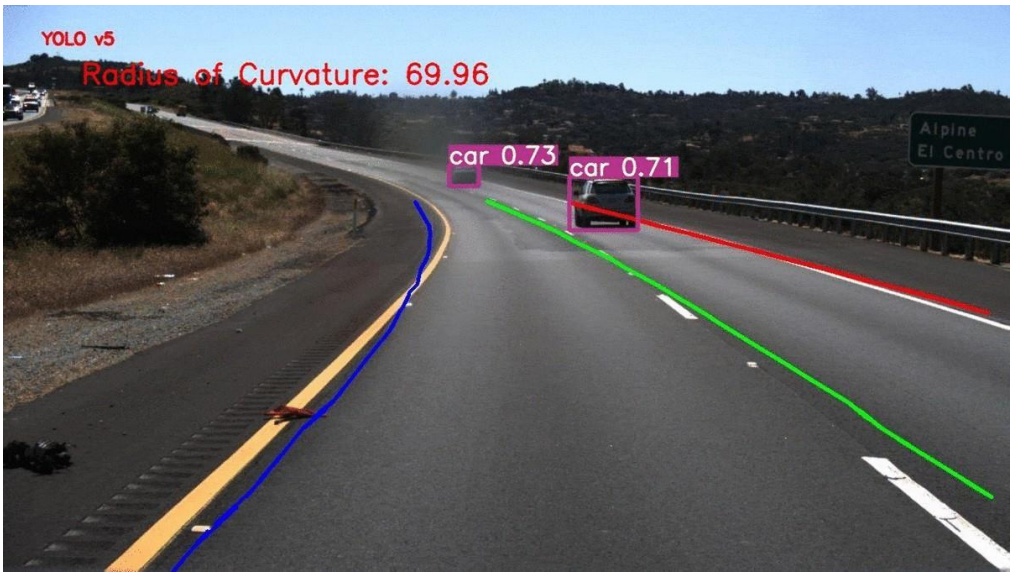

Figure 5: *Example of Output image*

3. Constraints of algorithms lane line detection

3.1. Lighting variability

Strong contrast fluctuations often interfere with the perception of the road, for example due to bridge shadows, trees or wet parts of the road. The wide variation in lighting conditions makes it difficult to determine the geometry of the runway. To be robust to these illumination changes, the image must always be processed for local contrast changes

Figure 6: Lighting variability in images

3.2.Type of marking

The type of marker affects the detection method. For example, discontinuous or worn marks make the task finer than continuous marks.

Other markings other than the edge of the road, Example. Lowering the arrow will result in false detection.

Figure 7: Type of marking in images

3.3. Presence of other vehicles

When overtaking at a crime scene, the vehicle obscures part of the markings, creating an informational gap that the algorithm must overcome.

Figure 8:Presence of other vehicles

3.4. Real time constraint

Visual lane detection and tracking methods require measurements at a minimum rate of ten frames per second to obtain satisfactory results. Therefore, each image must always be processed before the next image is captured, which requires the use of algorithms that are not computationally expensive

4. Lane detection methods

4.1. Feature based computer vision

In this approach possible lane marking candidates in an image are fitted to a model representing lane boundaries. Lane marking candidates are often extracted from the image by using color filters and edge detectors like Canny and Sobel. The shape of the lane estimated from possible lane marking candidates is represented by curves. Their parameters are estimated by using methods like Random Sample Consensus (RANSAC). In order to simplify this task, the image is projected to a bird's eye view using inverse perspective mapping. For that, it is necessary to know the position of the point in an image where parallel lines leading towards horizon are appearing to converge, also known as a vanishing point. Its position depends on the pose of the camera with respect to the road

Borkar et al. [4] proposed a model-based lane detection algorithm which uses Hough transformation and RANSAC. It was evaluated on videos recorded on highways with moderate traffic conditions and the method achieved 99.08% detection accuracy.

However, lane boundaries in urban environments are not always painted on the road. Very often they are made of curbs or just borders between different surfaces, for example, asphalt and grass. These algorithms do not perform well in such scenarios[5].

4.2. Probability-based computer vision

Another approach [6] implemented in TUeLaneTracker [7] does not binary size image like the previous method. It uses a Sobel operator to compute pixel gradients and calculates scores for each pixel depending on gradient magnitude and grayscale intensity. Each pixel has a gradient direction. Histograms of pixel scores intersecting base and purview lines are computed, as shown in Figure 1. The bin widths of the histograms are defined in metric unit.

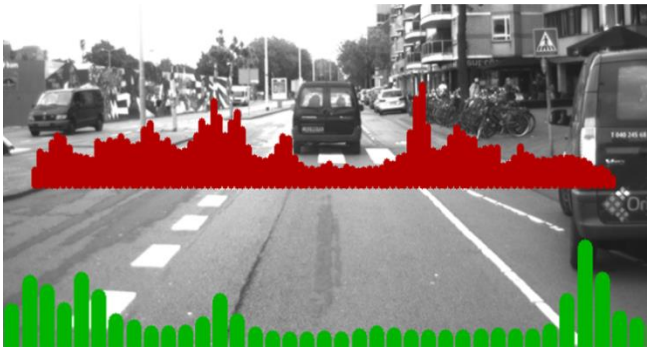

Figure 9: Histograms of baseline (green) and purview line (red) in TUeLaneTracker

Ego-lane representation consists of four parameters: the vertical and horizontal position of the vanishing point and the position of the left and right ego-lane boundary. The method tracks the vanishing point. The score of each pixel is adjusted depending on the difference between its orientation and orientation of the line connecting it with the vanishing point[5].

4.3. Deep learning algorithms

Deep learning methods are able to automatically find features in the dataset and learn how to calculate the output after they are trained on a dataset which is large

enough. Convolutional Neural Networks are very often used for computer vision problems. They have a hierarchical structure and layers containing convolution filters. First layers are detecting simple features like shapes and colors Consecutive layers are learning higher-level features, depending on training data.

Wang et al. [8] proposed a deep learning-based lane detection algorithm which uses two networks. The first network is used for pixel-wise lane edge classification and the second network detects lanes based on lane edge proposals. The paper does not show performance on roads with curbs but ensures that the neural network can easily adapt to any type of lane boundaries.

The dataset from TuSimple for Lane Detection Challenge [9], due to a large amount of annotated lane boundaries, allows comparing different architectures of neural networks for lane detection.

The method which performed the best on the TuSimple Benchmark Lane Detection Challenge [10] exploits the most important difference between semantic segmentation and lane detection: even with partially occluded lane markings, the driver has to infer position of the rest which usually is not the case with semantic segmentation where each pixel is assigned to one class (pixel with road occluded by car should be classified as car). It proposes the Spatial Convolutional Neural Network, which effectively propagates information between neurons on the same layer which helps to estimate the position of occluded lane markings. The method creates a probability map which is fitted to cubic splines representing lane boundaries[5].

5. The objective of the project

Our purpose in this paper is to investigate the proposed algorithm for detecting roads by vision. Then we implement one of the algorithms and test it on a series of images.

6. Conclusion

In this chapter, we introduce the general architecture of a lane line detection system. The latter is based on three processes: acquisition, lane detection and output image. The main process is lane detection, which requires modeling the latter to better localize vehicles based on lane markings.

The next chapter presents the state-of-the-art in lane line detection algorithm.

Chapter2 state of the art of lane detection

1. Introduction

Lane detection is a well-studied area of computer vision with applications in both autonomous vehicles and advanced driver assistance systems. Part of the reason is that while white markings on dark roads look simple, it can be difficult to identify markings on different road types. These difficulties include occlusions in the shadows of other vehicles, changes in the road surface itself, and different types of road markings. Lane detection systems must collect and filter all types of miscellaneous road markings to reliably estimate the path of the vehicle's position. Lane detection plays an important role in driver assistance systems. In general, the lane detection step locates the lane boundaries in the image of the specified path and can help estimate the geometry of the ground and the lateral position of the own vehicle on the road. Lane detection in an intelligent cruise control environment is used for lane departure warning, modeling path [11].

2. Lane Detection Algorithm (LDA) Overview

Lane detection algorithms identify lane markings and lane edges and estimate the vehicle's position in the lane. Lane detection provides a framework to support many single-camera based Mobile Eye features in addition to vehicle detection. In this case, it helps to correctly position the vehicle in the same lane. Provided that road markings are visible and their meaning is not limited by debris, such as B. Shadows, rain, snow or other disturbances on the road are obstructed. LDA detects most white, blue and yellow markers around the world, with the Mobil-Eye system accounting for about 99% of cases[11].

Different types of markings such as Solid line, dotted line, bot point are double and triple road markings that have been proven and successfully integrated into production. Additionally, LDA identifies unmarked curbs (roadsides), such as grass or gravel banks, to obtain more information about adjacent routes to support cautious strategies. In addition, a system has been developed that allows for better differentiation of ambiguous markers, double markers, triple markers, markers, etc. The system has been refined and adjusted to properly reflect the differences found in

different countries. Authorization mechanisms can also use color information for better separation[11].

LDA has been tested in series production projects in Europe, North America, Africa, the Middle East and Asia, and has been proven on multiple continents and in a variety of scenarios, including bright sunlight and weather around the world. This system is not suitable for construction areas with many overlapping marks. Different colored lane markings (such as Korean blue markings) have been successfully developed and drive a monochrome imager on the same input as all other functions.

Mobil Eye is currently developing a rear LDA whose unit is already in production (rear camera) for recycling applications. Overall system performance is improved in tunnel entry or driver assistance situations, example front cameras for sun visors and reflectors caused by road seams and tar[11]..

Figure 10:lane line detection

2.1. Feature of lane marking

- Adapts to various types of roads.

- Color , style and width of markings recognition.
- Detects all road markings in the picture.
- Integrated navigation system, see the track ego lane change offer advice.
- Adapts, to different weather and light condition[12].

2.2. Types of Road Lines

- **Continuous Centre lines**

You can cross a continue Centre line to enter or leave a road, but cannot overtake.

- **Broken center lines**

You are allowed to overtake across a broken Centre line or broken center line.

- **Continuous edges lines**

Boundary lines(edges lines) are used to select the edge of the road .The area to Left edge of the lines is the axis of the road which is also called shoulder of the road. This is not just an extra lane for vehicles to travel in. But , cyclists may also travel on the shoulder road, stopping at the side of a road, turning at an intersection etc. [13]

3. Proposed lane detection algorithm[14]

3.1. Region of Interest (ROI)

The first step in the proposed algorithm is taking the part of the image that contains the information necessary to detect the lanes accurately. The image of the road is divided into road region and non-road region. The role of this stage is to take only the part of the image where the road lanes exist the road region part of the image. In the proposed algorithm, we use a fixed ROI which has two benefits:

1. Focus the attention only on a subregion of the input image, which helps in reducing the runtime considerably.

2. It is the most suitable for the real-time application.

3.2. Inverse Perspective Mapping (IPM)

The second step in proposed algorithm is to generate a top view of the road image. This will get rid of the perspective effect in the image, and so lanes that appear to converge at the horizon line are now vertical and parallel. This uses the main assumption in this paper that the lanes are parallel (or close to parallel) to the camera. To get the IPM of the input image, we assume a flat road, and use the camera focal length, optical center, pitch angle, yaw angle, and height above ground parameters to perform this transformation. Starting by defining a world frame F_w as $\{F_w\} = \{x_w,y_w,z_w\}$ which centered at the camera optical center, a camera frame $\{F_c\} = \{x_c,y_c,z_c\}$, and an image frame $\{F_i\} = \{u,v\}$ as

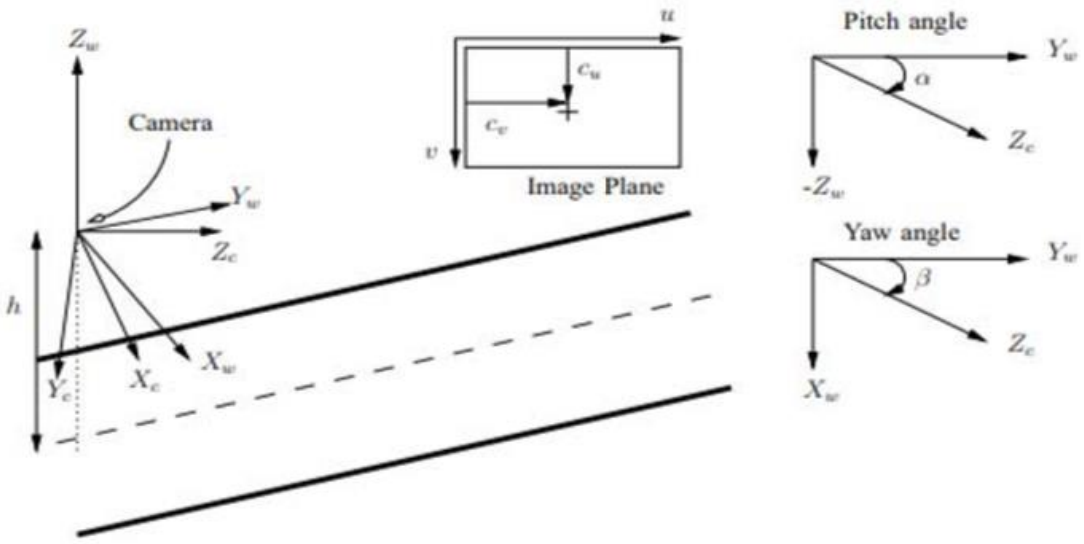

Figure 11:IPM coordinates. Left: the coordinate axes (world, camera, and image frames). Right: definition of pitch α and yaw β angles.

We assume that the camera frame X_c axis stays in the world frame X_w $Y_w$plane i.e. we allow for a pitch angle α and yaw angle β for the optical axis but no roll. The height of the camera frame above the ground plane is h. starting from any point in the image plane $p^i=\{u,v,1,1\}$, it can be shown that its projection on the road plane can be found by applying the homogeneous transformation as formula (1).

$$ {}^{g}_{i}T = h \begin{bmatrix} \frac{-1}{f_u} c_2 & \frac{1}{f_v} s_1 s_2 & \frac{1}{f_u} c_u c_2 - \frac{1}{f_v} c_v s_1 s_2 - c_1 s_2 & 0 \\ \frac{1}{f_u} s_2 & \frac{1}{f_v} s_1 c_1 & -\frac{1}{f_u} c_u s_2 - \frac{1}{f_v} c_u s_1 c_2 - c_1 s_2 & 0 \\ 0 & \frac{1}{f_u} c_1 & -\frac{1}{f_v} c_v c_1 - s_1 & 0 \\ 0 & \frac{1}{h f_u} c_1 & \frac{1}{h f_v} c_v c_1 - \frac{1}{h} s_1 & 0 \end{bmatrix} \quad (1) $$

When $p^g = {}^{g}_{i}T^{i}p$ it will be the point on the ground plane corresponding to p^i on the image plane, where $\{f_u, f_v\}$ are the horizontal and vertical focal lengths, respectively, $\{c_u, c_v\}$ are the coordinates of the optical center, and $c_1 = \cos\alpha$, $c_2 = \cos\beta$, $s_1 = \sin\alpha$, and $s_2 = \sin\beta$. These transformations can be efficiently calculated in matrix form for hundreds of points. The inverse of the transform can be easily found using formula (2).

$$ {}^{i}_{g}T = \begin{bmatrix} f_u c_2 + c_u c_1 s_2 & c_u c_1 c_2 + s_2 f_u & -c_u s_1 & 0 \\ s_2 (c_v c_1 - f_v s_1) & c_2 (c_u c_1 - f_v s_1) & -f_v c_1 - c_v s_1 & 0 \\ c_1 s_2 & c_1 c_2 & -s_1 & 0 \\ c_1 s_2 & c_1 c_2 & -s_1 & 0 \end{bmatrix} \quad (2) $$

Where starting again from a point on the ground $p^g = \{x_g, y_g, -h, 1\}$, subpixel coordinates can be found on the image frame by $p^i = {}^{i}_{g}T^{g}p$ and then rescale the homogeneous part. Using these two transformations, we can project a window of interest from the input image onto the ground plane. Figure.11 shows a sample IPM image. The left side shows the original image with the region of interest in yellow, and the right image how the transformed IPM image.

Figure 12:IPM sample. Left: input image with region of interest in yellow. Right: the IPM view

3.3. Line Segment Detection (LSD)

LSD is an algorithm used for detection of line segments in a photo and it stands on Helmholtz principle which states that no detection is perceived in noise which can be applied in the computer vision field as any object desired to be detected shall be combined with a geometrical event and its regularity is validated in the Contrario model in case of line segment this event is rectangle as line segment can be viewed as rectangle containing pixels with various orientations [9]. Some definitions should be known first like level line field which is a vector calculated to all the pixels of the image and its magnitude is the gradient of the pixel and its direction is from the dark side to the lighter side on the edge of transition from one pixel to another and its orientation is defined by an angle called the level line angle (LLA). Contrario model in this algorithm is defined as a photo with same dimensions of the original one where level line fields of pixels are defined as independent random variables and their level line angles have uniform distribution from 0 to 2π [10]. Steps of the algorithm are discussed as follows:

3.3.1. Image Scaling

Details in image perceived by the human eye differ when the image is seen at different scales and this happens exactly in the algorithm as line segments detected differ when they are validated at different scales which can cause what is known as staircase effect at which some line segments are falsely detected or not detected at all due to image scale as shown below in Figure. 12 so image scaling is the first stage of the algorithm and the scale used for image is 80% as this is the smallest which overcomes the staircase effect while also preserving most details of the photo. Image scaling is done using Gaussian subsampling after filtering the high frequencies in the image to avoid aliasing [9].

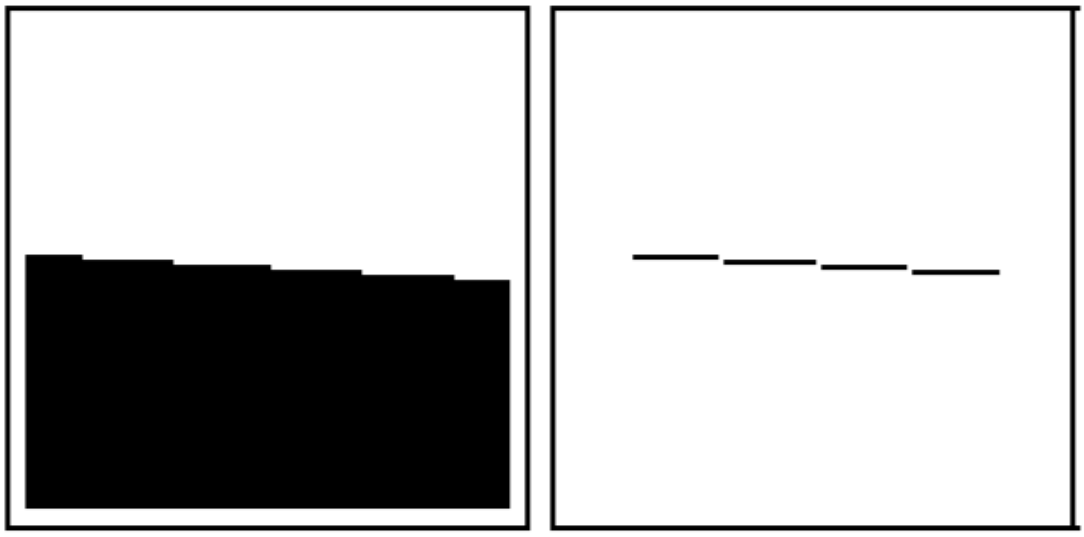

Figure 13: Image before scale where line segment is detected as four small line segments

Figure 14: Line segment detected correctly after using 80% scaling

3.3.2. Gradient Computation

The second step is gradient computation which is done by 2x2 mask which calculates the difference in intensities between a pixel and the neighboring pixels to it and we use 2x2 mask to reduce the dependencies among the level line fields magnitudes of the neighboring pixels to keep faithful to our assumption in the Contrario model that they are independent random variables. And pixel level line calculated from (5).

$$gx = \frac{i(x+1,y)+i(x+1,y+1)-i(x,y)-i(x,y+1)}{2} \qquad (3)$$

$$gy = \frac{i(x,y+1)+i(x+1,y+1)-i(x,y)-i(x+1,y)}{2} \qquad (4)$$

Level line angle = $arctan\left(\frac{gx(x,y)}{-gy(x,y)}\right)$ (5)

Gradientmagnitude = $G(x,y) = \left(gx(x,y) + gy(x,y)\right)^{1/2}$ (6)

Then pixels are pseudo-ordered due to their gradient magnitudes calculated from (3), (4), (6) into quantized 1024 bins to optimize sorting in linear time and small gradient magnitudes are rejected because they introduce a large quantization error [9].

3.3.3. Region Growing and Rectangle Approximation

Region Growing is performed on ordered pixels where a seed pixel is chosen as the highest gradient magnitude unused pixel and if the absolute difference between its LLA and the neighboring ones LLA is within tolerance τ̄ it joins the region and the region angle is updated as the average angle of the region's pixels and the tolerance is a design parameter chosen by the designer and it shouldn't be too extreme and it was chosen in our implementation by the value π/8 . After that, the region is approximated to a rectangle with its length and width are chosen such that they are the smallest dimensions which cover the region and the rectangle orientation is calculated by interpreting the rectangle as a solid object and the gradient magnitudes of its pixels as weight of its points and the orientation of the rectangle is calculated as the direction of the first axis of inertia of the object [9].

3.3.4. Validation by Calculating NFA

Each rectangle is validated by calculating the probability that a rectangle can be present by the same regular structure or more in the Contrario model. the regularity of the structure of rectangle is defined by the number of aligned points in it and aligned points are defined as the points whose level line angle similar to the rectangle

orientation within certain tolerance so we can use binomial probability distribution to measure that

probability which leads us to an exhausting number of tests as (NM)5/2 as shown in the Figure. 14 where N and M are dimensions of photo after scaling.

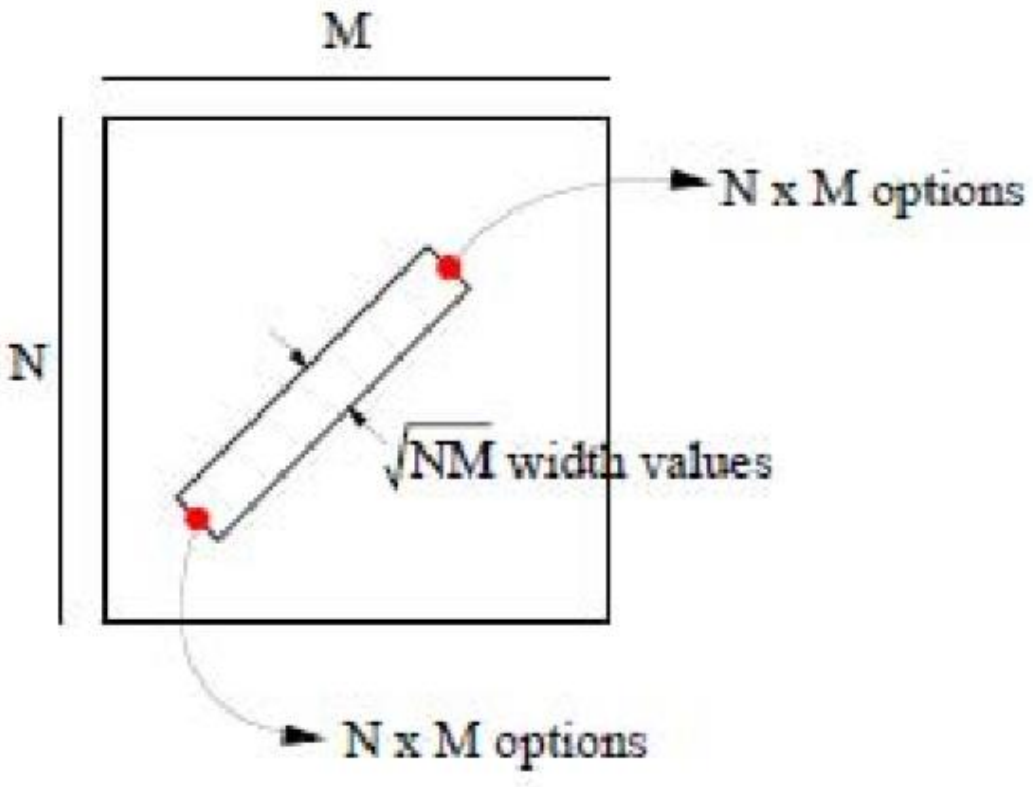

Figure 15: Number of tests determination

Probability of having rectangle with same regular structure or more regular in the Contrario model defined as NFA (number of false alarms) = ((NM)^(2.5))*$\sum_{k=j}^{n}\binom{n}{j} * p^{j} * p^{1-j}$ where p is the probability where LLA of pixel with in tolerance $\bar{\imath}$ and it is calculated as $\frac{\bar{\imath}}{\pi}$ and the resulted NFA is compared by threshold set to 1 a as mentioned before. Finally, the algorithm outputs the starting and ending points of line segments in the photo and it works with complexity proportional to the dimensions of the photo.

3.4. Hough transform

Hough transform is a feature extraction technique that aims to find imperfect instances of objects in a particular type of shape by voting program. The voting procedure is carried out in a parameter space in which candidates are obtained as local maxima in the so-called accumulator space, which is explicitly constructed by the

algorithm used to compute Hough transform. The basic Hough transform is to detect straight lines (segments) from black-and-white images. The main advantage of Hough transform is that it can tolerate the gap in feature boundary description and is relatively free from image noise. The simplest Hough transform is to detect straight lines. We know that the equation representation of a straight line can be expressed by slope and intercept (this representation is called oblique intercept), as follows:

$$y = mx + b$$

If expressed in parameter space as (b, m), a straight line can be represented by slope and intercept. However, there is a parameter problem. The slope of the vertical line does not exist (or is infinite), which makes the value of the slope parameter mm close to infinity. To this end, in order to better calculate:

$$r = x\cos\theta + y\sin\theta$$

Where R is the distance from the origin to the nearest point on a straight line (others may record it as rho, and below can also regard r as parameter rho), theta is the angle between the x-axis and the line connecting the origin and the nearest point. As shown in Figure 15[15].

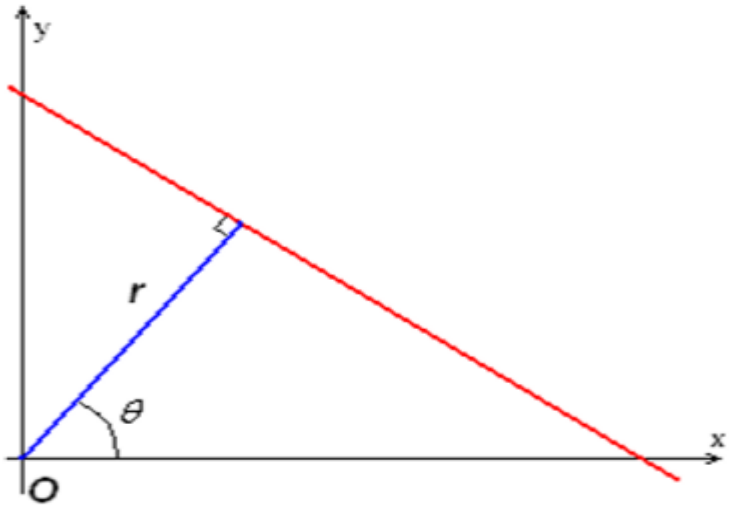

Figure 16: The relationship between the origin and angle of a straight line and the position of a straight line

Therefore, each line of the image can be correlated with a pair of parameters (r, θ). This parameter (r, θ) plane is sometimes called Hough space and is used for the set of two-dimensional straight lines.

3.5. Yolo v3 algorithm

YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images. YOLO uses features learned by a deep convolutional neural network to detect an object. Versions 1-3 of YOLO were created by Joseph Redmon and Ali Farhadi.

The first version of YOLO was created in 2016, and version 3, which is discussed extensively in this article, was made two years later in 2018. YOLOv3 is an improved version of YOLO and YOLOv2. YOLO is implemented using the Keras or Open CV deep learning libraries.

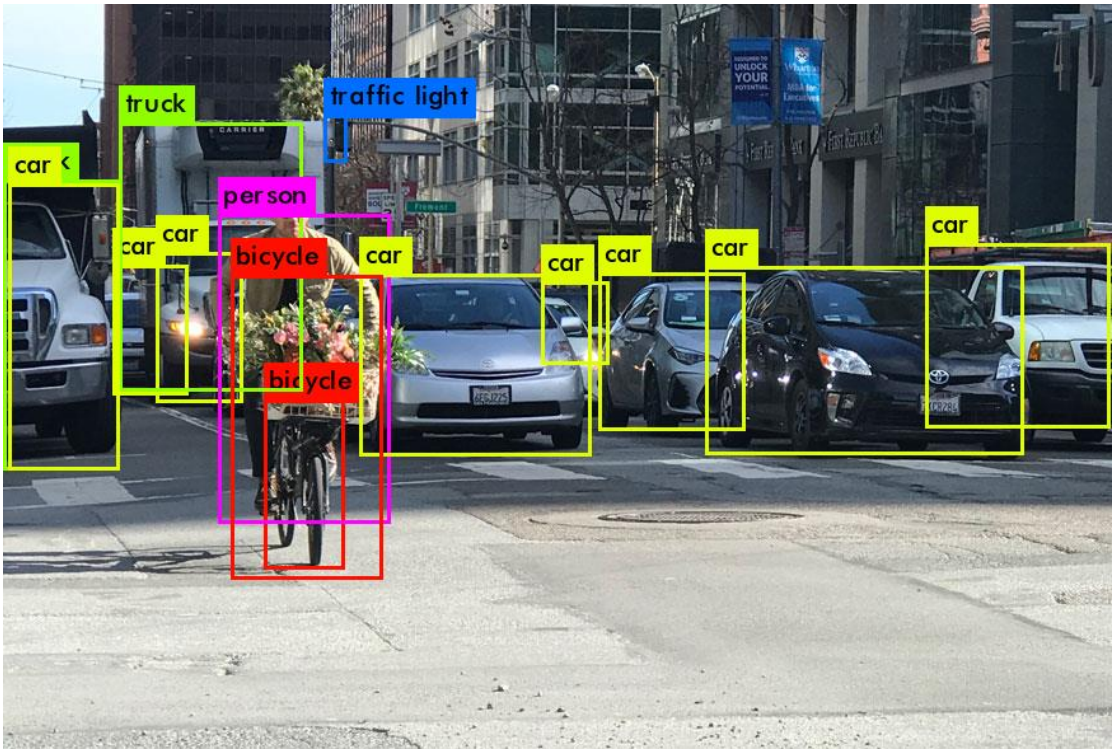

Figure 17: YOLOv3 Computer Vision Example

Object classification systems are used by Artificial Intelligence (AI) programs to perceive specific objects in a class as subjects of interest. The systems sort objects in images into groups where objects with similar characteristics are placed together, while others are neglected unless programmed to do otherwise.

3.6. Canny edge detector

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It

has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection include:

Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible

The edge point detected from the operator should accurately localize on the center of the edge.

A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Among the edge detection methods developed so far, Canny edge detection algorithm is one of the most strictly defined methods that provides good and reliable detection. Owing to its optimality to meet with the three criteria for edge detection and the simplicity of process for implementation, it became one of the most popular algorithms for edge detection[16].

Figure 18: Canny edge detection applied to a photograph

4. Conclusion

In this chapter, we introduce and present some of the methods that have been proposed Identifying roads visually, these algorithms are still in development Because mathematics and computer technology in the field are constantly evolving.

In the next chapter, we will discuss the yolov3 method and the implementation of the canny edge detector for the Lane detection.

Chapter3 approach adopted for lane detection.

1. Introduction

In this chapter, we introduce the implementation part. We will describe the algorithm employed, its general architecture, and the development steps used to implement our method.

2. The algorithm methodology

Such intelligent lane detection systems require algorithms that satisfy good performance conditions. For this, we will discuss our proven method. First we discuss image capture, image grayscale, region of interest, then we detail canny Edge method and yolov3 , then applying Hough transform, and finally removing unwanted lines according to predefined criteria.

3. Algorithms and method used

3.1. Lane line detection

3.1.1. Canny edge

Edge detection, particularly step edge detection is an important technique to extract structural information and considerably reduces the amount of data to be processed. John F. Canny in 1986 developed a Canny Edge detector[8], also known as optimal detector, Canny edge detector aims to satisfy the three general criteria of edge detection.

1. Low error rate- which means that it detects only existent edges.

2. Good localization- The difference between real edge pixels and the detected edge pixels have to be minimized.

3. Minimal response- the detected edge should marked only once not many times.

The steps of the Canny edge detection are as follows:

-Remove any noise using a filter. Gaussian filter[10] is used for removing the noise. An example of Gaussian kernel filter of size 5 is

$$k = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

-Find the gradient of the image

Gradient along x, and y directions are calculated using the convolution masks of 3x3 as follows.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Gradient strength and direction of the edges are calculated as follows:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = arctan\left(\frac{G_y}{G_y}\right)$$

Here G_x and G_y are gradients along x and y directions.

-Non-maximum suppression is performed. This step removes pixels which are not considered as part of an edge. Only thin lines will remain, these contain pixels which are considered to be part of an edge.

-Hysteresis- The final step, two thresholds are used named as upper and lower.

- If the gradient value of a pixel is higher than the upper threshold, then the pixel is considered as an edge pixel.

- If the gradient value of a pixel is less than the lower threshold, then the pixel is rejected.
- If the gradient value of the pixel is between the lower and upper thresholds, then the pixel will be accepted only if it is connected to a pixel that is above the upper threshold[17].

3.1.2. Hough transform

The Hough transform is a technique for isolating specific elements of a shape inside an image. The classical Hough transform is most typically employed for the detection of regular curves such as lines, circles, ellipses, and so on, because it requires the desired features to be provided in some parametric form. In situations where a straightforward analytic description of a feature (s) is not attainable, a generalized Hough transform can be used. We will limit our study to the traditional Hough transform due to the computational difficulty of the generalized Hough algorithm.

Despite its domain limitations, the classical Hough transform (hereafter referred to without the classical prefix) has a wide range of applications, as most manufactured parts (as well as many anatomical parts investigated in medical imagery) have feature boundaries that can be described by regular curves. The fundamental advantage of the Hough transform technique is that it is unaffected by picture noise and is tolerant of gaps in feature boundary descriptions.

How it work

The Hough technique is particularly useful for computing a global description of a feature(s) (where the number of solution classes need not be known a priori), given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (Example. coordinate point) indicates its contribution to a globally consistent solution (Example. the physical line which gave rise to that image point).

As a simple example, consider the common problem of fitting a set of line segments to a set of discrete image points (Example. pixel locations output from an edge detector). Figure 18 shows some possible solutions to this problem. Here the lack of a priori knowledge about the number of desired line segments (and the

ambiguity about what constitutes a line segment) render this problem under-constrained.

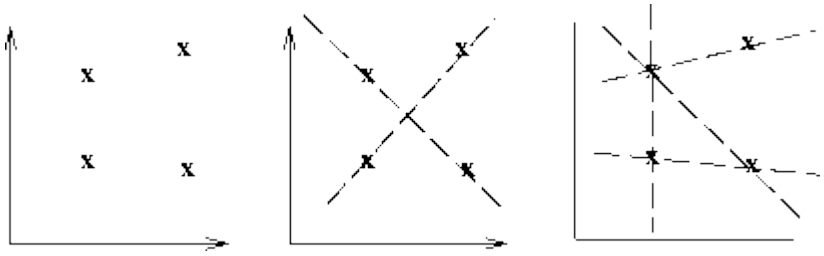

Figure 19: a) Coordinate points. b) and c) Possible straight line fittings.

We can analytically describe a line segment in a number of forms. However, a convenient equation for describing a set of lines uses parametric or normal notion:

$$x\cos\theta + y\sin\theta = r$$

Where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-axis. (See Figure 19.) For any point (x, y) on this line, r and θ are constant.

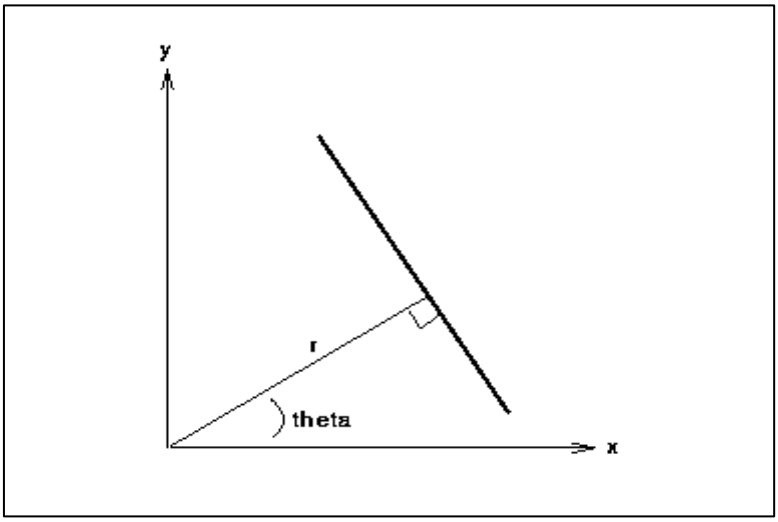

Figure 20: Parametric description of a straight line.

In an image analysis context, the coordinates of the point(s) of edge segments (i.e. (x_i, y_i)) in the image are known and therefore serve as constants in the parametric line equation, while r and θ are the unknown variables we seek. If we plot the possible (r, θ) values defined by each (x_i, y_i) , points in cartesian image space map to curves (i.e. sinusoids) in the polar Hough parameter space. This point-to-curve transformation is the Hough transformation for straight lines. When viewed in Hough parameter space, points which are collinear in the cartesian image space

become readily apparent as they yield curves which intersect at a common (r, θ) point.

The transform is implemented by quantizing the Hough parameter space into finite intervals or accumulator cells. As the algorithm runs, each (x_i, y_i) is transformed into a discretized (r, θ) curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

We can use this same procedure to detect other features with analytical descriptions. For instance, in the case of circles, the parametric equation is

$$(x - a)^2 + (y - b)^2 = r^2$$

where a and b are the coordinates of the center of the circle and r is the radius. In this case, the computational complexity of the algorithm begins to increase as we now have three coordinates in the parameter space and a 3-D accumulator. (In general, the computation and the size of the accumulator array increase polynomially with the number of parameters. Thus, the basic Hough technique described here is only practical for simple curves.) [18]

3.2. object detection

3.2.1. yolov3

YOLO is a Deep Learning architecture proposed by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi in the paper 'You Only Look Once: Unified, Real-Time Object Detection'uses a totally different approach. It is a clever convolutional neural network (CNN) for object detection used in real-time.

Further, It is popular because it has a very high accuracy while also being able to run in real-time or used for real-time applications. The YOLO algorithm "only looks once" at the input image that is it needs only one forward propagation pass through the network to make the predictions

How does YOLO work

Prior detection systems use localizers or classifiers to carry out the detection process. Then the model is applied to an image at different scales and locations. The regions of the image with High scoring are considered for detections.

YOLO algorithm uses a completely different approach. The algorithm applies a single neural network to the entire full image. Then this network divides that image into regions which provides the bounding boxes and also predicts probabilities for each region. These generated bounding boxes are weighted by the predicted probabilities.

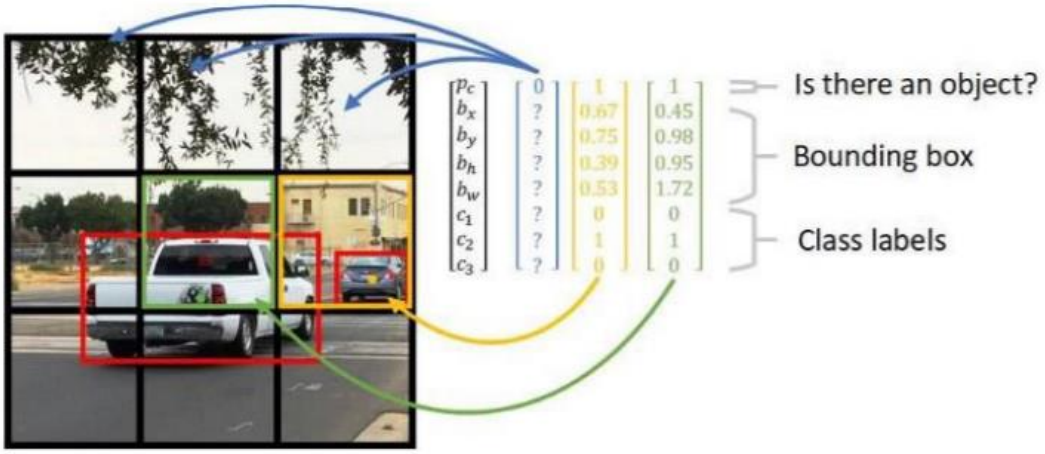

Figure 21:YOLO algorithm predicts probabilities for each region.

The non-max suppression technique makes sure that the object detection algorithm only detects each object once and it discards any false detections, it then gives out the recognized objects along with the bounding boxes.

Architecture

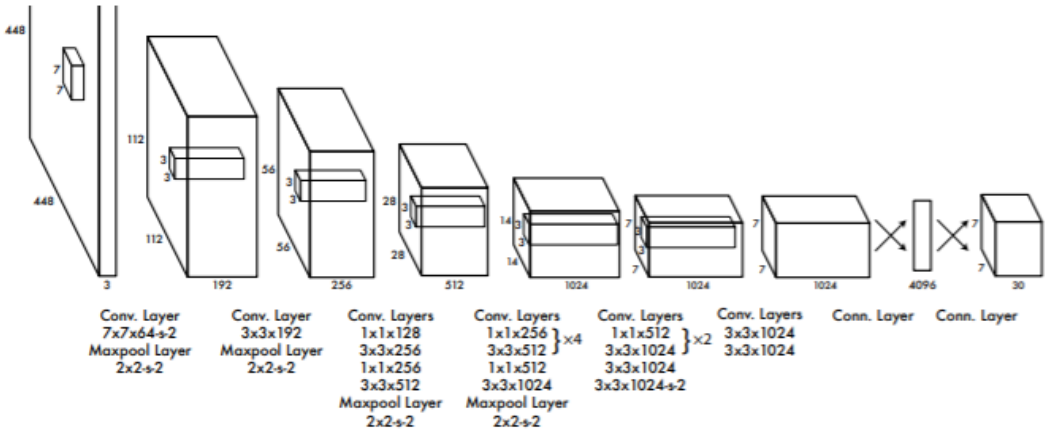

Figure 22:yolov3 architecture

YOLO is a convolution neural network. It consists of a total of 24 convolutional layers and followed by 2 fully connected layers. Each layer has its own importance and the layers are separated by their functionality.

-The First 20 convolutional layers followed by an average pooling layer and a fully connected layer is pre-trained on the ImageNet dataset which is a 1000-class classification dataset.

- The pretraining for classification is performed on the dataset with the image resolution of 224 x 224×3.

- The layers comprise 3×3 convolutional layers and1x1 reduction layers.

- For object detection, in the end, the last 4 convolutional layers followed by 2 fully connected layers are added to train the network.

- Object detection requires more precise detail hence the resolution of the dataset is increased to 448 x 448

- Then the final layer predicts the class probabilities and bounding boxes.

- All the other convolutional layers use leaky ReLU activation whereas the final layer uses a linear activation.

- The input is of 448 x 448 image and the output is the class prediction of the detected object enclosed in the bounding box[19]

4. Conclusion

In this chapter, we have given importance to the image processing methods necessary for our algorithm since they will be our subject of work, we have described the general architecture of the adopted algorithm, the area of interest, we have well detailed the methods (yolov3 and canny edge)

Chapter4: developed application which detects lane line

1. Introduction :

After presenting a state-of-the-art of channel detection methods and detailing our approach based on the canny edge and yolov3 transform, it's now time to put the previous chapters' steps into practice to create a "lane detection" application. This chapter focuses on the development of an application that can recognize road edges using a variety of approaches, technologies, and algorithms for image processing, as well as the tools that enabled this. Finally, we present the results of various tests performed on a given image sequence.

2. Hardware and software environment:

In this section, we will present the working environment to realize our system

Resources used

The physical resources used are:

- Intel® Core™ i7-2620M processor with a frequency of 2.70 GHz 2.70 GHz
- A RAM with a capacity of 8 GB
- An NVIDIA Quadro graphics card of 3000 MB

And on the software side :

- Operating System: Windows 10 Professional
- Microsoft Visual Studio 2022 Professional version 1.67.1
- Jupyter notebook version 6.4.11
- Open source Computer Vision: OpenCV version: 4 .5.5.64
- Numpy version 1.22.3
- Keras version 2.8.0
- Tensorflow version 2.8.0
- Matplotlib version 3.5.1

2.1. Programming language

The programming language chosen in our work is Python, which remains one of the most used languages in the field of pattern recognition. This is because the Python language has a high performance compiler to generate very fast code and it has high level structures and instructions.

Python is a high-level object-oriented programming language that was created by Guido van Rossum. It is also called general-purpose programming language as it is used in almost every domain we can think of as mentioned below [20]

- Web Development
- Software Development
- Game Development
- AI & ML
- Data Analytics

2.2. OpenCV version: 4 .5.5.64

OpenCV (Open Source Computer Vision Library) is a programming library geared mostly on real-time computer vision. It was originally developed by Intel, although Willow Garage and Itseez later supported it (which was later acquired by Intel). Under the open-source Apache 2 License, the library is cross-platform and free to use. OpenCV now has GPU acceleration for real-time operations since 2011.

Figure 23: OpenCV logo.

2.3.Numpy version 1.2.2.3

NumPy (pronounced "NUM-py" or "NUM-pee") is a Python library that adds support for huge, multi-dimensional arrays and matrices, as well as a vast number of high-level mathematical functions to manipulate these arrays. Numeric, NumPy's forerunner, was built by Jim Hugunin with help from a number of other people. Travis Oliphant built NumPy in 2005 by heavily modifying Numeric and combining features from the competitor Numarray. NumPy is an open-source project with a large number of contributors. NumPy is a financially supported NumFOCUS project.

Figure 24: NumPy logo.

2.4.Keras version 2.8.0

Keras is an open-source software library for artificial neural networks that includes a Python interface. Keras serves as a user interface for TensorFlow.

Until version 2.3, Keras supported a variety of backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML. As of version 2.4, only TensorFlow is supported. It is user-friendly, modular, and expandable, with the goal of allowing quick experimentation with deep neural networks. It was created as part of the ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) research project, and François Chollet, a Google engineer, is the principal author and maintainer. Chollet is also the creator of the deep neural network model XCeption

Figure 25: Keras logo.

2.5.Tensorflow version 2.8.0

TensorFlow is a machine learning and artificial intelligence software library that is free and open-source. It can be used for a variety of tasks, but it's best known for deep neural network training and inference.

The Google Brain team created TensorFlow for internal Google use in research and production. In 2015, the first version was released under the Apache License 2.0. In September 2019, Google launched TensorFlow 2.0, an improved version of TensorFlow.

TensorFlow is compatible with a wide range of programming languages, including Python, Javascript, C++, and Java. This adaptability lends itself to a wide range of applications in a variety of industries.

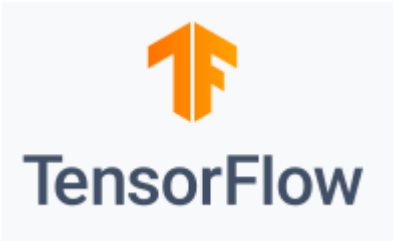

Figure 26: TensorFlow logo.

2.6.Matplotlib version 3.5.1

Matplotlib is a charting library for the Python programming language and NumPy, the language's numerical mathematics extension. It provides an object-oriented API for embedding charts into applications utilizing GUI toolkits such as Tkinter, wxPython, Qt, or GTK. There's also a procedural "pylab" interface built on a state machine (like OpenGL) that's meant to look like MATLAB, however it's not recommended. Matplotlib is used by SciPy.

John D. Hunter was the creator of Matplotlib. It's had a thriving development community since then, and it's distributed under a BSD-style license. Shortly before John Hunter's death in August 2012, Michael Droettboom was chosen as matplotlib's primary developer, and was later joined by Thomas Caswell. Matplotlib is a financially supported NumFOCUS project.

Python versions 2.7 to 3.10 are supported by Matplotlib 2.0.x. Matplotlib 1.2 was the first version to support Python 2.6, while Matplotlib 1.4 is the most recent version

to do so. By signing the Python 3 Statement, Matplotlib has vowed not to support Python 2 after 2020.

Figure 27: Matplotlib logo.

3. How does our application work

3.1. Input

The code opens a video file and reads the first frame , if it is true then the code writes that frame to a new file.

3.2. The process

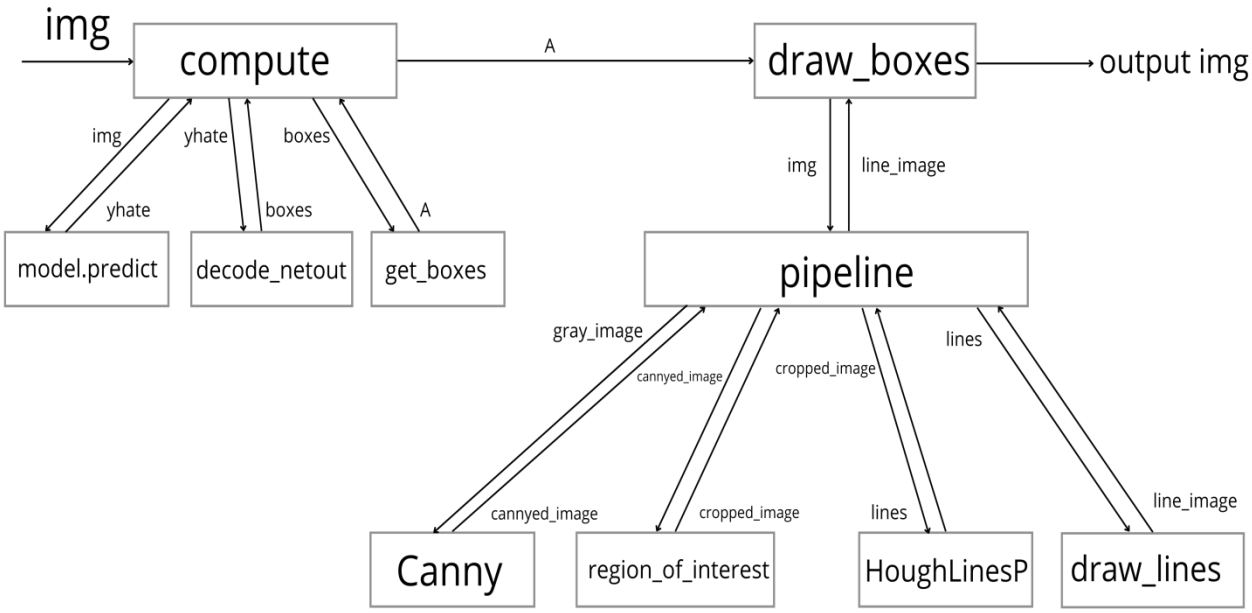

Figure 28: Diagram explaining how the code works

Information about the scheme

Figure 29: Scheme to understand the previous figure

3.3.Output

View images drawn after processing , than the program writes the processed frames into a new video

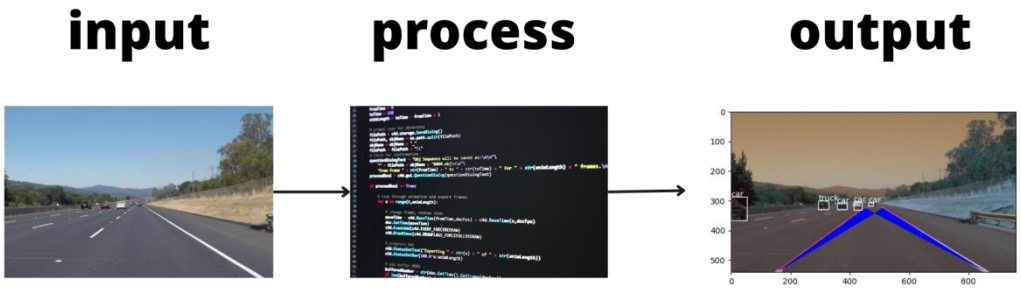

Figure 30: Input and output after processing

4. Conclusion

We have explained the stages of development of our lane detection system, as well as the hardware and software environment, in this chapter. Based on what has been done in it, we can say that our application meets some of the following features:

- Excellent performance.
- There are very few mistakes.

Conclusion

The lane detection project allowed us to experiment with various image processing approaches and gain a better understanding of lane detection's capabilities and limitations. The lane-detection system is now regarded as one of the most important driver aid gadgets. It has a wide range of applications, including road safety, intelligent so-called autonomous cars, robotics, and aeronautics for aircraft ground navigation, among others.

We were interested in the problem of lane detection in this project. This is why we offered state-of-the-art detection algorithms, which enabled us to develop our lane detector.

During our examination of detection methods, we discovered that there is a lot of research in the subject of intelligent vehicles. Indeed, they have spawned a plethora of solutions, ranging from the simple detection of lane markings and accurate vehicle placement to the recognition of road signs, pedestrian detection, vehicle detection, and so on. However, the methods offered thus far are not necessarily suitable in unstructured or uncontrolled contexts, so progress in this area will continue.

Bibliography

[1] Alexander BARGETON, "Fusion multi sources pour l'interprétation de l'environnement routière", thèse de doctorat, école national supérieure des mines de Paris (ParisTech), 2009.

[2] Shabana HABIB, Mohammad A HANNAN,"Lane departure detection and transmission using Hough Transform Method" ,PRZEGLAD ELECTROTECNIC-ZNY,vol:89,N°5,2013.

[3] Mohamed BOUMEDIENE, " Détection et suivi de voie de roulement par vision ", mémoire de magister, Université des Sciences et de la Technologie d'Oran, 2006.

[4] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and kalman filter," in 2009 16th IEEE International Conference on Image Processing (ICIP), Nov 2009, pp. 3261–3264.

[5] Szutenberg, M. R "Lane detection and tracking for self driving vehicles" .(Author) . 30 Sep 2019

Student thesis: Master, Eindhoven University of Technology

[6] ——. Tue lane tracker. Accessed on 15-Jul-2019. [Online]. Available: https://github.com/RameezI/TUeLaneTracker

[7] B. van de Belt, "A comparative study on modern lane detection algorithms," MSc thesis, vol. 20180830, 2018.

[8] Z.Wang,W.Ren,andQ.Qiu,"LaneNet:Real-TimeLaneDetectionNetworks for Autono mousDriving," arXiv e-prints, p. arXiv:1807.01726, Jul 2018.

[9] TuSimple. TuSimple Benchamark – lane detection. Accessed on 15-Jul-2019. [Online]. Available: https://github.com/TuSimple/tusimplebenchmark

[10] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding," arXiv e-prints, p. arXiv:1712.06080, Dec 2017.

[11] Amin BOROUN, "Lane Detection and Tracking Using a Linear Parabolic Model", Master of Science in Electrical and Electronic Engineering, Eastern Mediterranean University February 2015 Gazimağusa, North Cyprus.

[12] "http://www.neusoft.com/solutions/1234/ - Google Scholar." [Online]. Available: http://scholar.google.se.miman.bib.bth.se/scholar?q=http%3A%2F%2Fwww.neusoft.com%2Fsolutions%2F1234%2F&btnG=&hl=sv&as_sdt=0%2C5. [Accessed: 18-May2013].

[13] "http://www.tmr.qld.gov.au/safety/queensland-road-rules/road-rules-refresher/lines.aspx-GoogleScholar."[Online].Available:http://scholar.google.se.miman.bib.bth.se/scholar?q=http%3A%2F%2Fwww.tmr.qld.gov.au%2Fsafety%2Fqueensland-road-rules%2Froad-rulesrefresher%2Flines.aspx&btnG=&hl=sv&as_sdt=0%2C5. [Accessed: 18-May-2013].

[14] Ahmed Mahmoud1, Loay Ehab1, Mohamed Reda1, Mostafa Abdelaleem1, Hossam Abd El Munim2, Maged Ghoneima3, M. Saeed Darweesh4,5, and Hassan Mostafa1,5 1Electronics and Electrical Communications Engineering Department, Faculty of Engineering, Cairo University, Egypt 2Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University 3Mechatronics Engineering Department, Ain-Shams University, Egypt 4Electronics and Communications Engineering Department, Institute of Aviation Engineering and Technology, Egypt 5Nanotechnology Department, Zewail City of Science and Technology, Egypt

[15] Vertical Lane Line Detection Technology Based on Hough Transform

To cite this article: Xingxing Li et al 2020 IOP Conf. Ser.: Earth Environ

[16] https://ieeexplore.ieee.org/abstract/document/8342913

[17] CANNY SCALE EDGE DETECTION R. Pradeep Kumar Reddy#1, Dr. C. Nagaraju*2, I. Rajasekhar Reddy#3 #1Assistant Professor, Department of CSE Y.S.R.Engineering College of YV University, Proddatur, Andhra Pradesh, India.

[18] https://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm

[19] https://www.analyticsvidhya.com/blog/2021/06/implementation-of-yolov3-simplified/

[20] https://www.analyticsvidhya.com/blog/2021/05/introduction-to-python-programming-beginners-guide/

Abstract:

The field of road safety is increasingly reliant on computer systems, and their role in detecting and intervening in road accidents is based on a variety of technologies; first, capturing the road through on-board cameras, and second applying image processing techniques to locate the road in the captured images edge, and finally detecting the lane.

Key words: lane-line detection ; Yolov3; canny; Hough transform

---

Résumé

Le domaine de la sécurité routière repose de plus en plus sur les systèmes informatiques, et leur rôle dans la détection et l'intervention dans les accidents de la route repose sur une variété de technologies ; premièrement, capturer la route à l'aide de caméras embarquées, et deuxièmement appliquer des techniques de traitement d'image pour localiser la route dans le bord des images capturées, et enfin détecter la voie.

Mots clés : détection de ligne de voie ; Yolov3; prudent; Hough transformer

---

الملخص:

يعتمد مجال السلامة على الطرق بشكل متزايد على أنظمة الكمبيوتر ، ويستند دورها في الكشف عن حوادث الطرق والتدخل فيها على مجموعة متنوعة من التقنيات ؛ أولاً ، التقاط الطريق من خلال الكاميرات الموجودة على متن المركبة ، وثانيًا تطبيق تقنيات معالجة الصور لتحديد موقع الطريق في حافة الصور الملتقطة ، وأخيراً اكتشاف الممر.