Self-Driving Cars: Lane Detection and Vehicle Detection

Divesh Badod, Varsha Venkatachalam and Udit Wasan
CSCI 631 Final Project
Rochester Institute of Technology
Rochester, NY 14623
{db9932;vv2032;uw7854}@rit.edu

Abstract

The recent technological breakthroughs have given rise to various techniques that can be employed to improve the functionalities of an autonomous vehicle. For any smart driverless vehicle, it is important to maintain the safety of the passenger and its surrounding. Intending to improve traffic safety, identification and tracking of road lines and vehicles have become a critical component of vision-guided intelligent vehicle navigation as these tasks provide essential information that helps in estimating its location with respect to the lane and the location of other vehicles relative to itself. This paper is a comparative study between an existing implementation of lane and vehicle detection and a modified version of both these tasks on videos captured from a roof mounted camera of a moving car. Since both these implementations involve primitive Computer Vision techniques, the comparative aspects of this study involve time constraints and the quality of the output.

1 Introduction

Autonomous technology is advancing towards revolutionizing the future of intelligent transportation with the aim of achieving smooth drivability and maximum safety. The autonomous technology concerning vehicles has 6 levels namely no automation, driver assistance, partial automation, conditional automation, high automation and full automation. Lane and vehicle detection fall under the driver assistance level and play a vital role in the partial automation phase for Lane-departure warning, Lane-keeping assistance as well as Vehicle-proximity alert. Vehicle accidents remain the leading cause of transportation deaths and injuries in many countries and most of these incidents occur on national highways. Thus the idea of reducing road accidents and improving safety paved the way for research in vision-guided intelligent vehicles.

Lane and vehicle detection act as key components for building an intelligent transportation system (ITS). In ITS, an intelligent vehicle collaborates with a smart framework to optimise the safety of its passenger and its environment and achieve better traffic conditions. One of the major technologies involved in these tasks is Computer Vision. It is widely employed in various applications of ITS due to its versatile nature, low-cost camera technology and it acts as a powerful tool for intelligent systems to get a sense of their environment.

Detection involves recognition and localization of specific features such as the markings on the surface of any painted road or features that separate the vehicle from its surrounding. Lane and vehicle detection is a complex and challenging task due to the varying nature of the road which can be encountered while driving and similarities between the street and the vehicles concerning its colour. But how to improve the ability of future road vehicles to recognise the lane, efficiently detect curves and the vehicles has been the question all along. However, the recent technological breakthroughs and the various implementation strategies proposed by esteemed individuals working in this domain were able to improve its ability to detect lanes and vehicles [1]. For this study, we are considering one such implementation of lane [7] and vehicle [8] detection and try to modify it. Since the original approach executes both these tasks separately, we have combined both these tasks in a single implementation with a few modifications to the original execution pipeline. With the proposed pipeline, both these tasks can be implemented at the same time by executing a single script, unlike the original approach that has 2 different files and outputs for the respective tasks. Our main focus was to check the extent of improvements in the results by employing traditional Computer Vision techniques rather than using CNN [12] or YOLO [9].

1.1 Data

Since the inception of this theory, there have been various attempts to store data for ITS by capturing images from the front of a car, similarly capturing videos from a roof-mounted camera has been a popular technique in the contemporary data collection methods. Our approach to this problem could have been justified only by using photos rather than videos but since ITS needs real-time detection while driving on roads we decided to use videos as our primary data. Since there are various sources for video data, we decided to collect videos from different sites like Youtube (Figure 1), Kaggle and Github which are not only focused on lanes but also have other vehicles in its view. Some also used some data from the CULane dataset [6]. It is a large scale dataset which was collected by roof-mounted cameras of 6 different vehicles driven by different drivers in Beijing.

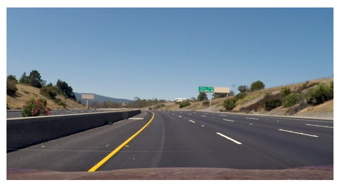

Figure 1: Example frame of an input video.

1.2 Project novelty

The original work for this project [7][8] used primitive CV features in developing the pipelines for lane and vehicle detection separately in addition to that it was not combined into a single script i.e Lane detection and Vehicle detection were independent implementations. Our proposed pipeline was inspired by the original work as well as the work of Ziqiang Sun [10] and motivated us to work with only primitive CV features and bring out the best within its domain. Additionally we have combined the two steps into one pipeline where with a single process through the pipeline, the output will give us lane and vehicle detection simultaneously. Furthermore in Section 2 we will be discussing more about the original pipeline that we used for comparison and Section 3 gives an insight to our modified pipeline.

2 Past Related Work

The concept of Self-Driving cars have been around since the last century but in recent years, the combination of artificial intelligence and autonomous driving technology has proliferated with the inclusion of tech giants such as Tesla, Uber, Waymo and Lyft. At its simplest artificial intelligence helps in training the computers to detect lanes, vehicles, pedestrians and many more obstacles with the help of millions of examples corresponding to the aforementioned subjects. Since it is too complex to write a set of rules for every scenario possible, it is key to train these cars by gaining experience from learning and figuring out how to navigate on their own.

Many of the previous works in lane detection [1] [7] are based on the pipeline as shown in Figure 2. Recent works in road and lane detection [1] mention the inspiration for tackling this particular problem, since road colour textures, boundaries and lane markings are a visual construct for human drivers, autonomous vehicles which have to share these roads with the human drivers need to sense and adapt to the environment similar to the human drivers.

Vehicle detection is a subset of object detection has been researched extensively [4]. Since it's a subset of object detection methods like R-CNN, Fast R-CNN, Faster R-CNN, Haar classifier, YOLO and many work wonders when implemented. The original implementation [8] uses HOG+SVM for feature extraction and decision making respectively.

Autonomous vehicle technology depends on advanced sensing systems that can detect lane boundaries, signs and signals and since there have been several kinds of researches carried out to find various methodologies to build a system capable of sensing lane boundaries and various objects, here we are going to discuss one such technique of lane and vehicle detection which is absolutely an imperative feature in autonomous vehicles aka vision guided vehicles. This lane detection system uses the pipeline shown in Figure 2 and involves the following Phases:

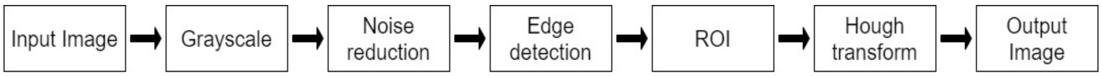

Figure 2: Original pipeline for lane detection.

The first phase of this implementation deals with lane detection and the employed pipeline involves the following steps:

- Converting the input image to grayscale.
- Applying Gaussian blur to smoothen the image.
- Canny edge detection was used to extract the edges.
- A region of interest was created to eliminate all the unnecessary edges.
- Hough line transform was employed to successfully trace a line over the detected edges.
- Finally, these lines were superimposed on the input image to indicate the lanes.

In the field of computer vision, features are a representation that encodes information relevant to the object to be detected. Histograms of oriented gradients [2] are one such feature descriptors which are used in Computer vision and image processing for the purpose of detecting objects. Figure 3 depicts the HOG descriptors for a car present in an image and not present in an image.

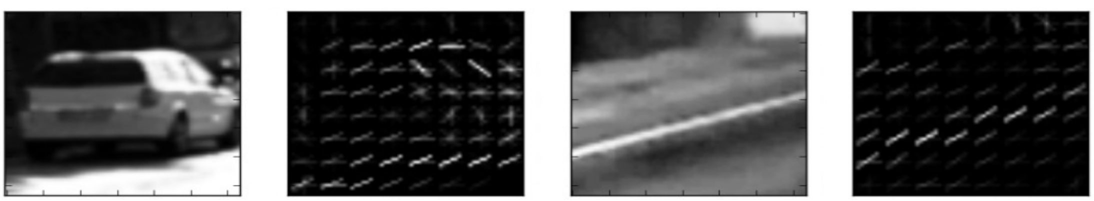

Figure 3: Representation of HOG descriptors for a car patch (left) and a non-car patch (right) [8]

The next phase is based on vehicle detection and it involves the steps as stated below:

- HOG was used for feature extraction.
- A linear SVM was trained for binary classification(car vs non-car).
- Implement a sliding-window technique to detect the presence of the car in the image.
- Combine overlapping boxes and remove false positive detections.

3 Proposed Approach

Our proposed approach is based on the original pipeline as discussed in Section 2 with a few modifications. The pipeline shown in Figure 4 involves the following Phases:

Phase 1 involves lane detection and it employs the steps stated below:

- Converting the input image to HSV to extract yellow lines.
- Converting the input image to grayscale to extract white lines.

- Combining both the above images using colour thresholding for white and yellow lines.
- Applying Gaussian blur to smoothen the image.
- A region of interest was created to eliminate all the unnecessary edges.
- Hough line transform was employed to successfully trace a line over the detected edges.
- Finally, these lines were superimposed on the input image to indicate the lanes.

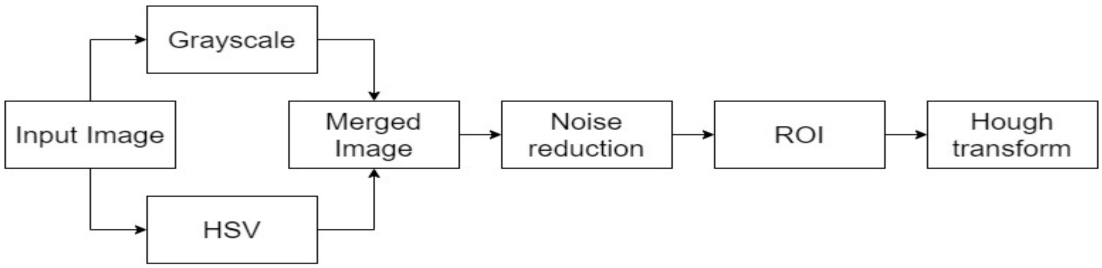

Figure 4: Proposed pipeline for lane and vehicle detection.

For the next phase which involves vehicle detection, implementation can be done by choosing 2 different approaches namely:

- By employing Haar cascade classifier [11].
- By using YOLO [9].

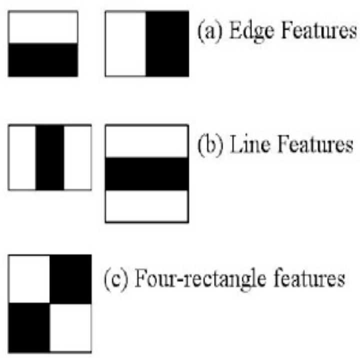

Figure 5: Haar like features [11]

Since we are more focused towards only using rudimentary features of CV we decided to move forward with Haar cascade classifier for vehicle detection. Cascade based classifiers using Haar was proposed by Viola and Jones [11]. A cascade file is prepared using a machine learning approach where it is trained using a lot of positive and negative images of the object. Then features are extracted from it. The Figure 5 shows the Haar features proposed by Viola and Jones [11]. They are just like convolutional kernels. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle. This phase involves the following levels:

- Cascading classifier was used in the second method.
- A pre-trained cascade file(. xml) was used to detect cars in the scene.
- A boundary rectangle was then returned after detecting the cars in the scene.

A video captured from a roof-mounted camera of a moving car is taken as input frame by frame. Each frame is treated as an image and passed in through the pipeline. The output for each phase of the proposed as well as the original pipeline in shown in Figures 7 and 6 respectively and are discussed in detail in the upcoming Section 4.1.

4 Evaluation, Experiments and Results

Since our paper is mostly a comparative study between the original work and our proposed approach the evaluation was conducted based on robust uses, time and space complexity, performance efficiency and technical constraints like feature extraction methods. In the upcoming subsection we will be discussing more about the results obtained.

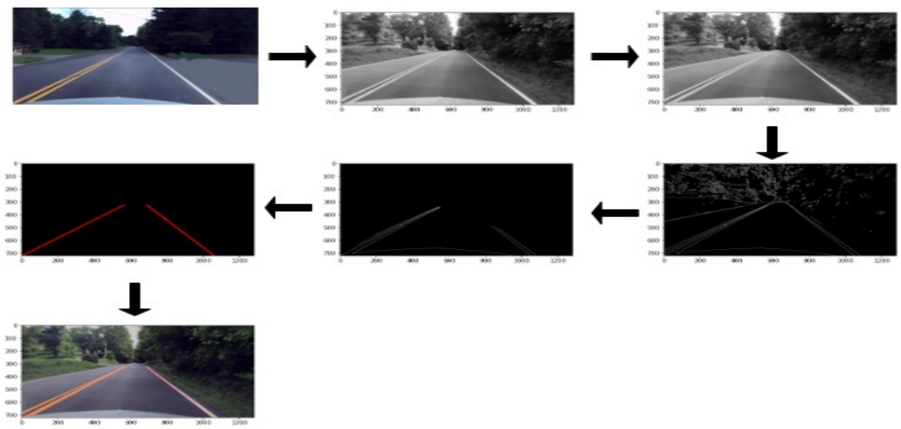

Figure 6: Step by step output for Figure 2

4.1 Results and discussions

The original approach uses canny edge detection for detecting edges in 1st phase. Since canny edge detection is computationally expensive given a large dataset the performance of the technique decreases and runtime becomes excessive. Our proposed method converts the original image to HSV and Grayscale image to extract yellow and white pixels from the converted image respectively using colour thresholding, there are cases where the pixel values closer to these colours are recorded and sent to the next step in pipeline resulting in lines being drawn on unwanted objects like white or yellow coloured cars and extremely bright skies.

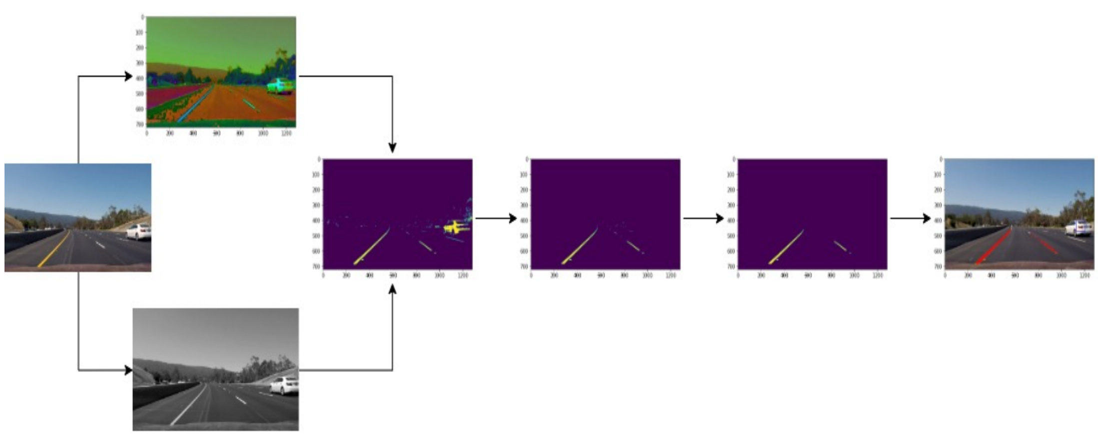

Figure 7: Step by step output for Figure 4

For the purpose of vehicle detection i.e the phase 2 part of proposed and original work, the original work uses HOG for feature extraction which uses a lot of parameters such as size of the cells in which gradients are accumulated, number of gradients as well as the number of cells that compose a block. Haar-like features work faster when compared to HOG+SVM implementation due to lesser parameter constraints and ensemble learning implementation while the file is being prepared. Thus time taken by the original pipeline for lane and vehicle detection separately takes longer than our overall proposed approach.Recent works [5] prove that Haar classifiers are better at vehicle detection than HOG+SVM technique.

Figure 8: Vehicle detection output for original approach.

Recent surveys [3] indicate that there is no possible quantification of results for lane detection, there have been many papers which have tackled the problem of lane and vehicle detection but those papers use their own metrics to quantify the results and no two papers use the same metrics hence comparison becomes difficult. Hence our evaluation also lacks the quantification of results for phase 1 of both the works, and the only option is to evaluate it visually. Our proposed approach worked around 28.5% faster with performance time for final output being 0.05 seconds on average for every frame, and 0.07 seconds for the original work.

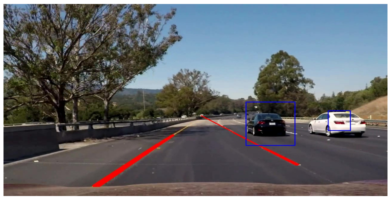

Figure 9: Complete output of the proposed pipeline.

4.2 Member contributions

Divesh Badod:

- Data collection via internet.
- Downloaded and executed the code [7] [8].
- Carried out the implementation for the proposed pipeline.
- Wrote the Proposed Approach and the Evaluation section of the report.

Varsha Venkatachalam:

- Created the Figures 2 and 6.
- Suggested improvements for the original pipeline and helped with the implementation.
- Data collection via internet.
- Wrote the Past Related Work and conclusion section of the report.

Udit Wasan:

- Created the Figures 4 and 7.
- Data collection via internet.
- Wrote the abstract and introduction part of the report.
- Suggested improvements for the original pipeline and helped with the implementation.
- Carried out any necessary changes to the contents of other sections as well as the citations to be made.

5 Conclusion and Future Work

This paper presented a comparative analysis of two of the most important and basic features required in ITS. Our method was superior to the original approach in many aspects but this doesn't mean that it is a better approach overall to tackle the problem. The paper [3] helps in further raising the questions of lane detection problems in different scenarios like low visibility situations during fog, snow or heavy rain, then there are different types of lanes with different widths, colours and purposes like diversion or turning left or right. These questions are only the starting point into ITS for lane detection algorithms and as mentioned in Section 2 imagining every possible situation and carving out rules for said situations will be a long and enduring task. That is why there are so many papers out there and researches conducted focusing on the problems of Autonomous vehicles. We only have the rudimentary implementation in this paper which serves as the starting point to a whole ocean of existing solutions. Since vehicle detection is a subset of object detection YOLO [9] is the all-time fastest and most reliable algorithm for such purpose, moving forward with this project using this algorithm is the only logical solution. Lane detection on the other hand as mentioned above doesn't yet have a good global quantification of metrics, hence lane and road detection are the most challenging tasks. Using CNN or other deep neural networks will without any doubt help with the robustness of the changes in scenarios some of which are mentioned above. But an introduction to newer domains will be never-ending hence continuous research in these tasks will be arduous.

References

[1] N. B. Chetan et al. "An Overview of Recent Progress of Lane Detection for Autonomous Driving". In: *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*. 2020, pp. 341–346. DOI: 10.1109/DSA.2019.00052.

[2] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[3] Aharon bar hillel et al. "Recent progress in road and lane detection: A survey". In: *Machine Vision and Applications* 25 (Apr. 2014). DOI: 10.1007/s00138-011-0404-2.

[4] Mario Hirz and Bernhard Walzel. "Sensor and object recognition technologies for self-driving cars". In: *Computer-Aided Design and Applications* 15 (Jan. 2018), pp. 1–8. DOI: 10.1080/16864360.2017.1419638.

[5] Safirin Karis et al. "Fruit sorting based on machine vision technique". In: 8 (Jan. 2016), pp. 31–35.

[6] The Chinese University of Hong Kong Multimedia Laboratory. *CULane Dataset*. https://xingangpan.github.io/projects/CULane.html.

[7] Andrea Palazzi. *Project 1 - Finding Lane Lines on the Road*. https://github.com/ndrplz/self-driving-car/tree/master/project_1_lane_finding_basic. 2016.

[8] Andrea Palazzi. *Project 5 - Vehicle Detection Project*. https://github.com/ndrplz/self-driving-car/tree/master/project_5_vehicle_detection. 2016.

[9] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[10] Z. Sun. "Vision Based Lane Detection for Self-Driving Car". In: *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*. 2020, pp. 635–638. DOI: 10.1109/AEECA49918.2020.9213624.

[11] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.

[12] Q. Zou et al. "Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks". In: *IEEE Transactions on Vehicular Technology* 69.1 (2020), pp. 41–54. DOI: 10.1109/TVT.2019.2949603.