ECE 59500 – Digital Image Processing

Road Lane Detection Using Computer Vision and Deep Learning

Project Report

Team:

Aishwarya Madupuri Suresh

Gowri Kurthkoti Sridhara Rao

Sivakami Thinnappan

Abstract

Self-driving cars are one of the most talked topics where applications of computer vision is discussed. In this project, we aim to implement one of the basic functions of a self-driving cars; lane detection. Using the camera mounted on the front of the car, road lanes can be detected, and these detected lanes are eventually used to help the car maintain itself in the center of lane and not sway out of it. Lane detection can be carried out using both traditional computer vision techniques and deep learning algorithms. Our project draws a comparative study between both the methods and their results.

Background and Objectives

A Self-driving car/Autonomous car is a vehicle capable of sensing its environment and operating without human involvement. A human passenger is not required to take control of the vehicle at any time, nor is a human passenger required to be present in the vehicle at all. An autonomous car can go anywhere a traditional car goes and do everything that an experienced human driver does.

Autonomous cars rely on sensors, actuators, complex algorithms, machine learning systems, and powerful processors to execute software. Autonomous cars create and maintain a map of their surroundings based on a variety of sensors situated in different parts of the vehicle. Radar sensors monitor the position of nearby vehicles. Video cameras detect traffic lights, read road signs, track other vehicles, look for pedestrians and carry out lane assistance. Sensors like Lidar (light detection and ranging) and Ultrasonic sensors help measure distances, detect road edges, and identify lane markings. Sophisticated software then processes all this sensory input, plots a path, and sends instructions to the car’s actuators, which control acceleration, braking, and steering. Hard-coded rules, obstacle avoidance algorithms, predictive modeling, and object recognition help the software follow traffic rules and navigate obstacles.

Driver inattention and visual interference are some of the leading causes of accidents and road fatalities. Development of advanced driver assistance systems (ADAS) can improve driving safety. Several ADAS, such as the lane departure warning system, lane keeping system, target localization, and obstacle avoidance are applications of lane detection. Lane mark recognition is one of the most important parts of road understanding.

Automated driving requires a full understanding of the environment around the automated vehicle through its sensors. Vision-based methods have lately been boosted by advancements in computer vision and machine learning. Regarding environmental perception, camera-based lane detection is important as it allows the vehicle to position itself within the lane.

Vision-based Lane detection is a technique to locate the lane boundaries in an image without prior information of the road. Such an algorithm influences the performance of lane tracking, which tracks the lane edges from frame to frame. The feature-based, approach locates the lanes, such as line width, edge, color, intensity, texture, and gradient features. For example, the lane edge features can be extracted by Hough transform (HT). Road lane-markers, candidates, and obstacles can be segmented using the gray level histogram of the road.

Accurate and reliable lane detection is vital for the safe performance of lane-keeping assistance and lane departure warning systems. However, under certain challenging circumstances, it is difficult to get satisfactory performance in accurately detecting the lanes from one single image as mostly done. Since lane markings are continuous lines, the lanes that are difficult to be accurately detected in the current single image can potentially be better deduced if information from previous frames is incorporated. Hence, machine learning algorithms, especially deep learning algorithms are used for the detection.

A comparison between traditional vision feature based and deep learning approaches is essential to understand the pros and cons of the two. We believe our project will highlight the differences and similarities and we aim to present the results of both the implementation.

Methods

Two different methods are discussed in this section. First one being the computer vision-based method and the second one is the deep learning-based method.

- **Computer vision approach**
  Tools and libraries: Python programming language, open-cv
  The video is processed frame by frame as images, and a series of operations are carried out on each frame to identify the lanes in that frame.
  Steps to identify the lanes include:
  1. Canny edge detection
  2. Masking the image not inside region of interest
  3. Identifying all the lines inside the ROI using Hough transform
  4. Averaging the lines to identify the left and right line
  5. Displaying the lines on the frame

  Canny edge detection : A digital image consists of changes in intensities based on the object in the image. In the image, if there is a sharp change from one intensity to another, these changes are referred to as edges. Canny edge detection is a multi-stage edge detection algorithm to detect a wide range of edges in images. Major steps in canny edge detection include gaussian filtering, identifying intensity gradients in the image, thresholding, non-max suppression, hysteresis. Canny edge detection has low error rate and very good edge detection.

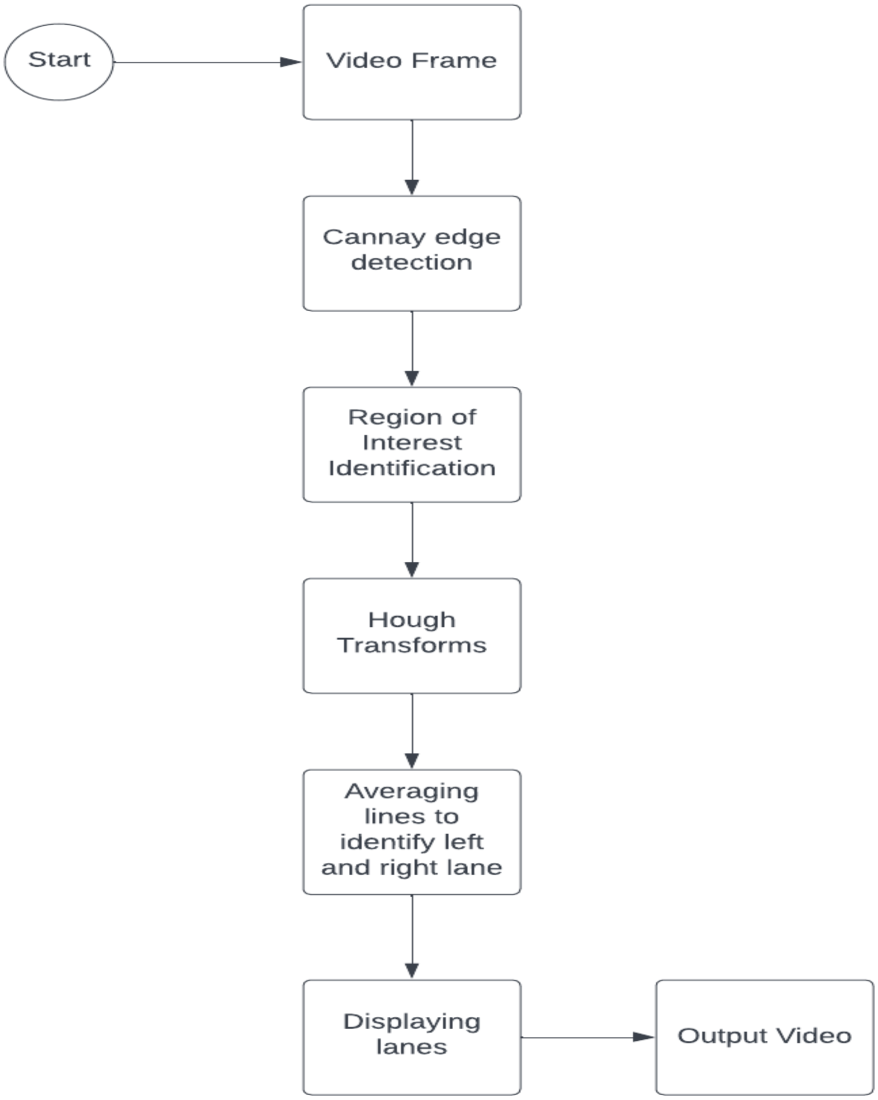

Fig 1: CV approach steps

Region of Interest Identification: When camera is mounted on the car for the lane detection and assistance purposes, the lanes are present in a triangular shape towards the lower half or 2/3rd of the frame. Which means, the upper part of the frame is not necessary, and any edges present in that region can be ignored. Hence, a triangular mask that keeps only the lower triangular portion of the frame is used.

Hough transforms : The Hough transform is a feature extraction technique to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform. The classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. Using Hough transform, all the lines in the masked image within the triangle are identified. These lines are further used for identifying the lanes.

Averaging lines to identify left and right lane: Using the Hough transform will result with all the lines that are present on either side of the camera. These lines have to be averaged or smoothed to identify one line on either side. The idea here is to iterate through all the lines detected, group them as left and right line groups based on their slopes. These lines are averaged, and the averaged lines parameter is then returned as the left and right lane.

Displaying lanes : Although this step seems like a simple task, it includes identifying the coordinates of the lanes using the averaged parameters. These coordinates usually start from the bottom of the image and are marked until 3/5th part of the image height. Open CV has a function that can draw the lines provided the x and y coordinates of the start and end of the line.

These steps are carried out for each incoming frame to detect lanes in a video.

- **Deep Learning approach**
  Tools and libraries: Python programming language, open-cv, Keras, matplotlib, numpy

Introduction to CNN

CNN is a type of deep learning model for processing data that has a grid pattern, such as images, and designed to learn spatial hierarchies of features automatically and adaptively, from low- to high-level patterns. CNN is a mathematical construct that is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification. A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation. In digital images, pixel values are stored in a two-dimensional (2D) grid, i.e., an array of numbers, and a small grid of parameters called kernel, an optimizable feature extractor, is applied at each image position, which makes CNNs highly efficient for image processing, since a feature may occur anywhere in the image. As one layer feeds its output into the next layer, extracted features can hierarchically and progressively become more complex. The process of optimizing parameters such as kernels is called training, which is performed so as to minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation and gradient descent, among others.

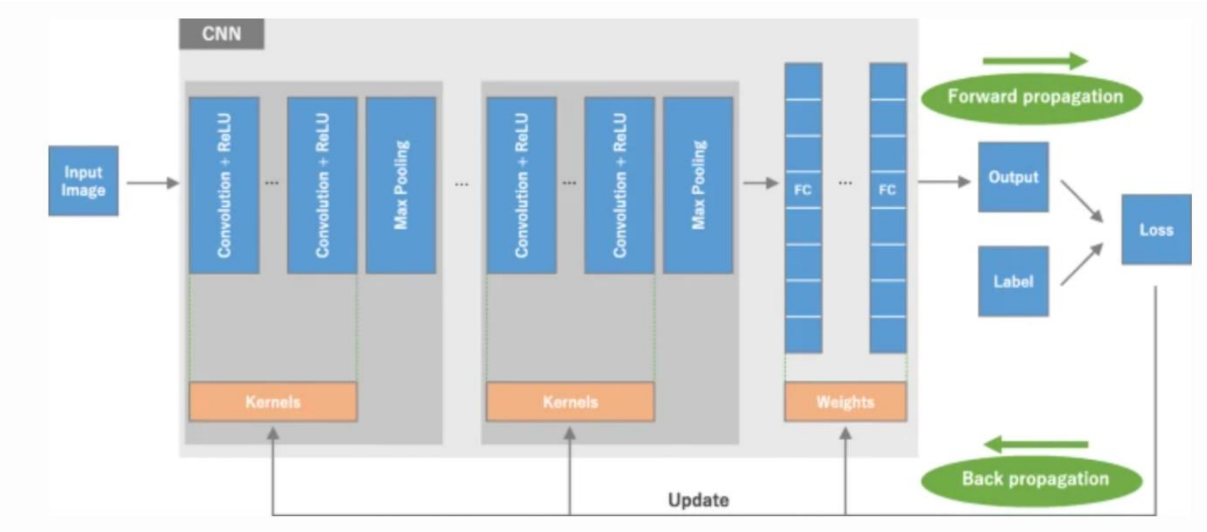

Fig 2: Convolutional Neural Network Architecture

A pre-trained CNN model has been used to make the lane markings on the image/video. The pre-trained model has the following layers:

1. Conv2D
2. MaxPooling2D
3. Conv2DTranspose
4. UpSampling2D
5. Dropout

An image is used as the input to the model. In case we want the lane markings to be made on a video, we divide the video into individual frames and pass each frame as the input to the model.

The following are the steps:

1. Take in the image of a road
2. Re-size the image to the size required by the model
3. Predict the lane to be drawn from the model in green color
4. Recreate an RGB image of a lane
5. Merge with the original road image

These steps are carried out for each incoming frame to detect lanes in a video.

Results

- **CV – results (images)**
  In this section, we present the image results of the computer vision approach. As it can be observed in figure 3, on a straight and clear road image, the lanes are detected correctly. In figure 4 and 5, there are some deviations from the solid line on the road. This is due to the averaging operation. All the lines on either side of the road are averaged and one left lane and right lane are identified. If the image picks up farther lines, then it might affect the smoothening process.

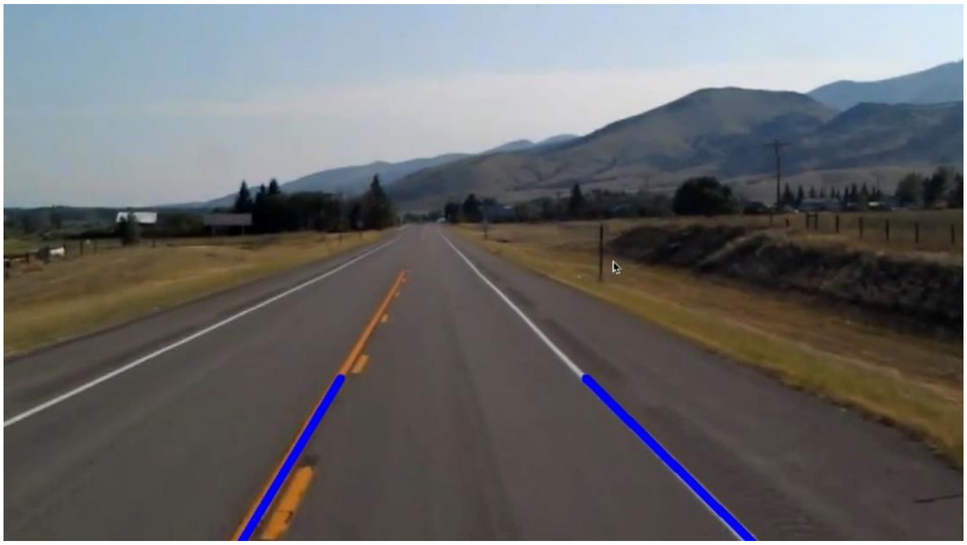

Fig 3: CV Result image 1

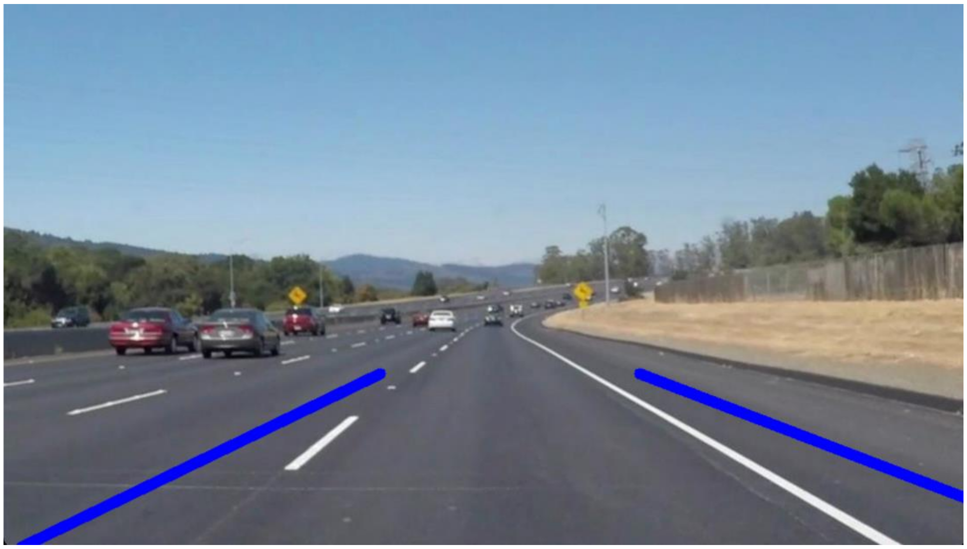

Fig 4: CV Result image 2

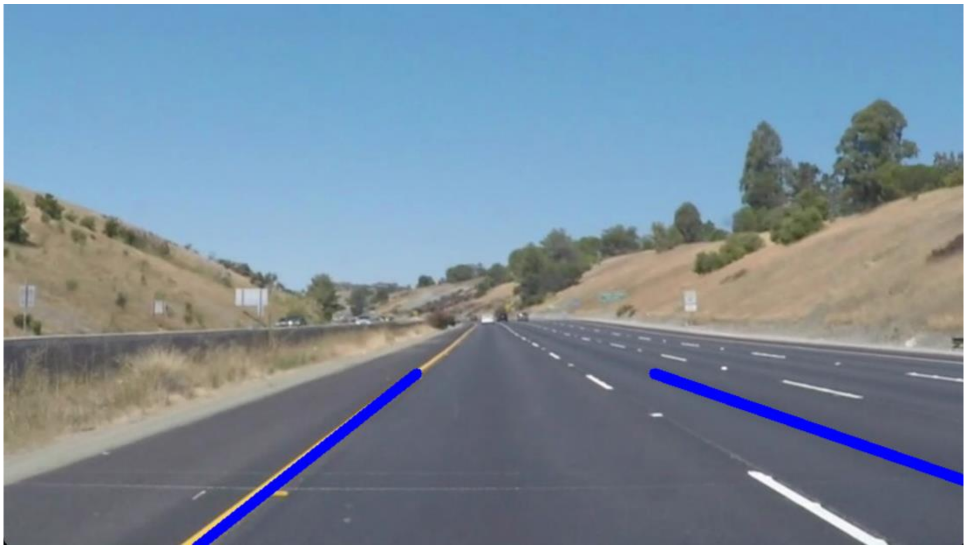

Fig 5: CV Result image 3

- **DL – results (images)**
  In this section, we present the image results of the deep learning approach. As it can be observed in figures 6 and 7, deep learning approach is capable of detecting the lane accurately in most of the circumstances. In these two images, the background intensity is reduced and added with the lane image while carrying out the weighted addition process.

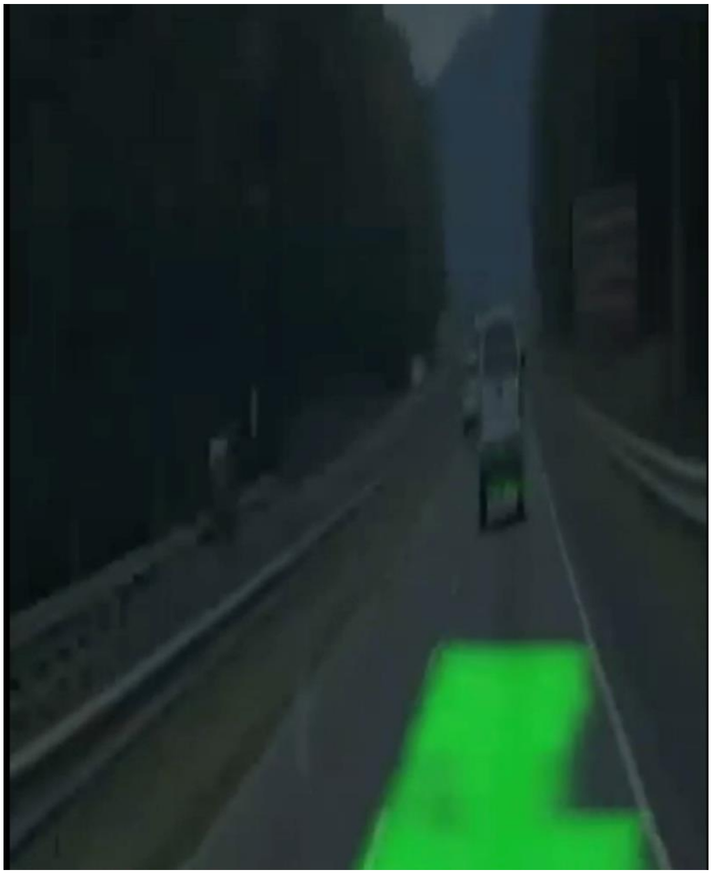

Fig 6: DL result image 1

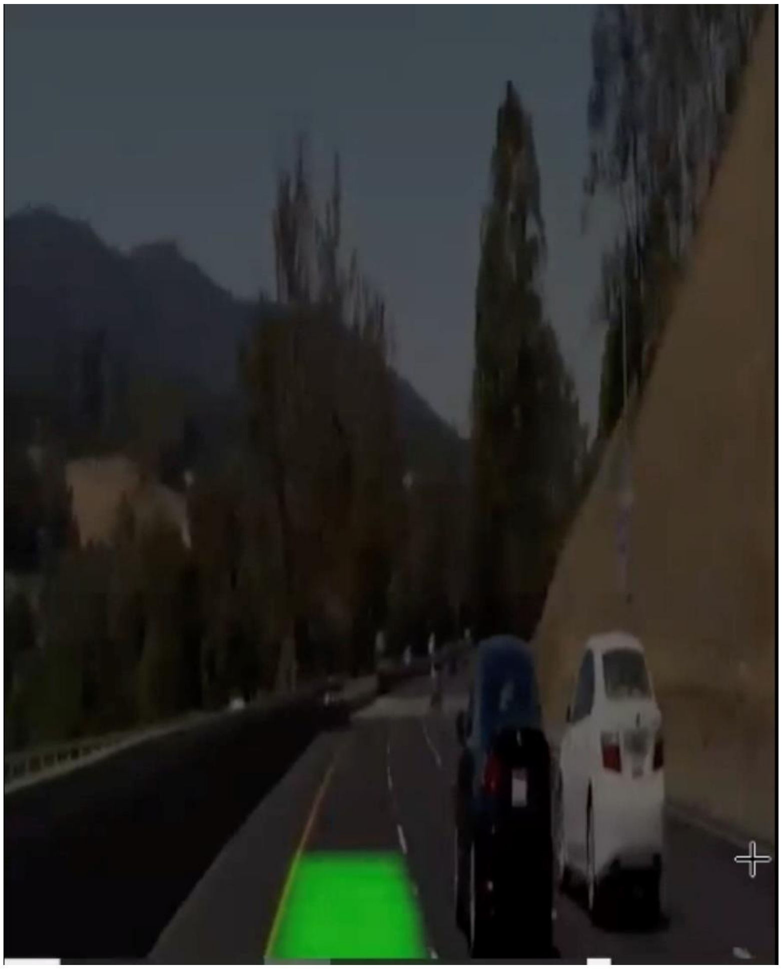

Fig 7: DL result image 2

Conclusions

The objective of our project is to carry out lane detection using computer vision approach and deep learning algorithms. Both the methods were implemented, and the results were observed. Using the observations, following conclusions are made.

In CV results, when the road has horizontal lines which are averaged with the vertical actual lanes, provides faulty result deviating the detected lanes further away from the actual lines. In the curve, the detection is not very smooth, and the code crashed at one point without being able to estimate the lane with the help of slope and intercept parameters. On the bright side, CV approach is computationally in-expensive. It does not require too many libraries or complicated algorithms to detect the lanes. Hence, coding this up into a small programmable camera also can provide reasonable results.

DL on the other hand, performed fairly accurately for almost all the input videos. On the downside, deep learning approach is computationally expensive. We experienced RAM crashing, and dead kernel situations a couple of times while trying to predict the lanes using the pre trained model.

In conclusion, both the approaches have their own advantages and disadvantages. In an ideal scenario, it would be best if deep learning methods can be made computationally in-expensive.

References

[1] What is an Autonomous Car?

[2] Practical limitations of lane detection algorithm based on Hough transform in challenging scenarios

[3] Building a lane detection system

[4] Deep Learning vs. Traditional Computer Vision

[5] DIY Lane Keeping

[6] Hough Transforms

[7] Understanding of a convolutional neural network

[8] Lane Detection with Deep Learning

[9] A Computational Approach to Edge Detection

[10] Hands-On Tutorial on Real-Time Lane Detection using OpenCV (Self-Driving Car Project!)