Machine Learning/Deep Learning Assignment

Overview : For completion of this lane detection assignment, a two way approach was followed. To understand the core fundamentals of detection in a image, firstly an foundation approach was selected which uses canny detection and hough transform algorithm with some preprocessing to detect lanes as illustrated in approach 1. Onces this fundamental were build a more advances approach with deep learning was selected, which utilizes the fully connected CNN pre trained model to do prediction on lane as illustrated in approach 2.

Approach-1: Lane Detection using canny detection and Hough transform (without Deep Learning)

In this approach opencv is used predominantly to detect lanes in an image[1]. An input pipeline consists of techniques applied on the image in a sequences as illustrated in below Fig 1.

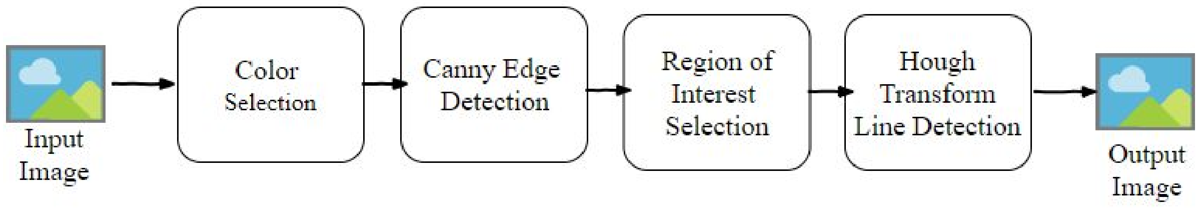

Fig 1: Input Pipeline of Approach 1

Color selection : The images are loaded in RGB color space. The intention of this technique is to select only yellow and white colors in the images. Using ***cv2.cvtColor***, we can convert RGB image into different color space. For example, HSL and HSV color space. The output of this will be both the white and yellow lines being clearly recognizable.
Canny Edge Detector : The main objective of this technique is to to detect edges in order to find straight lines especially lane lines. For this we have to convert images to grayscale, smooth out rough lines, find the edges So firstly converting the white and yellow line images from the above process into gray scale for edge detection. Than we should make the edges smoother, as the above images may have many rough edges which causes many noisy edges to be detected. And lastly use canny edge detection to detect the edges.
Region of Interest selection : When finding lane lines, we don't need to check the cloud and the mountains in a image. Thus the objective of this technique is to concentrate on the region of interest for us that is the road and lane on the road.
Hough Transform Line Detection : The objective of the Hough Line Transform algorithm is to detect lines in the edge image outputted from the above process. Moreover there are multiple lines detected for a lane line. Thus we should come up with an averaged line for that. Also, some lane lines are only partially recognized. Thus extrapolation of the line to cover full lane line length is required.

Performances Metrics:
The performances metrics to be used for evaluating the build model are confusion matrix, Precision, recall, ROC curve, PR curve, Mean Squared Error(MSE) and mAP score.

Disadvantage:

- Only detects straight lines.

Predicted Results:

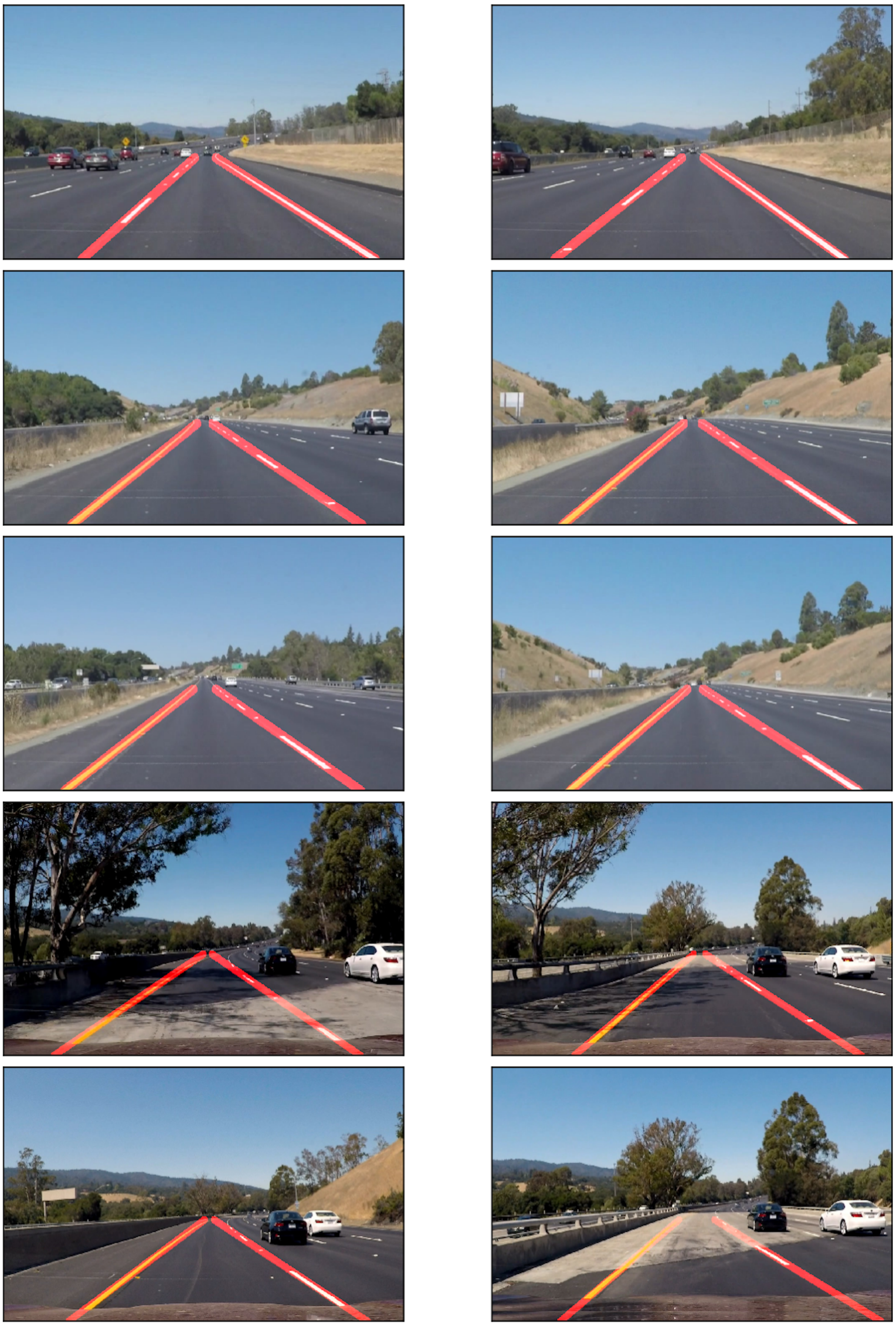

Fig 2: Predicted Results of Approach 1

Approach-2: Lane Detection using Fully connected CNN (with Deep Learning)
In this approach deep learning is used to detect lanes in an image, to solve the problem of identifying curves lane which the earlier stated approach faced. As the deep learning model may be more effective at determining the important features in a given image than a human being using manual gradient and color thresholds in typical computer vision techniques, as well as other manual programming needed within that process[2]. Specifically, the plan is to use a convolutional neural network (CNN), which are very effective at finding patterns within images by using filters to find specific pixel groupings that are important. The aim of this approach is for the end result to be both more effective at detecting the lines as well as faster at doing so than common computer vision techniques.

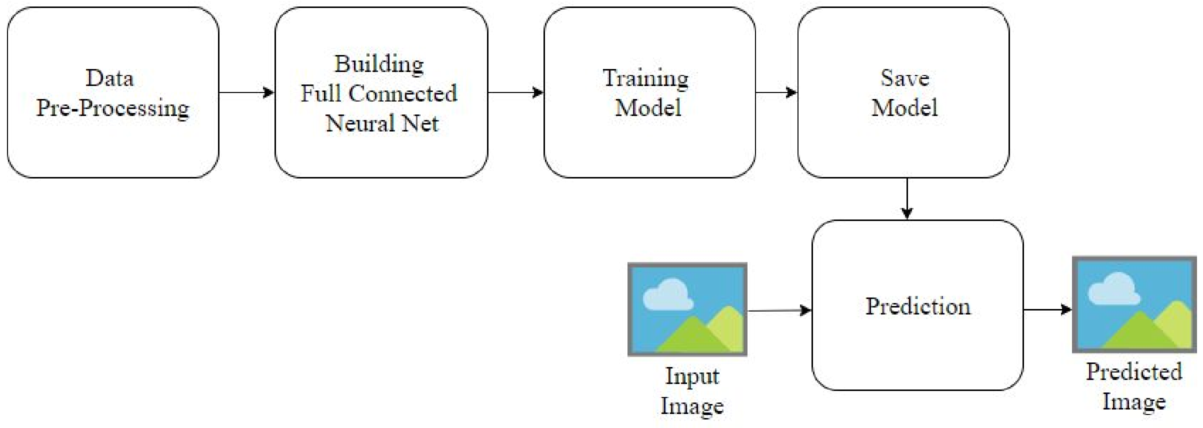

Fig 3: Input Pipeline of Approach 2

The input pipeline illustrated in the above diagram is followed to solve the lane detection problem using deep learning and fully connected CNN neural nets.
Data Pre-Processing : The dataset used for this approach is a BDD100K Lane Marking Dataset. The dataset consist of Images and label. In this phase the dataset is read and then saved as a pickle files with “train_images.p” and “train_labels.p” as seen in the github folder structure.
Building Fully Connected Neural Net : In this phase building of a neural network architecture is started which is inspired by the Udacity self-driving car challenge [3],which consist of 7 convolutional layers, 3 polling layers, 3 upsampling layer and 7 deconvolution layers, built with Keras on top of Tensorflow. The neural net assumes the inputs to be road images in the shape of 80 x 160 x 3 with the labels as 80 x 160 x 1. The output layer of deconvolution layer is a filled with the lane detected as a predicted image[3].
Training Model: Before the model training start the dataset is divided into training set and a validation set created using sklearn train_test_split method. The model is compiled with mean squared error as the loss function and trained over 20 epoch on the whole training dataset and validated on the validation dataset.
Save Model: After the model training is completed. This trained model is saved in a json along with its weights. Now this saved model can be used for doing prediction on the unknown data.
Prediction: In this phase the saved model is loaded along with its weights. The test image is pre processed to be of the shape 80 x 160 x 3, as the model expect that as the input shape. Onces done the preprocessed image is given as an input to the models which predicts the output over which the filling of green color is added as an region of the lane detected.

Performances Metrics:
The performances metrics to be used for evaluating the build model are confusion matrix, Precision, recall, ROC curve, PR curve, Mean Squared Error(MSE) and mAP score.

Predicted Results:

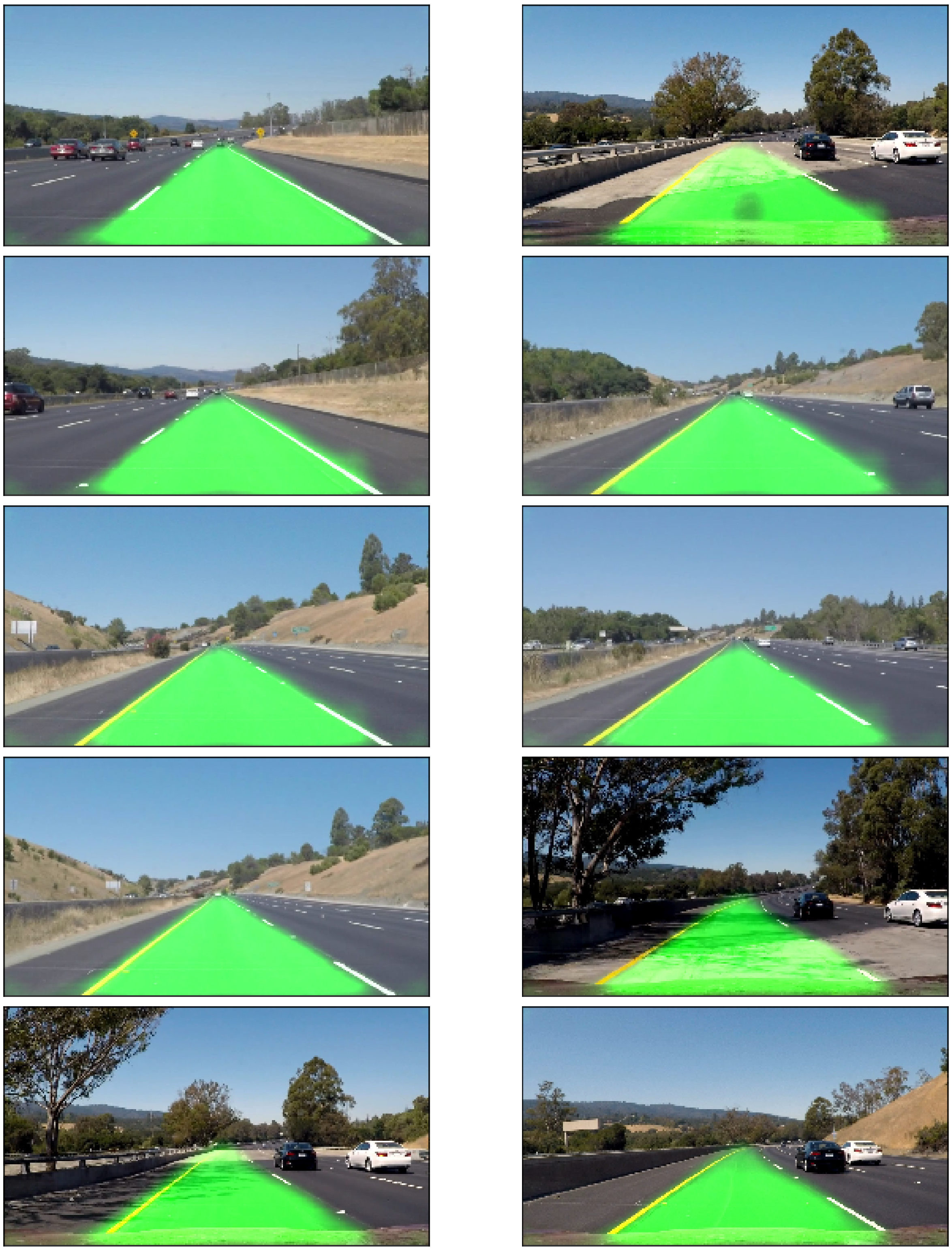

Fig 4: Predicted Results of Approach 2

References :

[1] Chuan-en Lin (2018 , Dec. 17). "Tutorial: Build a lane detector", Available: https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132.[Apr 06, 2019]

[2] Aydin Ayanzadeh (2018 , Mar. 19). "Udacity Advance Lane-Detection of the Road in Autonomous Driving", Available:https://medium.com/deepvision/udacity-advance-lane-detection-of-the-road-in-autonomous-driving-5faa44ded487.[Apr 06, 2019]

[3]Udacity, self-driving-car, (2017), GitHub repository, https://github.com/udacity/self-driving-car.[Apr 06, 2019]