Traffic Lane Detection & Recognition

Pranav Madathil Sasi

Abstract

Lane detection is a difficult task. For decades, it has drawn the attention of software engineers especially computer vision community. Lane identification is fundamentally a multi-feature detection problem that has shown to be challenging for ML and computer vision techniques. Although several ML methods are utilized for road lane detection and recognition, which are mainly used for classification rather than feature creation. Modern machine learning algorithms, on the other hand, may be used to select features that are rich in recognition and have demonstrated reliable performance in feature detection and identification tests. However, these strategies have not been completely applied in terms of lane detecting efficiency, accuracy and safety of peoples. Increasing safety and minimizing traffic accidents, hence saving lives, are two major goals of ADAS (Advanced Driver Assistance Systems). Lane detection is a critical building block in the development of ADAS and autonomous cars. This thesis proposes a solution to this problem utilizing computer vision and deep learning approaches. This entails using a multi-layered neural network, which use mathematical principles to minimize the loss from prediction v/s actuals in order to converge towards a final model, essentially learning as they train on live data. The suggested approach is both robust and quick enough to detect lanes in real time.

Keywords: Lane Detection, Deep learning, Convolution Neural Network, Computer vision.

1. Introduction

The purpose of developing Automated Driving Systems (ADSs) is to minimize the traffic accidents, pollution and reducing stress related driving (J. Crayton & Meier, 2017). The majority of road accidents are caused by situational factors associated to the driver, such as intoxication, tiredness, and inappropriate driving procedures. To some extent, smart self - driving cars can mitigate these human influences. Intelligent vehicles are equipped with smart infrastructure to provide a safer driving environment. The recognition of road lanes has played an essential part in driver aid systems present in smart vehicles, which give information such as lane structure and vehicle location relative to the lane. The most compelling rationale for introducing autonomous capabilities to vehicle, is to meet the safety criterion. ADS system comprises with lane departure warning (LDW) (Bhujbal & Narote, 2015), Lane Keeping Aid, and Adaptive Cruise Control (ACC) (V & S, 2015) can assist drivers in analyzing their present driving environment and providing appropriate feedback for safe driving or alerting the driver in risky situations. This type of assistant driving technology is intended to evolve over time to produce more accurate outcome (Feiniu, et al., 2015).

According to the analysis, the chance of accidents is significantly higher than usual in the complicated traffic condition such as numerous cars on the road and over speed driving. Even in such congested traffic situation, humans are able to identify and extract the road conditions like color, uneven surfaces, texture, boundary and lanes (Pei, et al., 2014). These perceptual points of human while driving are essential for the foundation of the autonomous vehicle design and development.

When a human drives a car, the easiest thing is to keep it in its right lane. Most people can perform this after basic training as long as they are not distracted, drunk, or otherwise impaired. However, it is quite simple for a human to maintain a vehicle in the middle of the lanes. For a machine or computer to solve the task, it is far more challenging.

How come this is challenging task for a computer? Because the computer has no idea what a yellow line and a white line on a road imply, or what the spacing between each line in a video feed or stream means. The first step is to teach them the significance of these lines and gaps by teaching the computer to recognize lines and gaps in a video stream. Various computer vision algorithms can assist a computer in learning to recognize these lines or lanes. Techniques such as camera calibration, gradient thresholds or concentration of color and perspective transformation of view. Camera calibration is removal of the inherent distortion of the camera used to collect road footage. Obtaining a bird's eye view of a road is referred to as perspective transformation. Other difficulties encountered in detecting lanes include harsh weather conditions and poor sight. Such as heavy rain, fog, snow, and so on.

To overcome these problems, this thesis proposes road lane detection system in real time for autonomous vehicle. This is accomplished by utilizing computer vision with the help of multi layered fully Convolutional Neural Network. The Lane detection comprises of road localization, determining the relative location of the vehicle to the road, and analyzing the vehicle's heading direction. One of the primary methods for detecting road limits and lanes is to use the vehicle's visual system. Thus, ensuring the autonomous vehicles are within the appropriate lanes to prevent collision and accidents with the other vehicles on the road even on the extreme weather and poor visibility conditions.

2. Background

The ability to discern road lanes has been extensively researched. Several tactics and approaches have been proposed in an attempt to overcome these obstacles. The availability modern super-fast computer led many specialists and researchers were inspire to learn more about the lane detection. Lane detection is classified into two types: feature-based lane detection and model-based lane detection. To detect the lane, the lane is separated from the original road image using features such as edge and color of the scene. This kind of detection is known as feature-based lane detection.

(Hojatolah & Babak, 2016) introduced a new method to detect the lane using feature-based detection technique. The system used color characteristics to learn and predict the straight section of the route as well as offer vanishing spots. A mix of newly learnt color characteristics and a vanishing point algorithm were utilized to extract the road area. The result, with an accuracy of 90%, recall of 92%, and F-Measure of 84%, demonstrates the effectiveness of the new approach when compared to existing methods, demonstrating the superiority of the suggested method.

(Alireza & Routeh, 2017) suggested a novel lane detection approach that consists of three components: (1) lane detection, (2) perspective mapping, and (3) lane selection. To address the perspective problem in the acquisition of input images, the authors provide a unique technique based on Principal Component Analysis (PCA). This back projection also aids in following the boundary of lines in the distance and locating lanes more precisely. Furthermore, the authors use a rotated rectangle model, which results in more precise lane recognition. Gaussian filter, Sobel operator, canny edge detector, and contour retrieval technique are used in lane detection. The suggested technique is tested in five scenarios: occlusion, light changes, crowded and non-crowded scene, and noisy picture. Lane selection is accomplished by distinguishing between accurate and incorrect contours.

(MD, et al., 2008) paper describes a vision-based lane identification approach that works in real time and is robust to changes in lighting and shadows. The input dataset consists of a sequence of color images captured from a moving vehicle using a camera mounted on the vehicle. The gathered image size will be reduced to 620x480 pixels using the Gaussian pyramid method. The F.H.D. algorithm is used to remove the shadow in the road from the captured image. And the lanes were detected using the Hough transformation with restricted search area. The system was implemented using MATLAB 7.1.

(B, et al., 2015) proposed an algorithm for detecting road limits and painted lines for intelligent and autonomous vehicles. It incorporates the Hough Transform to initialize the algorithm each time it is required, as well as the Canny edges detector, least-square approach, and Kalman filter to minimize the adaptive region of interest, forecast the future road borders' position, and lines parameters.

(J, 2019) the authors explained about the Advanced driver assistance system (ADAS) and its future scope. Here multiple perceptions are integrated into single sensor model and multiple algorithms are used to detect the lanes such as Convolution neural network (CNN), Recurrent Neural network (RNN), Vision-based Lane detection algorithm- LANA, Hough transform algorithm, Random sample consensus (RANSAC) and B-snake model system

(B, et al., 2008) the adaptive double threshold approach was used to extract the effective characteristics of the lane in the road scene after converting the original RGB picture to CIE color space. (A & A, 2009) developed a technique for identifying the left and right lanes using a two-dimensional linear filter that could remove interference noise during the detection process while preserving the intrinsic properties of the lane as much as possible in the distant view. (J, et al., 2015) concentrated on the effect of light fluctuations on lane detecting Yellow and white lanes were chosen as candidate lanes based on their lighting invariance, and the lanes were found using the clustering approach from candidate lanes. (Bar, et al., 2014) completed a survey during the previous 5 years on the methods and algorithmic techniques developed for the various modalities. They give a basic breakdown of the problem into functional building pieces and comment on the vast range of possible techniques within this framework. (H, et al., 2013) presented a gradient-enhancing conversion approach for lighting-resistant lane detecting. Relying on linear discriminant analysis, the suggested gradient-enhancing conversion method creates a new gray-level picture from an RGB color image. At lane borders, the transformed pictures exhibit substantial gradients. (P, et al., 2014)suggested lane-mark extraction in pictures by evaluating the borders of

Regions Of Interest (ROIs) and splitting the boundary images into sub-images to determine the local edge orientation of each block and eliminate problematic edges. (S. Zhou, 2010) proposed a lane recognition technique that uses Gabor filters and a lane geometrical model with four parameters (beginning location, original orientation of the lane, width of the lane, and curvature of the lane).

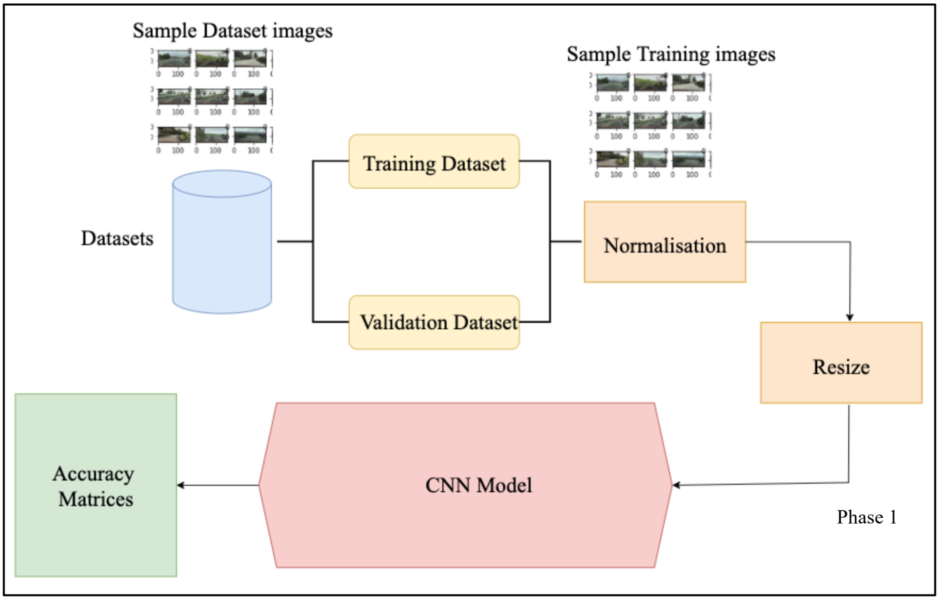
3. Methodology

The goal of this research is to detect and recognize the road lanes in real time for autonomous vehicles. A model that recognizes the lanes in a video is created by using multi-layer convolutional neural network, Deep Learning methods, and Keras libraries.

3.1 (Jason, 2019)System Architecture

The proposed system contains two phases:

- The first phase involves training. The constructed model will be taught to achieve the required result. The dataset is split into two sections: training and validation. To feed the model, the training data set will be processed through image normalization and resizing. Based on these images, the model will predict the lanes, and the model will continue to be trained in order to make accurate predictions. This procedure is illustrated in Fig.1phase 1.
- The second phase involves testing and validating the trained model with real time inputs. During this step, the input video will be fed into a trained model that will predict the lanes in the video in real-time. This procedure is shown in Fig.1phase 2



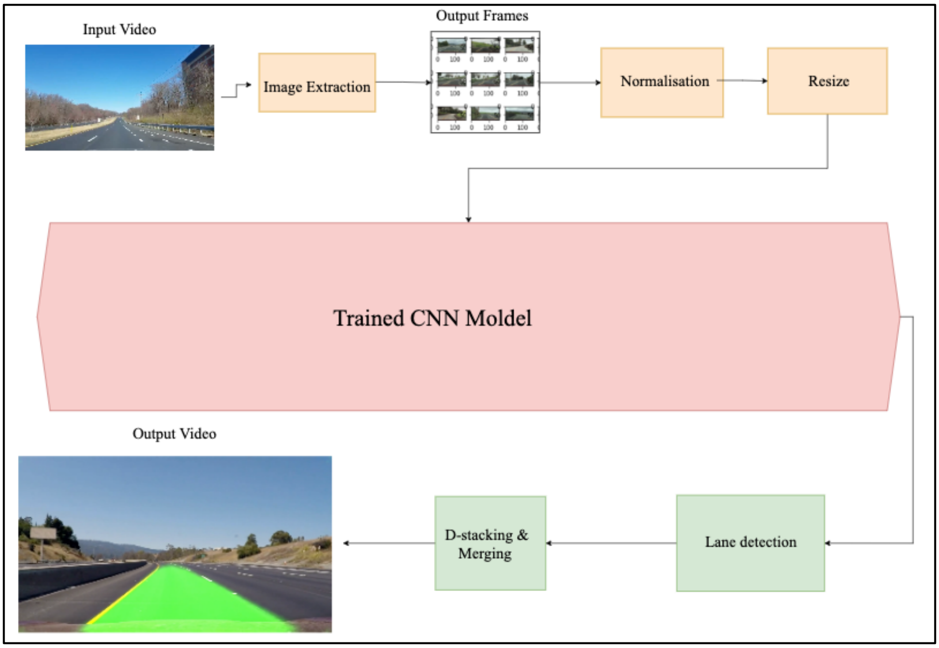


Fig.1 Training and Testing phase Architecture.

3.2 Dataset Overview

The dataset used for this project is from public dataset in Kaggle. The attributes of dataset are.

- The videos utilized were captured in 720p landscape or horizontal format with the ratio of 30 Frame Per Second and 1280 * 720 pixel resolution in x and y axis respectively.
- The collection contains almost 21,000 pictures extracted from various traffic footage. All photographs were captured from the video on various lighting settings such as day and night, as well as weather conditions such as fog, rain, snow and sun. Images also includes various traffic scenarios and road curvatures (Fig. 3).
- The collected road images also include various challenging places such as traffic junctions, road construction sites, and maintenance.
- Approximately 400 images are worthless due to concealed lines and blurriness.
- More than 1600 images are extracted from more curved road lane videos in order to get a label with a larger dispersion.
- Approximately 1500 photos were first taken from those datasets to compensate the time series.

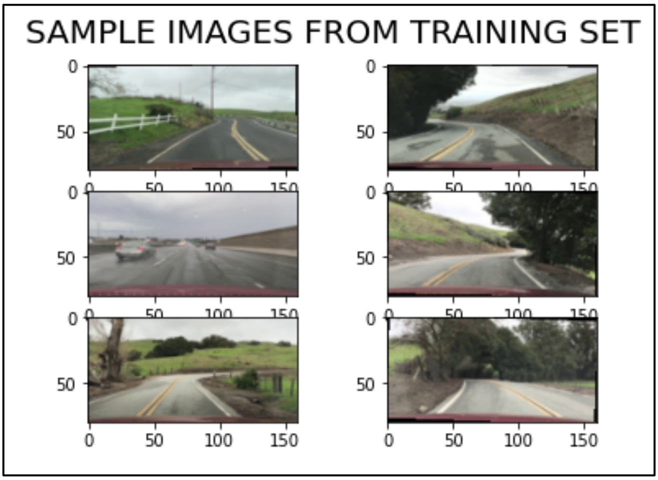


Fig.3 shows sample images from dataset.

3.3 Dataset Preprocessing

Data preprocessing is an important task in ML. It focuses on addressing faults that might disrupt the learning process, including as noise, outliers, and omission. Which helps to improve the efficiency of model by generating good accuracy (Zelaya, 2019).

As part of the data pre-processing, the images are resized to 80x160x3 has been done without normalization and grey scaling. This has a slight difference in aspect ratio when compared with the original image. This process is just like scaling down the image aspect ratio. Due to different image sizes present in the dataset, image should be resized in to one shape before feeding to the model. The resultant data shape is (11487,80,160,3). Here 11487 is the training image count each measuring 80 * 160 px and includes colored images as per the RGB value 3. After, that resized image carried out to normalization process. Normalization is the process of eliminating unnecessary characteristics from image and redundant data. Image normalization refers to how we modify the pixel intensity of an image. The normalization is carried out by specifying the type as cv2.NORM_MINMAX. in this situation the minimum value of alpha is zero and the maximum value of beta is 1. So that the normalization function can work in between these values very effectively. The normalization technique will help to get a clear image by altering the intensity of pixels and boosting the entire image contrast (Rob, 2021). To double the images in training set by adding the horizontally flipped images without changing its actual label of original images extracted from the different road videos. After this the dataset is split into test and validation by using *sklearn. train_test_split* with the ration of 80 % train set and 20 % validation set.

3.4 Building a CNN Model

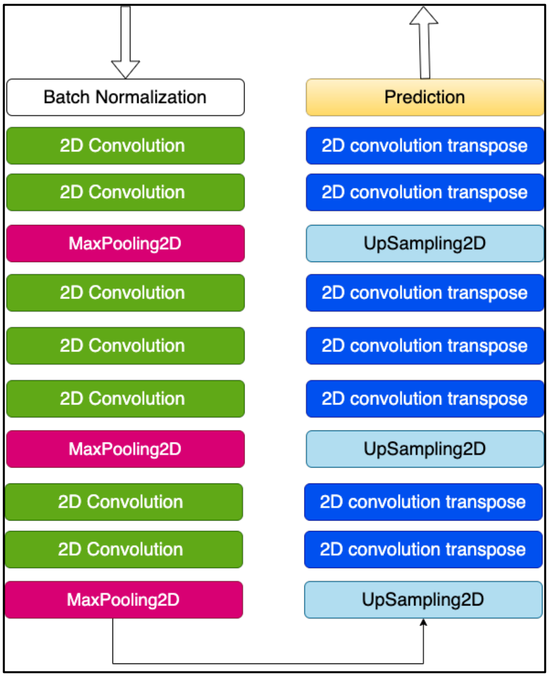


Fig. 2 CNN model layer specification

This thesis followed the LeNet-5 CNN architecture with some modification in layers to create the CNN model. The model created with batch normalization which consists of total 30 layers that includes 7 convolutional layers, 7 transpose convolutional layers, 3 max pooling layers, 3 2D up sampling layers and 10 dropout layers (showed in Fig 2). Each convolutional and transpose convolutional layer in the model uses RELU as its activation function, with one layer of padding. As this activation function is found to be faster and effective rather than other available activation functions. Other activation functions were tried, but RELU functioned more efficiently and quickly as expected. And assisted in avoiding model overfitting.

The model was initially built with batch normalization in each layer. However, it consumes more RAM than is necessary. This issue was resolved by simply utilizing it in the beginning. The first two convolutional layers consists of 8, 16 kernels with the size of 3 * 3 and one layer of padding with RELU activation function respectively. And then followed by one max pooling layer of 2*2 pool_size. Max pooling is used to gather the most data or elements from the region covered by the layers of filter. The primary objective of the max pool is to extract the most noticeable characteristics from the last result. The next 3 convolutional layers with kernel size of 16,32,32 with a drop layer of 0.2 drop. I.e., By zeroing off the neuron values, 20% of the neurons will be turned off at each training phase. Two convolutional layers with the kernel size of 64 are positioned between two max pooling layers. Following that, the deconvolutional layers are initiated with an up-sampling layer. Deconvolutional layers are used in the inverse order of convolutional layers. Following drop out layer, the flatten technique is used to turn the picture array into a single dimension. Finally, using RELU as the activation function, the two completely linked deconvolutional layers were used to generate the final output array. Following construction, the model was produced with Adam Optimizer, along with an accuracy measure and loss.

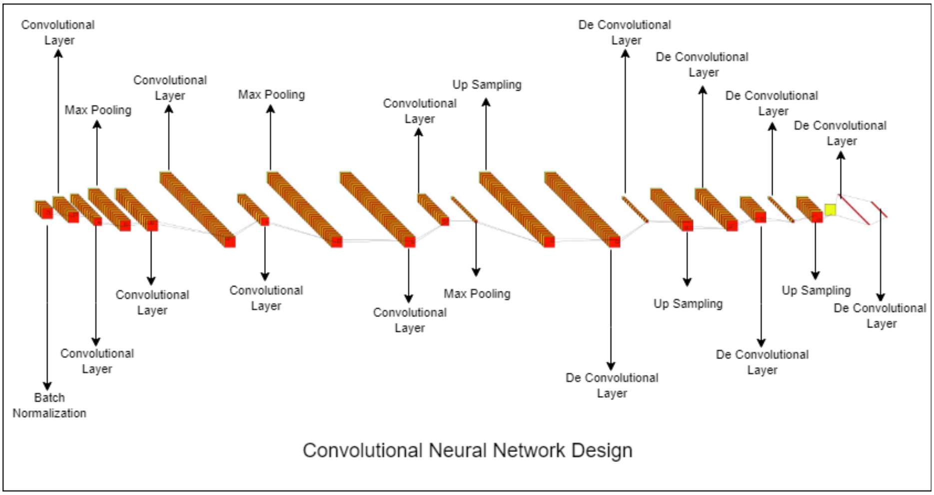


Fig. 2 Convolutional Neural network

3.5 CNN Model Training and Testing

The training of created model is carried out by specifying the optimizer as ADAM and loss as MSE (mean squared error). The use of the loss function MSE is done extremely well in this new model. The performance is far better than using any other loss function in model training. The model's performance was always good for diverse sorts of pictures, such as rotated images, either in vertical or horizontal axis, and channel changes or flips.

The model is trained by the function called model.fit() with different parameters like batch size and epochs. The model was initially trained for 25 epochs, resulting in overfitting. When the number of epochs was lowered to 10, the model became stable. The accuracy of the model is improved using different hyper parameter optimization techniques like change in epochs, learning rate, different activation functions, optimizers, batch size and number of layers. The model performs well when the batch size is set to 128, 10 epochs are used, and Adam is used as the optimizer.

3.6 CNN Model validation in real time

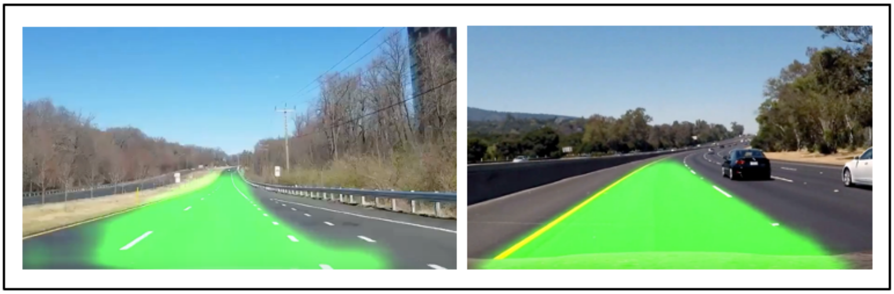


Fig. 4 shows real time testing before and after hyperparameter optimization

For real-time lane detection and recognition, a new collection of videos is examined to recognize the lane in the video. These videos are taken from the internet which shows various driving conditions such as curvy roads, hill areas, and so on. which are not used during the training and testing phases. The input video is processed as frame by frame of size 720* 1280 pixels to feed into the model for detection. The frame is then resized to 80*160*3 in order to generate smaller images for input into the trained model. Gray scaling of images is avoided because they tend to hide the yellow lanes on faded pavement portions. Since the detected lane portion is indicated in green, each frame is retuned as a green 'G' color channel. After predicting the lane area, the 'R' (red RGB) and 'B' (Blue RGB) are zeroed out to merge with the initial road image. The labels are normalized by dividing with 255, since the labels values are from 0 to 1 for Green 'G' pixel values. To know the good and bad prediction of the model, the new video is tested in the model before and after parameter tuning (Fig 4). During testing, omitting image augmentation produces better outcomes than applying it. Following prediction, each image frame will be merged into a video file as the output file.

4. Result

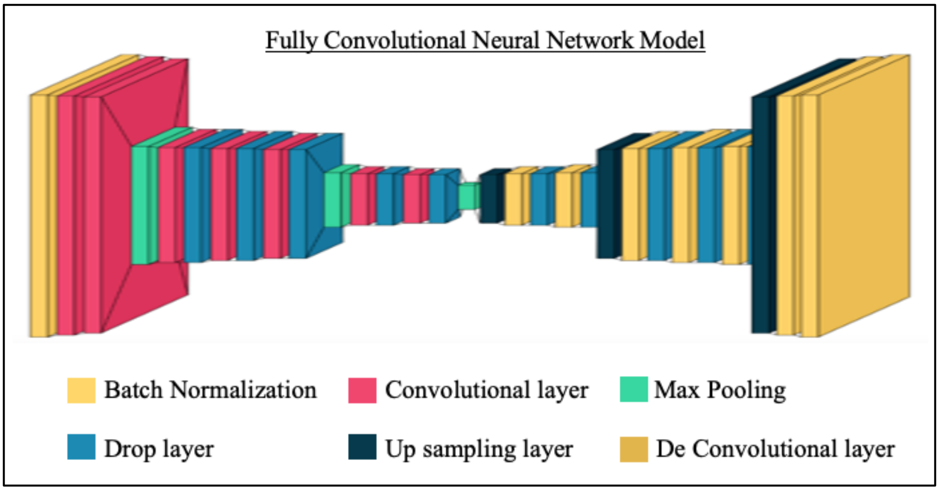


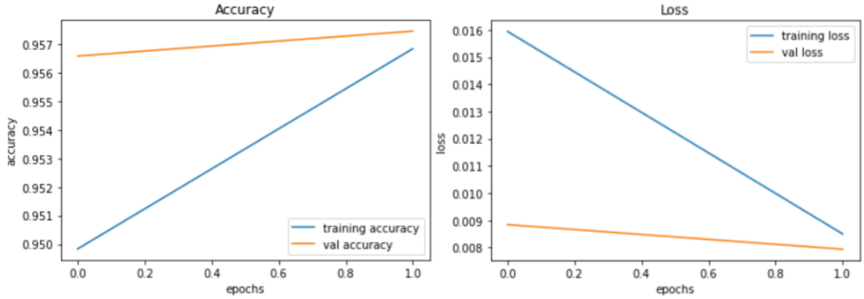
Fig. 5 Fully Convolutional Neural Network Model

The final topology of the fully convolutional neural network utilized in the thesis is seen in Figure 5. The final model is a combination of convolutional layers with a gradual reduction in pool size. When it hits the midway, the model changes to a reverse pooling approach with a De convolutional layer of identical size. The final layer comprises of one filter since the lane area has to be highlighted in green for easy visualization. The image will be returned in the color G channel.

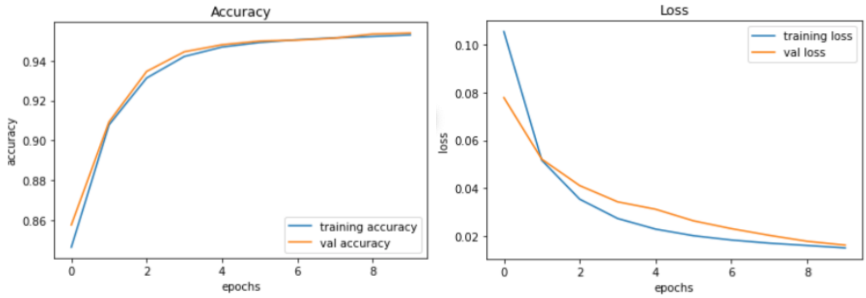
The findings acquired during hyper parameter optimization to determine the optimal combinations for excellent accuracy with correct prediction will be described in the next section. The outcomes of video input will be displayed as well.

Experiment Result 1: Model performance during hyperparameter optimization

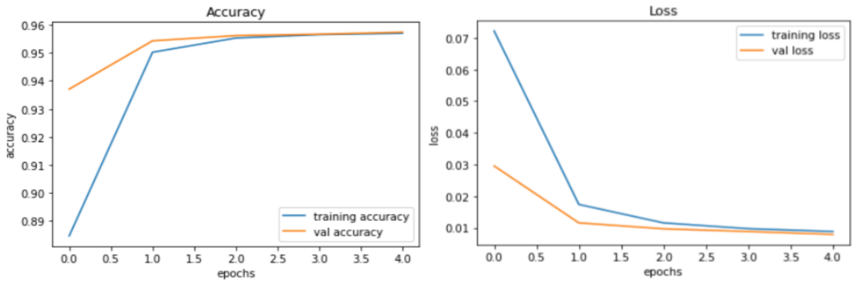
Hyperparameter optimization is essential to improve model accuracy while minimizing loss and avoiding overfitting. During the optimization step, several combinations of batch size, epochs, and learning rate are applied. At the end, accuracy and loss are computed for each combination. Every optimization leads to a different model accuracy and loss. For clarity, the graphs are drawn at the end of each optimization step (Fig 6).



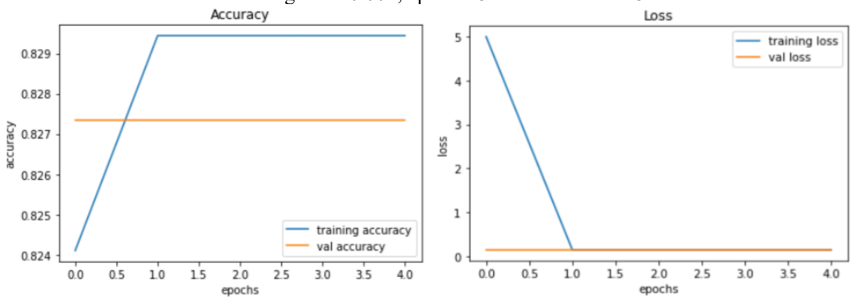
Learning rate: 0.001, epochs: 2 & batch size: 10



Learning rate: 0.0001, epochs: 10 & batch size: 128



Learning rate: 0.002, epochs: 5 & batch size: 128



Learning rate: 0.005, epochs: 5 & batch size: 128

Fig. 6 Graph plots for different hyperparameter combination.

Experiment Result 2: Model performance after hyperparameter optimization

After multiple optimizations, the model accuracy increased when the epoch, batch size, and learning rate were set to 10, 128, and 0.001 respectively. The image below (Fig 7) illustrates the relationship between train and validation data, demonstrating that the model achieved excellent accuracy with minimal loss. The model's accuracy after hyperparameter alteration is 95% without overfitting the result.

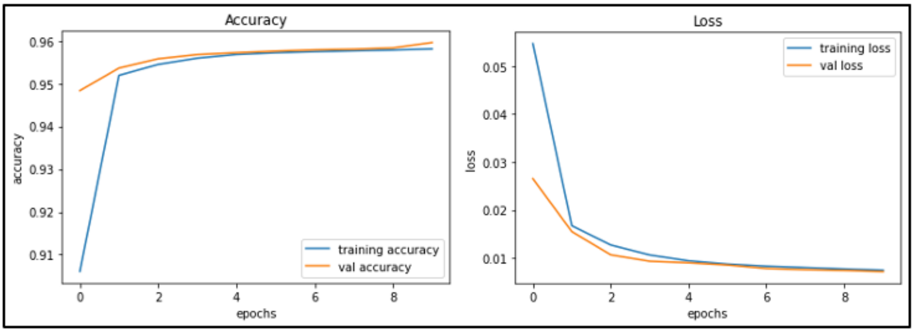


Fig. 7 Accuracy & loss graph plots for the final model.

Experiment Result 3: Model response towards real time video

Following training, the model is examined using real-time videos that were not used during training. To gain a better understanding of the accuracy of road lane prediction, various road conditions are fed into the trained model as input. The experimental results for each video input, such as low light, multiple lanes, curvature roads, and shadowed roads, are shown in Fig (8).

Fig. 8 Model response for real time video

When designing the neural network, the learning rate is the most essential hyperparameter **(Jason, 2019)**. When the learning rate value is increased from the ideal default value of 0.001, the model's accuracy drops **(Jason, 2019)**. An increase in the number of epochs results in great accuracy. **(Afaq, 2020)** However, increasing the epochs too much might result in overfitting. When there are insufficient epochs, resulting in inadequate accuracy will result underfitting. **(Afaq, 2020)**

5. Discussion

6. Conclusion

A lane recognition system for autonomous cars in difficult road conditions and dynamic situations was presented in this work. The suggested lane detection technique demonstrated apparent advantages in terms of accuracy and execution time consumption. The CNN model is built using the LeNet-5 CNN architecture. The model was created using batch normalization and has a total of 30 layers. Instead of using normal Convolution layers with forward pass, here the model is created with deconvolutional layers in combination of convolution layers, upsampling and maxpooling layers to achieve the back propagation technique in CNN. The dataset used for this research is a public dataset from Kaggle that contains images extracted from various road videos. Hyper parameter optimization is done to improve the model's accuracy in road lane prediction. According to the trial outcomes, the average road lane prediction accuracy achieved based on the road video is 95.96%. Thus, the model is capable to detect the lane areas from different videos under the different circumstances.

Here the thesis studied the problem of lane detection and suggested a new method to detect the road lane with high accuracy. Even after analyzing lane detection and testing the model under various conditions, there are still certain aspects that may be improved. For improvement in future, the model should be trained with even more large dataset to obtain better prediction. The dataset can also include the images that contains pedestrians and motorcycles to avoid collision on the road and thereby increase the safety. To advance, the model should be able to recognize the driving lane even when there are no markers on the road, and it should also be able to detect the lane even at high speeds.

Bibliography

Ataq, S., 2020. "Significance Of Epochs On Training A Neural Network". *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH ,* 9(6), pp. 277-616.
A, H. & A, K., 2009. "2D line filters for vision-based lane detection and tracking". *Proceedings of the 2009 International Workshop on Multidimensional (nD) Systems (nDS),* p. 1–5.
Alireza, B. & Routeh, S. S., 2017. "A novel PCA perspective mapping for robust lane detection in urban streets". *2017 Artificial Intelligence and Signal Processing Conference (AISP),* p. 145–150.
Bar, H., A, L., R, L. & G, R., 2014. "Recent progress in road and lane detection: a survey". *Machine vision and applications,* pp. 1-19,.
B, F., D, G., L, V. & P, H., 2015. "Autonomous Vehicle and Real Time Road Lanes Detection and Tracking". *2015 IEEE Vehicle Power and Propulsion Conference (VPPC),* Volume 10.1109, pp. 1-6.
Bhujbal, P. N. & Narote, S. P., 2015. “Lane departure warning system based on Hough transform and Euclidean distance”. *Proceedings of the 3rd International Conference on Image Information Processing, ICIIP 2015,,* p. 370–373.
B, Z. G., B, T. X., J, D. M. & D, G. Z., 2008. "Automatic detection technique of preceding lane and vehicle". *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL),* p. 1370–1375.
Feiniu, Y. et al., 2015. "Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis”. *IET Image Processing,* 9(10), p. 849–856.
Hojatolah, . A. & Babak, K., 2016. "New approach to road detection in challenging outdoor environment for autonomous vehicle". *2016 Artificial Intelligence and Robotics (IRANOPEN),* p. 7–11.
H, Y., U, Y. & K, S., 2013. "Gradient-enhancing conversion for illumination-robust lane detection". *IEEE Transactions on Intelligent Transportation Systems,* Volume 14, pp. 1083-1094.
J. Crayton, T. & Meier, B. M., 2017. "Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy". *Journal of Transport & Health,* Volume 6, p. 245–252.
Jason, B., 2019. *"Understand the Impact of Learning Rate on Neural Network Performance ".* [Online]
Available at: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/#:~:text=Specifically%2C%20the%20learning%20rate%20is,is%20adapted%20to%20the%20problem.
[Accessed 19 August 2022].
J, G., 2019. "An Overview of Recent Progress of Lane Detection for Autonomous Driving". *2019 6th International Conference on*

Dependable Systems and Their Applications (DSA), Volume 10, pp. 341-346.
J, S., H, Y., S, K. & K, S., 2015. "Real-time illumination invariant lane detection for lane departure warning system". *Expert Syst. Appl.,* Volume 42:1816–1824.
MD, I. O., K, S., A, K. & A, A. M., 2008. "Real time lane detection for autonomous vehicles. Computer and Communication Engineering,". *ICCCE 2008. International Conference,* Volume 10.1109/ICCCE, pp. 82-88.
Pei, C. W., Chin, Y. C. & Chang, H. L., 2014. "Lane-mark extraction for automobiles under complex conditions". *Pattern Recognition,* 47(8), p. 2756–2767.
P, W. C., Chang, C. Y. & H, C. L., 2014. "Lane-mark extraction for automobiles under complex conditions". *Pattern Recognition,* 47(8), pp. 2756-2767.
Rob, C., 2021. *pythonpool.* [Online]
Available at: https://www.pythonpool.com/cv2-normalize/
[Accessed 18 August 2022].
S. Zhou, Y. J. J. X. J. G. G. X. a. H. C., 2010. "A novel lane detection based on geometrical model and gabor filter". *Intelligent Vehicles Symposium (IV) 2010 IEEE,* pp. 59-64.
V, G. & S, L., 2015. “Lane Departure Identification for Advanced Driver Assistance,”. *IEEE Transactions on Intelligent Transportation Systems,* 16(2), p. 910–918.
Zelaya, C. V. G., 2019. "Towards Explaining the Effects of Data Preprocessing on Machine Learning". *2019 IEEE 35th International Conference on Data Engineering (ICDE),* pp. 2086-2090.