Lane Detection Using Hybrid RNN and CNN Approach

Ibrahim Ahmethan, P2699379

Abstract*— Lane detection in autonomous vehicles is one of the fundamental and crucial functions in terms of ADAS 'advanced driver assistance system', mainly we can classify lane detection methods into two main categories: deep learning based approach and conventional approach. the evolution of lane detection in autonomous vehicle started with conventional methods for lane detection by utilizing image processing techniques and handcrafted feature extraction, however conventional approach performs well only in specific driving scenes and it has no ability to be adapted to different driving scenes due to their fixed threshold values that is already embedded in the system and can not be changed or adapted to a new environment. In recent years many researches done in academia and industry in order to implement deep learning approach to achieve robust lane detection, deep learning approach is more reliable and adaptive in detecting lanes under different circumstances such as bad weather conditions, shadow, car occlusion and mark degradation, however most of the deep learning researches and approaches focus on detecting lane detection from one single frame "image" which is also can lead to unsatisfactory performance of lane detection under different challenging conditions. To overcome such a problem a hybrid deep neural network implemented by combining recurrent neural networks (RNN) and convolutional neural networks (CNN), this hybrid approach investigates the lane in more than one single frame which includes the previous frames of a driving scenes.

***Index Terms*— lane detection, hybrid method, RNN, LSTM, autonomus vehicle, image segmentation**

I. Introduction

THE leap in science, big data and increasing in computational power of hardware devices led to emergence of autonomous vehicles, lane detection is the backbone of autonomous vehicles once lane coordinates are calculated and obtained by a computer vision or any other method like an odometry, a vehicle will figure out its track and position in a lane and correct its position continuously in a lane without the need for human intervention. the purpose of such a technology is to ensure that autonomous driving offers safe, reliable and comfortable travel. In recent years evolution in artificial intelligence systems has encouraged industry and academia to allocate huge amounts of resources in the field of autonomous driving. In this paper a hybrid deep neural network (RNN+CNN) is utilized to achieve its objective and handle different challenging driving scenarios and scenes such as shadow, snowing, illumination variation, mark degradation and so on, meanwhile Many conventional approaches such as lane detection with Geometric modelling [2], and other supervised methods such as lane segmentation[6][9] have been published, however the common point among these proposed methods they limit the detection of lane only in a single current frame of driving scene, Thus the could perform poorly under different driving scenes and scenarios, in this case the lane might not be detected, partially detected of even false prediction of the lane might occur, the main reason behind this fluctuation in their performance is that the information obtained by a single one frame is not sufficient to predict the lane accurately Since a way / road consists of continuous lanes, while each lane is separated from the other line by a dashed or solid lines and the scene of driving is continuous, thus there will be high degree of correlation and overlapping among the current frame and multiple previous frames, by taking advantage of overlapping frames, in this proposed hybrid method, a lane of current frame can predicted by utilizing previous images "frames" of a continuous scenes.

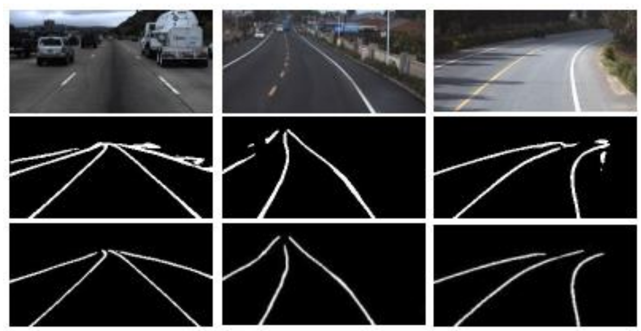


Figure 1. top line scenes of different driving situation middle line detected lane by using a single frame bottom line detected lane by using multiple previous frames

as shown in Figure 1 In the top row we can see challenging cases such as shadow and car occlusion that effects lane detection, the performance of the model is more robust in handling different situation while using multiple frames compared to one single frame. [3]

In this proposed hybrid method recurrent neural networks (RNN) and convolutional neural networks (CNN) is implemented as end-to-end neural networks (considering RNN + CNN as a single trained neural networks), since we have two types of neural networks CNN+RNN and both are functioning together as single neural network, each one of them will have a its own task, mainly CNN abstracts features from the input images or videos, the core task of CNN in this stage is to reduce the dimension of the raw input image by abstracting features that is necessary for lane detection that is acquired during the training the network, for example if the raw image is 420 x 1280 roughly we will have about half a million features of vectors which is not tolerable to use these amount of vectors in the further stages of RNN and it is computationally very expensive, so CNN here reduces the dimension to maintain tolerable operations in the network[4][5]

II. Literature Review

In the last decade using deep learning for many different application became very popular, especially in object detection and recognition, lane detection is one of interesting topic in terms of self driving cars and autonomous vehicle, variety of researches have been done in this field, one of them is published in IEEE 2021 (A Deep Learning Approach for Lane Detection) where CNN was implemented as feature extractor (same thing in my mini project), then a post processing implemnted to detect and analyze the road markings in order to perfrom curve fitting, mainly it consists of main two block as the following.

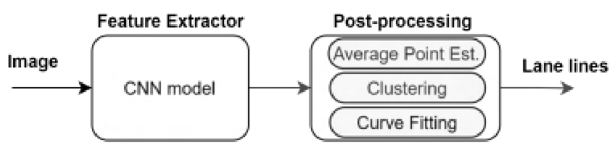


Figure 2. Diagrama of mentioned deep learning approach

Another interesting and enlightening research (Learning Lightweight Lane Detection CNNs by Self Attention Distillation), in this research the fundamental principle is using Self Attention Distillation in order to make a model spontaneously learning by itself without the need to extra labeling or supervision.[6]

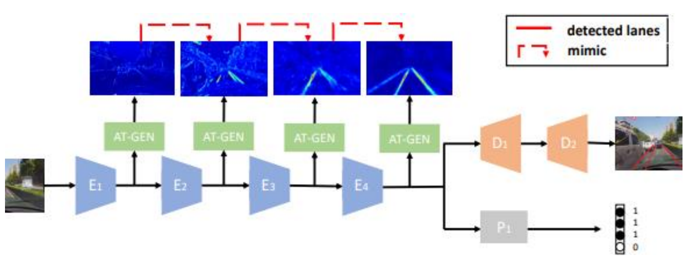


Figure 3. the main block of SAD that mimics the detected lines to enhance learning process

III. Proposed Method

In this part of the mini project, we will take a look at the hybrid neural network (the combination of convolutional and recurrent neural networks), their network architecture and working principle to achieve the prediction of lane detection task with minimum possible errors.

A. overview of the system

In this proposed method to perform lane detection, mainly we have two type of deep neural networks:

1- Deep convolutional neural network (CNN)

2- Deep recurrent neural network (RNN)

The purpose of CNN is to take the input image or video and perform feature abstraction and extraction on each single image or frame, the idea behind applying CNN as feature extractor is minimize the dimension of the input while maintaining and considering the required information to detect a lane, the importance of minimizing the dimension is crucial step to ensure our network performs appropriately.
For example the dimension of the images used to train the network in this mini project is 256 x 128 which roughly about 3.3k that means our dimension vector or feature is about 3.3k, thus by applying CNN the dimension will be reduced to tolerable levels to enhance the performance of the network and make it computationally less expensive and more reliable

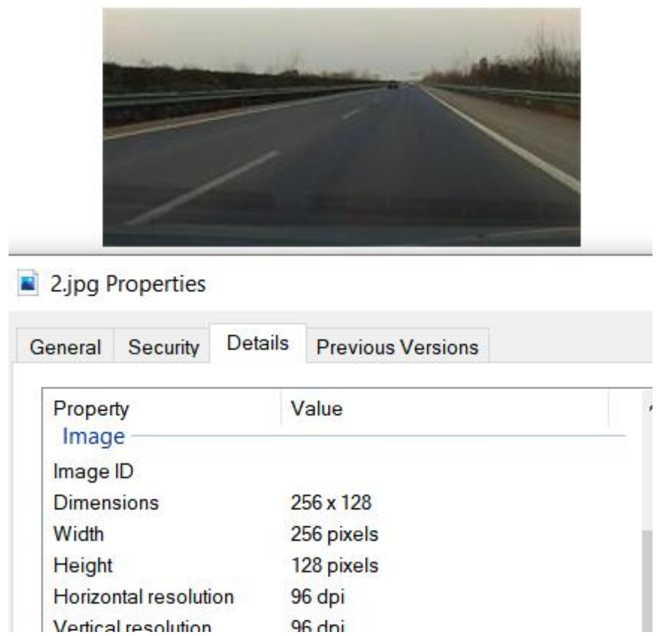


Figure 4. Example of an image used in during the training of the neural network and its dimension

On the other hand since driving scenes are continuous, it is considered as time – series process since the frames of a scene is captured from a video are sequential with respect to time, here emerges the importance of RNN.
RNN is proficient in predicting time - series problems, as a remainder we mentioned already that CNN processes the input images and then RNN predicts current image by analyzing multiple pervious images or frames, explicitly RNN processes input frame recursively by dividing the input into successive parts to build a fully connected layers, meanwhile the type of the RNN used in this mini project is long short memory LSTM

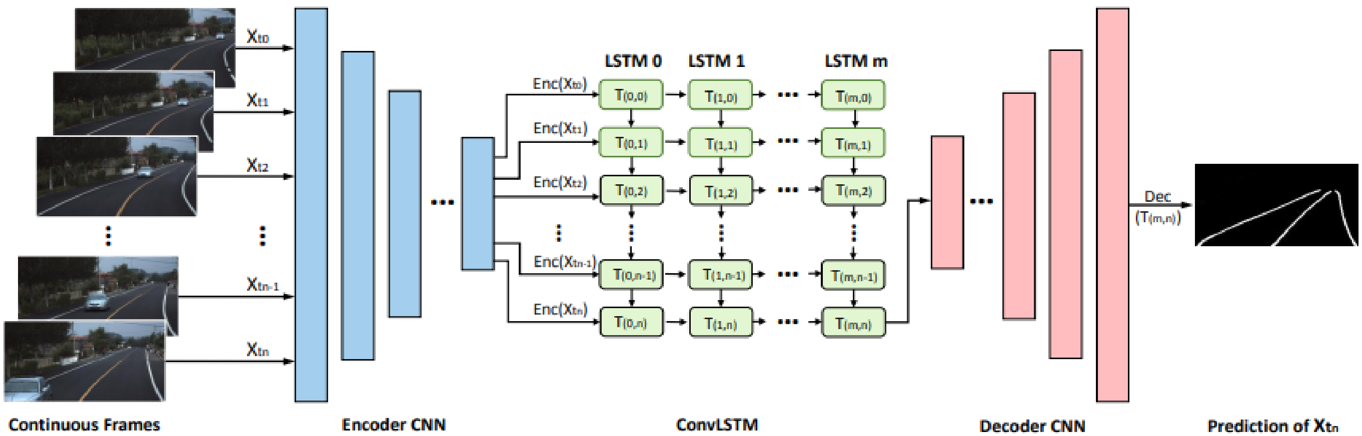


Figure 5. Network architecture of the model

B. Design of the network

The purpose of the proposed method is to take many previous images as input, then predict the current lane by semantic segmentation method, to achieve semantic segmentation a fully convolution (FCN) [smt]network is introduced within the CNN, where many other recursive operations such as pooling and convolution is carried out by the CNN to generate smaller size of feature maps obtained from continuous scenes which is time series problem, that latter will be fed RNN to be processed to predict lane features, the RNN/LSTM generates new frature maps to feed the decoder part of the CNN, operation such is deconvolution and upsampling will performed in the decoder part in order to reconstruct the image into its original size, since the images size was reduced in the encoder part, hence the size of the input image/frame and the output image/frame will be in the same size. meanwhile the architecture of CNN has two main block encoder and decoder (as shown in figure 6)
(both encoder and decoder are fully convolutional network FCN), the encoder of the CNN abstracts the features to be processed in the RNN, as shown in figure 5 the CNN has encoder and decoder blocks and the RNN (LSTM) is embedded between the encoder and decoder of the CNN to process time-series driving scenes.
the encoder and decoder was designed in order to integrate the CNN and RNN as a end to end neural network[7].

Moreover, as shown in figure 5 the RNN(LSTM) is embedded in the network between the encoder and the decoder, LSTM block fed by the encoder, the LSTM has the ability to remember important and forgetting unnecessary features, the task of each cell in LSTM is to check and estimate whether the received features from the decoder is necessary or not.
Convolutional long short memory (LSTM) is utilized instead of conventional fully connected LSTM to reduce to reduce computation power and time, since convolutional long short term memory utilizes convolution process in every cell instead of using matrix multiplication, as shown in the RNN block of figure 5 we have ConvLSTM.
In this mini project two different network designs are used in the LSTM, the first one is SegNet-ConvLSTM and the second one is Unet-ConvLSTM, each single of them has different input and output size, where the input and output sizes depends on the size of the feature map that is generated by the encoder, for SegNet-ConvLSTM the input/output size is 4 x 8 while in the case of Unet-ConvLSTM the input/output size is 8 x 16 and each network convolutional kernel size of 3 x 3 while the one hidden layer with dimension of 64, one hidden layer with dimension of 128, one hidden layer with dimension of 256 and at the end of the network we have two hidden layers with the dimension of 512 as shown in Figure 7 that summarizes the network designs of both SegNet and UNet ConvLSTM.

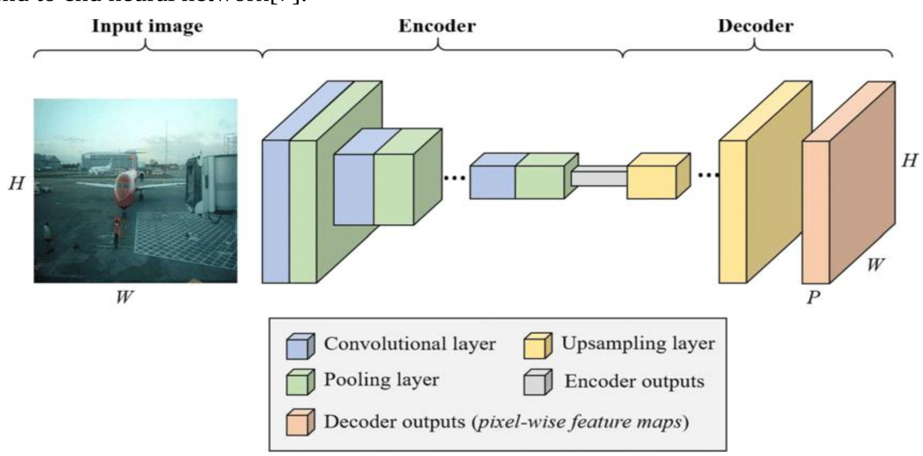
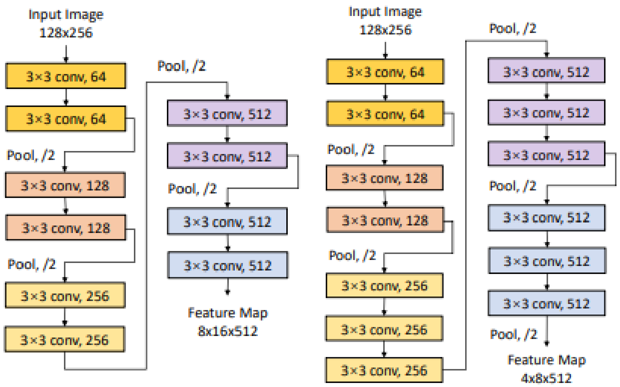


Figure 6. Basic structure of the CNN with encoder - decoder parts



(a) *(b)*

Figure 7. (a) network design of UNet-ConvLSTM and (b) network design of SegNet-ConvLSTM

the fundamentals of ConvLSTM network design is determine the number of convolutional kernels, layers and their sizes.

C. Training phase

One the network is constructed, then it could be trained to predict and detect lane, two different designs of LSTM are built for detection, while the models are trained by using TuSimple lane dataset, which contains more than 3000 packages images of different roads, high ways and rural ways in order to work with comprehensive dataset, where each package contains 20 of continuous image of the same scene and some images are labeled for the purpose of training as show in figure 8. for efficient train, different optimizers chose during different phases of the training, initially started with Adam optimizer, however since Adam optimizer falls in the local minima easily, thus to overcome this problem, training switched to SGD optimizer is after achieving high accuracy with Adam optimizer in order to find the global minima[8][9].

Figure 8. Labeled image from TuSimple dataset

Figure 9.. input image with its result

IV. Results and Conclusion

Two evaluation methods was considered in this proposed method

1- visual evaluation
2- metric evaluation

visual evaluation is the basic method to evaluate a system by observing the results visually, results of visual examination was satisfying as shown in figure 9, meanwhile evaluation metrics is was performed to measure the results in a scientific way, the following evaluation metrics was applied to the Unet-ConvLSTM model.

1- Accuracy, where the formula as the following

$$\mathbf{Accuracy} = \frac{\mathbf{True\ Positive + True\ Negative}}{\mathbf{Total\ Number\ of\ Pixels}}$$

the accuracy in the trained Unet-ConvLSTM model was **<u>97.872%</u>**

2- F1 score, where the formula as the following

$$\mathbf{F1} = \mathbf{2} \cdot \frac{\mathbf{Precision \cdot Recall}}{\mathbf{Precision + Recall}}$$

The F1 score in the trained Unet-ConvLSTM model was **<u>85.1%</u>**

3- Precision, where the formula as the following

$$\mathbf{Precision} = \frac{\mathbf{True\ Positive}}{\mathbf{True\ Positive + False\ Positive}}$$

the precision in the trained model Unet-ConvLSTM was **<u>78.85%</u>**

4- Recall, where the formula as the following

$$\mathbf{Recall} = \frac{\mathbf{True\ Positive}}{\mathbf{True\ Positive + False\ Negative}}$$

The recall in the trained model Unet-ConvLSTM was **<u>93.0%</u>**

As a conclusion, even though the precision is relatively low compared to the other evaluation metrics, the results are satisfying since the test of the model is done on a large dataset including variety of high ways, rural roads and other challenging situations.

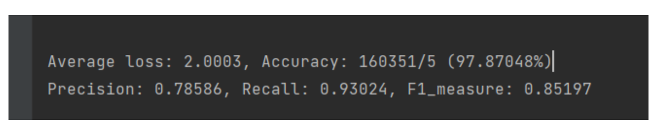


Figure 10. the results of UNet-ConvLSTM obtained after training and testing the model in Python programing language

V. LIMITATIONS

it is obvious that hybrid deep learning performs better under different situations than the conventional method, however there are no perfect system that performs perfectly without any additional drawbacks, every system has its limitation, pros and cons, one of the main problems that I faced while I was working in this mini project was during the training phase, hardware demanding to train a very large dataset, it took me 18 hours to train the UNet-Cnn LSTM model, meanwhile during the training phase my PC was operating at its highest capacity and performance in terms of GPU and RAM, my RAM is 12 GB and my GPU is NVIDIA GEFORCE 940 mx 2 GB model had to run excessively for many hours to train the model with relatively high temperature, my PC during the training hours was not available even to browse an internet page due to the heavy computational power of training the neural network, in the following figure 12 screenshot of the task manager shown.

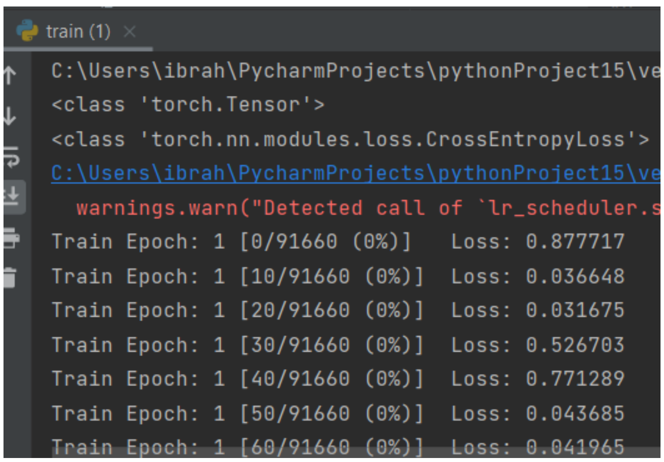


Figure 11. screenshot early stages of the training process

I had to reduce the batch size due to my limitation in the GPU and that caused a very long training time

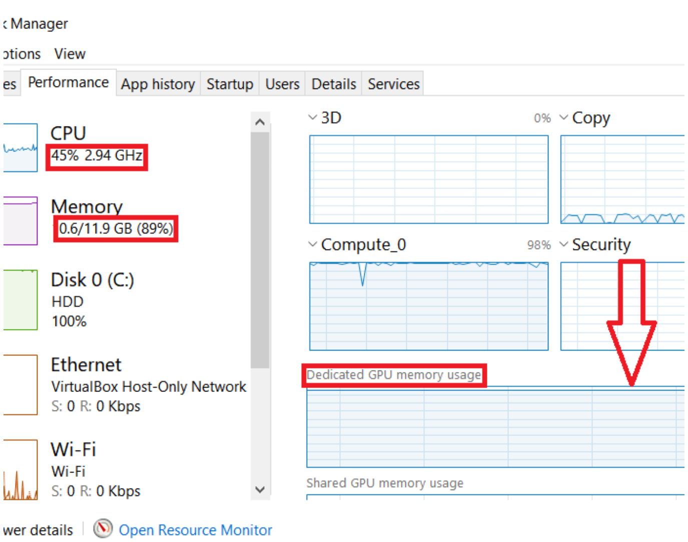


Figure 12.my PC's CPU, RAM and GPU perfprmance during the training phase

Another limitations deep learning working with large dataset even though the dataset is labeled and ready to use I had to pre-process the dataset to adjust the directories of more than 70,000 image including training and testing images, this preprocessing was done by the help microsoft excel, handling very large dataset requires patient and the ability to visualize the dataset. Another limitation is that runing the model on real time is

ACKNOWLEDGMENT

I would thank and express my deepest gratitude for Dr. Ioannnis kypraios and Professor Aladdin ayesh for being a source of motivation for me.

REFERENCES

[2] Zhu, H.; Yuen, K.V.; Mihaylova, L.; Leung, H. Overview of environment perception for intelligent vehicles. IEEE Trans. Intell. Transp. Syst. 2017, 18, 2584–2601

[3] L. Li, J. Song, F.-Y. Wang, W. Niehsen, and N.-N. Zheng, “IVS 05: new developments and research trends for intelligent vehicles,” IEEE Intelligent Systems, vol. 20, no. 4, pp. 10–14, 2005.

[4] Zou Q, Jiang H, Dai Q, Yue Y, Chen L and Wang Q, Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks, IEEE Transactions on Vehicular Technology, 2019.

[5] Zhou, Y.; Wang, G.; Xu, G.Q.; Fu, G.Q. Safety driving assistance system design in intelligent vehicles. In Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO), BalT. Getahun, A. Karimoddini and P. Mudalige, "A Deep Learning Approach for Lane Detection," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1527-1532, doi: 10.1109/ITSC48978.2021.9564965.

[6] J. Wu, H. Cui and N. Dahnoun, "Gradient Map Based Lane Detection Using CNN and RNN," *2020 22th International Conference on Digital Signal Processing and its Applications (DSPA)*, 2020, pp. 1-5, doi: 10.1109/DSPA48919.2020.9213294.

[7] Zou Q, Jiang H, Dai Q, Yue Y, Chen L and Wang Q, Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks, IEEE Transactions on Vehicular Technology, 2019.

[8] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. L. Price, S. Cohen, and T. S. Huang, “Youtube-vos: Sequence-to-sequence video object segmentation,” in European Conference on Computer Vision (ECCV), 2018.

[9] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, “Towards end-to-end lane detection: an instance segmentation approach,” CoRR, vol. abs/1802.05591, 2018.