# Road Lane Detection Using Deep Convolutional Neural Networks

Shivakumar Valpady
University of Massachusetts, Amherst
svalpady@umass.edu

Anmol Jawali Mallikarjuna
University of Massachusetts, Amherst
ajawalimalli@umass.edu

## Abstract

*Most of this work is a re implementation of work referenced [19]. Lane detection is an important feature for autonomous vehicles and advanced driver support systems in driving scenes. For autonomous vehicles, lane identification is extremely important. Many sophisticated lane detection methods have been proposed in recent years. Many approaches use lane boundary information to locate the vehicle on the street. Convolutional Neural Networks (CNNs) are, like many other computer vision-based activities, the state-of-the- art software for defining lane boundaries. Some techniques, however, concentrate on detecting the lane from a single image and often lead to unsatisfactory results when coping with some extremely bad situations such as heavy shadow, severe mark degradation, serious vehicle occlusion, etc.*

*As we know, lanes are continuous line structures on the road. The lanes that cannot be predicted accurately in one current frame may potentially be inferred out by gathering information from previous frames. Hence, lane detection by using multiple frames of a continuous driving scene is proposed by using a hybrid deep architecture of combination of CNN and Recurrent Neural Network (RNN). The idea is to extract features of continuous images using CNNs and these features of multiple frames, holding the properties of time-series, are then fed into RNN block for feature learning and lane prediction. To increase the accuracy of the obtained model, smoothing techniques are implemented. TuSimple lane detection dataset is used for training and testing.*

*However, most of the work is re implemented, there are some additional implementations suggested in the referenced paper are done in this paper. The future work suggested in our base paper is implemented in this paper by adding pre-processing steps to the images. Gaussian filters and Maximum filters were used for this step.*

*Index terms - Convolutional Neural Networks, LSTM, lane detection, semantic segmentation, Canny Edge detection.*

## 1. INTRODUCTION

With the huge development of high-precision optical sensors and electronic sensors, high-accurate computer vision and machine learning algorithms, our understanding of the real-time driving scene has become increasingly practical. Among various aspects of autonomous driving cars, lane detection is the basic and most important one. The vehicle will know where to go once the lane positions are obtained, thus avoiding the risk of running into other lanes[15]. There have been a number of methods proposed with sophisticated performance as reported in the literatures. These include lane detection with geometric models [16],[3], some formulate with energy minimization problems[17],[10], some also include deep learning based techniques[5][12]and some segment the lane by using supervised learning strategies [19][9] and so on.

Most of the mentioned approaches restrict their solutions by detecting road lanes from one current driving scene frame, resulting in low performance in handling difficult driving scenarios such as heavy shadows, extreme road mark loss, severe vehicle occlusion, as shown in top three images in Figure 1. In such cases, the lane can be predicted or projected in the wrong direction, or it can be partly detected, or it cannot even be detected. The important reason for that is, the information provided by the current frame is not sufficient.

Generally, the road lanes are rectangular structures and continuous on the pavement, appearing either in solid or in a dash. Considering that the driving scenes are continuous and mostly identical between two adjacent frames, the location of the lanes in the adjacent frames is highly related. More specifically, it is possible to predict the lane in the current frame using multiple previous frames, although the lane can suffer damage or deterioration caused by shadows, stains and occlusion. This is the main motivation for us to study lane detection by using images of a continuous driving scene.

A hybrid deep neural network is proposed for lane detection by using multiple continuous driving scene images, combining Deep Convolutional Neural Networks (DCNN)

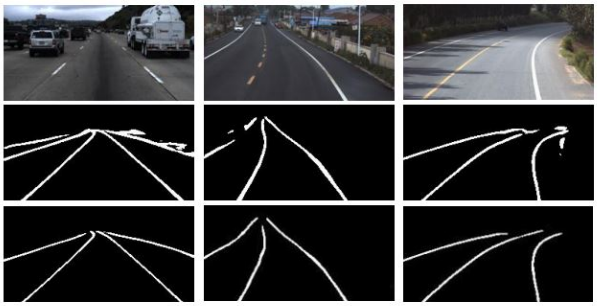


Figure 1. Lane detection in challenging situations. Top row: three example images of different driving scenes. Middle row: lane detection using only the current frame. Bottom row: lane detection using four previous frames and the current frame with the proposed method[19].

and Deep Recurrent Neural Network (DRNN). From a global perspective, the proposed network is a DCNN that takes multiple frames as an input and forecasts the current frame's path in a semantic segmentation manner. A fully convolution DCNN architecture is presented to achieve the segmentation goal. It contains an encoder network and a decoder network, which guarantees that the final output map has the same size as the input image. From a local perspective, DCNN's encoder network abstracted features are further interpreted by a DRNN. A long short-term memory (LSTM) network is employed to handle the time-series of encoded features.The DRNN output should have fused the continuous input frame information and is fed into the DCNN decoder network to help predict the routes.

The main contributions of this paper lie in two-fold:

- First, with regard to the issue of not being able to accurately detect a lane using a single image in the situation of darkness, road mask loss and vehicle occlusion, a novel approach is proposed that uses continuous driving scene images to detect a lane.Since more information can be extracted from multiple continuous images than from a single current image, the proposed method can more accurately predict the route compared to single-image methods, especially when dealing with the above-mentioned challenges.

- For implementing the hybrid network of DRNN with DCNN, a novel fusion strategy is presented, where DCNN[12] consists of an encoder and decoder with fully-convolution layers, and the DRNN is implemented as an LSTM network. The DCNN extracts features of images and LSTM takes each features map as a full-connection layer in time line and recursively predicts the lane. LSTM is used as it predictcs the information effectively, and significantly improve the performance of lane detection in the semantic segmentation framework.

The datasets used mainly is TuSimple dataset, which is expanded by labelling more frames.The remainder of this paper is structured as follows. Section II discusses the research associated with it. Section III describes the emerging deep neural hybrid network, including deep neural convolution, deep recurrent neural network, and training strategies. The experiments and findings are recorded in Section IV. Section V ends our work and briefly addresses potential future research.

## 2. BACKGROUND/RELATED WORK

Number of researches have been made in the field of lane detection and prediction. These methods can be classified into two categories: Geometric modelling and deep learning based methods.

### 2.1. Geometric Modelling

Most of the Geometric modelling follow two-step step solution, with edge detection and then line fitting [3] [1] [11]. Various types of gradient filters are employed for edge detection. In [1], edge fetaures of lanes in bird-view imags were calculated with gaussian filter, whereas in [11] and [18], Steerable filter and Gabor filter are investigated for lane-edge detection, respectively. Aparrt from gradient, color and texture were also studied for lane detection.

For line fitting, Hough Transformation (HT) models were exploited. Curve-line fitting can either be used as a post-processing tool for HT tests or as a direct replacement for HT. In addition, the geometry modeling-based lane detection can also be implemented with stereo visions [4] [6], in which the distance of the lane can be estimated.

### 2.2. Deep Learning based methods

Deep learning based methods are exploited in almost every major field of research, there are four major types of deep learning techniques employed for lane detection research. These include the Encoder – Decoder CNN model, Fully connected network, CNN + RNN network and GAN models.

Semantic segmentation is one of the most important tasks in the field of computer vision and is the main task of encoder-decoder models [2]. LaneNet which was built on the SegNet architecture has two decoders, the first decoder segments the image to detect lanes with a binary mask while the other decoder segments the road. The feature map then obtained is clustered and curve-fitted to produce the resulting image [19].

Most commonly used for lane detection are fully convolutional (FCN) neural networks with optimized algorithms. CNN with complete layer connectivity was used to detect the lane edges using rudimentary decoders, and RANSAC was applied to refine the results [19]. CNN with fully connecting layers. When implemented in an extreme-learning framework, this model demonstrated improved performance.

In the CNN RNN models, a road image is first divided into various continuous slices. The extractor (CNN) is then used to extract each slice. Finally, the lane in the picture slice charts has been retrieved by a recurrent neural network. It is claimed that this approach is better than using CNN alone. In this process, however, the RNN can only model time series feature in the one image, and two separated blocks are the RNN and CNN [19].

Also used for lane detection is the generative adversarial network (GAN), consisting of a generator and a discriminator [8]. The advantage of this model is that the lanes are small and compact, so that large soft boundaries that are typically formed by CNNs are not marked.

The model implemented in this project looks like a time series problem, compared with the above mentioned methods, and a lane detection for a variety of continuous frames instead of one frame is done. CNN and RNN are incorporated into the architecture and end-to-end trainable model to achieve robust, more consistent performance and under more difficult conditions.

# 3. PROPOSED METHOD

In this section, we introduce a novel hybrid neutral network by fusing DCNN and DRNN together to accomplish the lane detection task. Nonetheless, because of their unsatisfactory performance under challenging conditions such as heavy shadow, extreme mark loss and serious vehicle occlusion, these models are not promising to be applied to realistic ADAS systems. It is found that the information in a single image is inadequate to enable effective lane detection.

## 3.1. System Overview

Lanes on the pavement are solid- or dash-line structures that can be identified in a single image from a geometric-modeling or semantic segmentation mode.

The proposed method combines the CNN and RNN for Lane detection, with number of continuous frames of driving scene. RNN is adaptive for Lane detection and prediction task due to its talent in continuous signal processing, sequential feature extracting and integrating. And CNN is mainly for processing large images. By multiple layers of convolution operations and max pooling, an input image can be abstracted as feature maps in smaller size. These feature maps obtained on continuous frames hold the property of time-series, and can be well handled by an RNN block.

In order to integrate CNN and RNN as an end-to-end trainable network , we construct the network in an encoder decoder framework. . The architecture of the proposed network is shown in Figure 2. The CNN encoder and CNN decoder are two networks that are entirely convolutionary. The encoder CNN processes each of them with a number of continuous frames as an input and gets a time series of feature maps. Then the feature maps were input into the LSTM network for lane-information prediction. The output of LSTM is them fed into the decoder CNN to produce a probability map for lane prediction. The lane probability map has the same size of the input image.

## 3.2. Network Design

*1) LSTM network:* The RNN block in the proposed network accepts feature map extracted on each frame by the encoder CNN as input. Specifically, LSTM network is employed, which generally outperforms the traditional RNN model with its ability in forgetting unimportant information and remembering the essential features. A double layer LSTM is implemented, one layer for extraction of the sequential function and the other layer for integration. Hence, Convolutional LSTM (ConvLSTM) [13] is utilized in the proposed network. The ConvLSTM replaces the matrix multiplication in every gate of LSTM with convolution operation, which is mainly used in end-to-end training and feature extraction from time-series data.

The activations of a general ConvLSTM cell at time t can be formulated as:

$$C_t = f_t o C_t - 1 + i_t o \tanh(W_x c * X_t + W_h c * H_t - 1 + b_c)$$
$$f_t = \sigma(W_x f * X_t + W_h f * W_t - 1 + W_c f o C_t - 1 + b_f)$$
$$o_t = \sigma(W_x o * X_t + W_h o * W_t - 1 + W_c o o C_t - 1 + b_o)$$
$$i_t = \sigma(W_x i * X_t + W_h i * W_t - 1 + W_c i o C_t - 1 + b_i)$$
$$H_t = o_t o \tanh(C_t)$$

where Xt denotes the input feature maps extracted by the encoder CNN at time t. $C_t$, $H_t$ and $C_t 1$ , $H_t 1$ denote the memory and output activations at time $t$ and $t-1$, respectively. $C_t$, $i_t$, $f_t$ and $o_t$ denote the cell, input, forget and output gates, respectively. $W_x i$ is the weight matrix of the input $X_t$ to the input gate, $b_i$ is the bias of the input gate. The meaning of other W and b can be inferred from the above rule. σ(·) represents the sigmoid operation and tanh(·) represents the hyperbolic tangent non-linearities. '$*$' and '$o$' denote the convolutio operation and the Hadamard product, respectively.

*2) Encoder-Decoder Network:* The encoder- decoder framework models lane detection as a semantic-segmentation task. To make it an end-to-end framework, the output size of the encoder-decoder has to be the same size of input. In encoder part, convolution and pooling are used for image abstraction and feature extraction. While in the decoder part, deconvolution and upsampling are used to

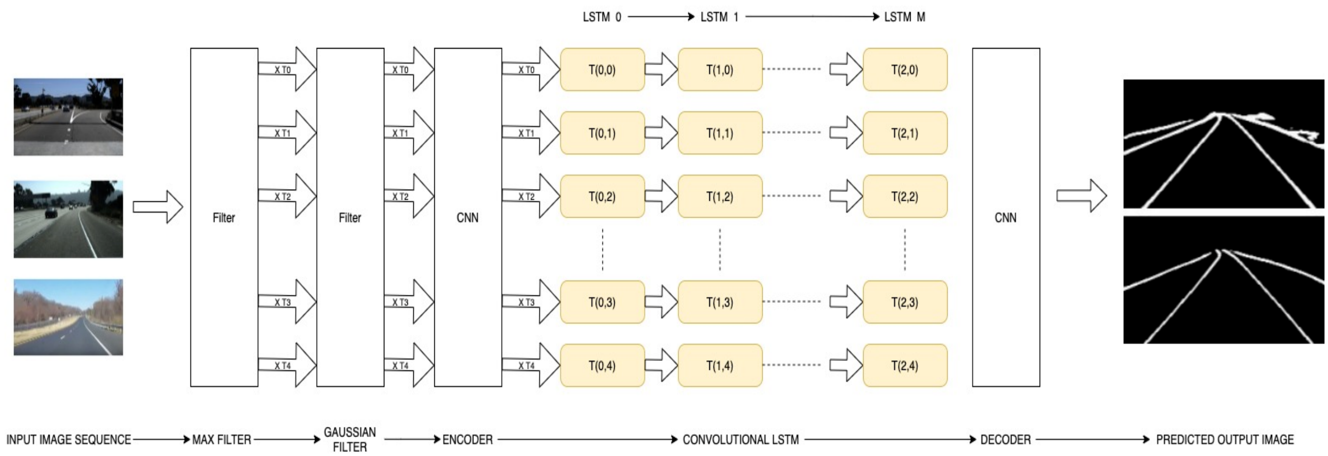



Figure 2. Architecture of the proposed network [19].

grasp and highlight the information of targets and spatially reconstruct them.

Inspired by the performance of SegNet [2] and U-Net encoder-decoder architectures in semantic segmentation, we develop our network by integrating the ConvLSTM block into these two encoder-decoder networks. The resulting networks are called respectively as SegNet-ConvLSTM and UNetConvLSTM. The encoder block and decoder block are fully convolutional networks. Thus, figuring out the number of convolutional layers and the size and number of convolutional kernels is the core of architecture design. VGGNet [14]'s 16-layer convolution-pooling architecture is used as its encoder in SegNet. By referring to SegNet, we design and refine the encoder by changing the number of convolutionary kernels and the overall conv-pooling layers. Thus we can get a balance between the accuracy and the efficiency. The encoder architectures are illustrated in Figure 3.

In the Seg-Net, an encoder network block contains two convolution layers compared to the last block with twice the number of convolution kernels and a pooling layer is used to downsample the feature map. After this operation, the size of the feature map will be reduced to half (on side length) while the number of channels will be doubled, which reflects semantic features of high level.The last block, as shown in Figure 3, does not double the number of kernels for the convolution layers in the proposed SegNetConvLSTM. There are two explanations for doing so. First, even with less sources, information in the original image can be well reflected. Typically, the lanes can be represented with primitives like color and edge that can be well extracted from the feature map and abstracted. Second, feature maps produced by the encoder network will be fed into ConvLSTM for sequential feature learning. Parameters of a full-connection layer will be quadrupled as the side length of the feature map is reduced to half while the channels remain unchanged, which is easier to be processed by ConvLSTM.

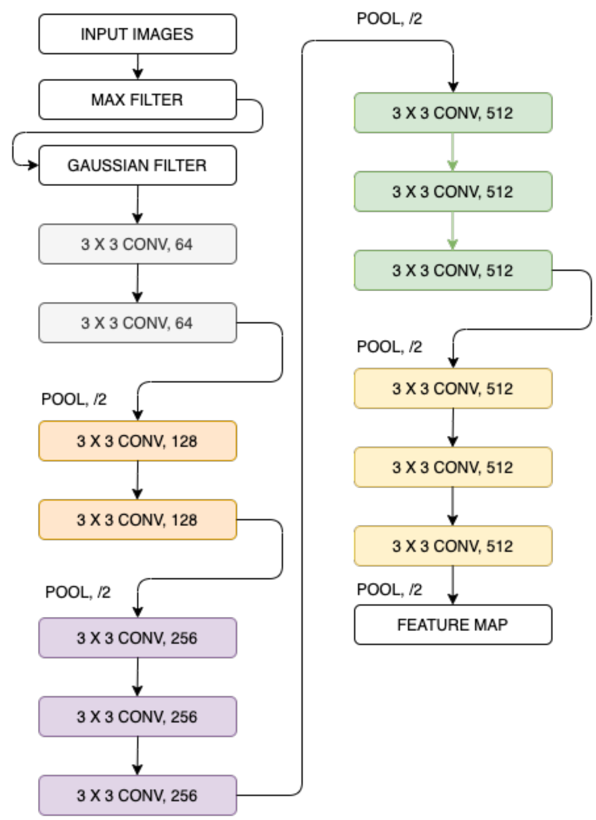



Figure 3. Encoder network in SegNet-ConvLSTM.[19]

The size and number of feature maps in the CNN decoder should be the same as their counterparts in the CNN encoder, while being positioned in the opposite direction to help recover functionality. Accordingly, the up-sampling and convolution in each sub-block of decoder match the corresponding operations in the sub-block of the encoder.

Within U-Net, the appending function map is conducted in a straightforward manner between the encoder and decoder's corresponding sub-blocks. The deconvolution in the decoder is performed while in SegNet using the indices reported in the encoder.The encoder summarizes the input information that is fed into the decoder for retrieval of target information.

### 3.3. Training Procedure

Once the end-to-end trainable neural network is designed, a back-propagation method may train the network to predict ground truth by updating the weight parameters of the convolutionary kernels and the ConvLSTM matrix. The following five aspects are considered in our training.

1. The original images are smoothened by applying the max filter with size 6. The resultant image is smoothened furthur by applying Gaussian filter having $\sigma$ = 1.5. The image noise is reduced and the lanes are highlighted by doing this method. This process is done to increase the accuracy of the model.

2. The proposed network uses the pre-trained weights of SegNet on ImageNet[7], a large classification benchmark dataset. Initializing with the weights pre-trained will not only save the training time, but will also pass appropriate weights to the proposed networks.

3. A set of N continuous driving scene images are used to define the lanes as data. So, the coefficient for each weight update for ConvLSTM should be divided by N in the back propagation. We set N=5 for comparison in our experiments. We also investigate experimentally how N affects performance.

4. A loss function is constructed based on the weighted cross entropy to solve the discriminative segmentation task, which can be formulated as,
   $E_loss = \sum_{x\in\Omega} w((\mathbf{x}) \log(p_{l(}(\mathbf{x})_{)}(\mathbf{x}))$
   where ‘ : 1, . . . , K is the true label of each pixel and w : R is a weight for each class, on purpose of balancing the lane class. It is set as a ratio between the number of pixels in the two classes in the whole training set.

5. To train the network efficiently, different optimizers were used at different stages. In the beginning, adaptive moment estimation (Adam) optimizer, which has higher gradient descending rate but is easy to fall in local minima was used, which resulted in turbulent learning. To avoid this, when the network is trained to a higher accuracy, we used stochastic gradient descent (SGD), which has smaller stride and hence finds the global minimum. The learning rate while changing the optimizer is matched accordingly with respect to the optimizers used.

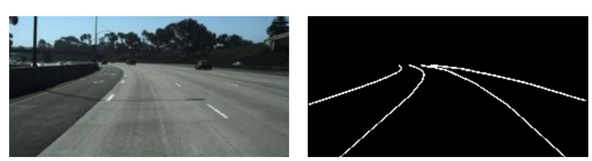
## 4. EXPERIMENTS AND RESULTS

In this section, experiments are conducted to verify the accuracy and robustness of the proposed method. The performances of the proposed networks are evaluated in different scenes and are compared with diverse lane-detection methods. The influence of parameters is also analyzed.

### 4.1. Datasets

We create a dataset based on the dataset of the TuSimple lane dataset. The data set for the TuSimple lane includes a number of 3,626 image sequences. These images are the front of the expressways. There are 20 continuous frames captured in one second in each series. The last frame, that is, the 20th picture, is labeled with lane ground truth for each sequence. To augment the dataset, we additionally labeled every 13th image in each sequence. For training, we sample 5 continuous images as an input to train the proposed network and identify lanes in the last frame. Based on the 13th and 20th frames ground truth mark, we can construct the training set.

Rotation, flip and crop operations are implemented for data augmentation, and a number of 19,096 sequences are generated, including a total of 38,192 labeled images for training. The input will be changed randomly into different lighting conditions, which will lead to a more accurate dataset. A unifying norm should be developed to define these lanes in ground truth due to the variety of lane types, e.g. stitch-shaped lanes, single white lane and double amber lanes. Thin lines are used for annotating the lanes. However, a lane looks like tobe more wide in a close view while more thinner in a far view. In the experiments, we analyze the image of the scene in lower resolution, provided that when images are smaller, lanes will become thinner with a width nearly one pixel. An Example is shown in Figure 4. By using a lower resolution, the model can also be shielded from being affected when complex textures in the background around the vanishing point.



(a) (b)

Figure 4. : An example of an input image and the labeled groundtruth lanes. (a) The input image. (b) The ground truth.

## 4.2. Implementation Details

For the experiments, the images for lane detection are sampled to a resolution of 256 x 128. The experiments for lane detection are implemented. The experiments are implemented on a single 16 GB RAM GeForce GTX TITAN X GPU. It took approximately 6 hrs for 40 epochs for training. The training loss vs the number of epochs graph is shown in Figure 5.

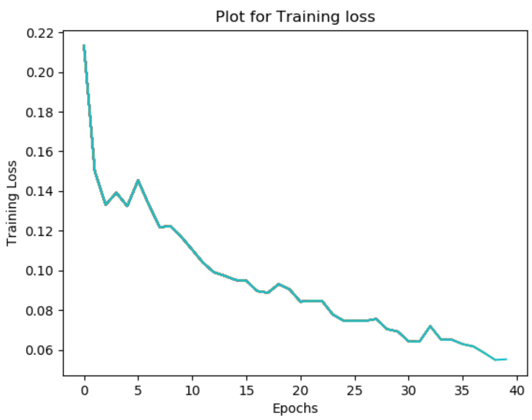



Figure 5. : Training loss vs Epochs

## 4.3. Qualitative Analysis

The Figure 6, Figure 7, Figure 8 and Figure 9 show the output images of Lane Detection using the implemented proposed model.

## 4.4. Quantitative Analysis

Through quantitative tests, we analyze the superior performance of the proposed method. The simplest measurement criterion is accuracy, which tests the overall performance of the classification based on correctly categorized pixels.

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

| Accuracy | |
|---|---|
| Gaussian Filtered Input | Original Input |
| 93.48 | 91.84 |

Table 1. Accuracy Table

The test set is evaluated in two different ways. First by applying gaussian filtering to the images and smoothening the image and again by without applying the filter. We see that the accuracy for the smoothened image is higher than the normal image.

By using the ConvLSTM between encoder and decoder for sequential feature learning, the accuracy grows upto 1.5 percent for SegNet. Although the proposed models have achieved higher accuracy and previous work [47] also adopts accuracy as a main performance matric, we do not think it is a fair measure for our lane detection. Because lane detection is an imbalance binary classification task, where pixels stand for the lane are far less than that stands for the background and the ratio between them is generally lower than 1/50. If we classify all pixels as background, the accuracy is also about 93 percent. Thus, the accuracy can only be seen as a reference index.

Precision and recall are employed as two metrics for a more fair and reasonable comparison, which are defined as :

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

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

In lane detection task, we set lane as positive class and background as negative class. From the Precision and Recall equations, true positive means the number of lane pixels that are correctly predicted as lanes, false positive represents the number of background pixels that are wrongly predicted as lanes and false negative means the number of lane pixels that are wrongly predicted as background. After adding ConvLSTM, there is a significant increase to the normal CNN-RNN model by 6 percent for SegNet model for precision and recal also imploves slightly.

| Precision | |
|---|---|
| Gaussian Filtered Input | Original Input |
| 0.4755 | 0.4262 |

Table 2. Accuracy Table

| Recall | |
|---|---|
| Gaussian Filtered Input | Original Input |
| 0.8636 | 0.8085 |

Table 3. Accuracy Table

Considering the precision or recall only reflect an aspect of the performance of lane detection, we introduce F1-measure as a whole matric for the evaluation. The F1 is defined as:

$$\textbf{F1} = 2 * \frac{\textbf{Precision * Recall}}{\textbf{Precision + Recall}}$$

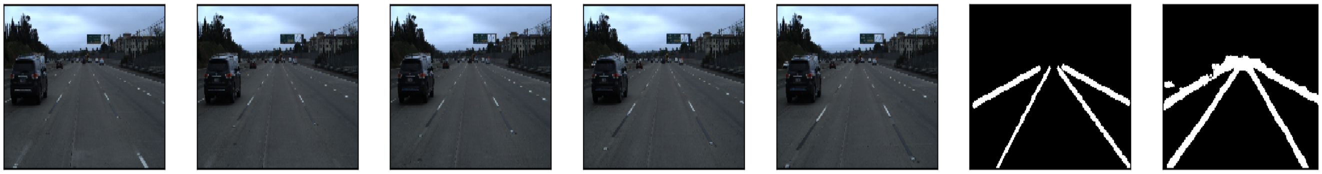


Figure 6. Output for a test image 1

Figure 7. Output for a test image 2

| F1 Scores | |
|---|---|
| Gaussian Filtered Input | Original Input |
| 0.5991 | 0.5406 |

Table 4. Accuracy Table

## 5. CONCLUSION

In this paper, a novel hybrid neural network combining CNN and RNN was proposed for robust lane detection in driving scenes. The proposed network architecture was based on an encoder-decoder system that takes as an input multiple continuous frames and predicts the current frame's lane in a semantic segmentation way. In this sense, a CNN encoder first abstracted features on each frame of the input. Instead, a ConvLSTM processed the sequential encoded features of all input frames. Ultimately, for information reconstruction and lane prediction, ConvLSTM outputs were fed into the CNN decoder.

Compared with baseline architectures which use one single image as input, the proposed architecture achieved significantly better results, which verified the effectiveness of using multiple continuous frames as input. And we also observed that smoothening the image with maximum filter and gaussian filter produced better than training and testing with the original images.Compared to the other model, the proposed model showed higher performance with relatively higher precision, recall, and accuracy values.

Working on this project was a huge learning curve with regard to concepts like Deep Convolutional Neural Networks, ConvLSTMs, Encoder-Decoder Models.

In the future, we plan to incorporate Lane classification for this model, with the TuSimple Lane classification dataset to distinguish different types of lanes with different colors which can be very helpful for autonomous driving cars. Also, the model can be improved by training for larger epochs and for a larger dataset. The accuracies can be evaluated and tested for different preprocessing methods like HLS techniques, and other type of smoothening filters.

The Code implemented for this project is available in the given link below.
https://github.com/shivavalpady/Computer-Vision-CS670-.git

## References


[1] M. Aly. Real time detection of lane markers in urban streets. *2008 IEEE Intelligent Vehicles Symposium*, Jun 2008.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017.

[3] A. Borkar, M. Hayes, and M. Smith. Polar randomized hough transform for lane detection using loose constraints of parallel lines. pages 1037–1040, 05 2011.

[4] C. Caraffi, S. Cattani, and P. Grisleri. Off-road path and obstacle detection using decision networks and stereo vision. *Trans. Intell. Transport. Sys.*, 8(4):607–618, Dec. 2007.

[5] P.-R. Chen, S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin. Efficient road lane marking detection with deep learning, 2018.

[6] R. Danescu and S. Nedevschi. Probabilistic lane tracking in difficult road scenarios using stereovision. *Intelligent Transportation Systems, IEEE Transactions on*, 10:272 – 282, 07 2009.

[7] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

Figure 8. Output for a test image 3

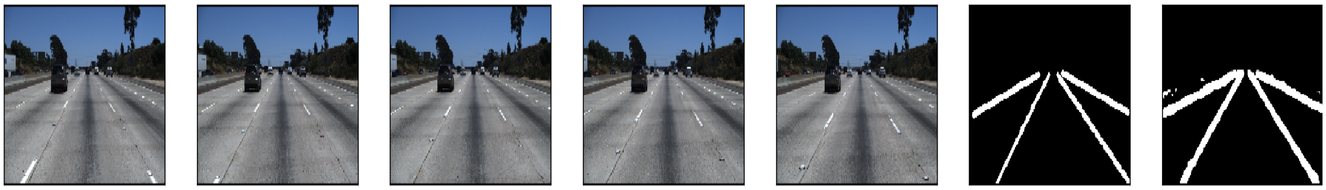


Figure 9. Output for a test image 4


[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[9] Y. Huang, S. Chen, Y. Chen, Z. Jian, and N. Zheng. Spatial-temproal based lane detection using deep learning. In L. Iliadis, I. Maglogiannis, and V. Plagianakos, editors, *Artificial Intelligence Applications and Innovations*, pages 143–154, Cham, 2018. Springer International Publishing.

[10] J. Hur, S.-N. Kang, and S.-W. Seo. Multi-lane detection in urban driving environments using conditional random fields. *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1297–1302, 2013.

[11] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, March 2006.

[12] F. Pizzati, M. Allodi, A. Barrera, and F. García. Lane detection and classification using cascaded cnns, 2019.

[13] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[15] W. Wang, D. Zhao, J. Xi, and W. Han. A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Transactions on Vehicular Technology*, PP, 07 2018.

[16] Y. Wang, E. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision Computing*, 22:269–280, 04 2004.

[17] C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, pages 733–747, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[18] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong, and H. Chen. A novel lane detection based on geometrical model and gabor filter. *2010 IEEE Intelligent Vehicles Symposium*, pages 59–64, 2010.

[19] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang. Robust lane detection from continuous driving scenes using deep neural networks, 2019.