REAL TIME ROAD LANE DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORK

[1] R Shreyas, [2] Sai Karthik P K, [3] R Ajay, [4] Prof. Karthik S A
[1][2][3]*BE Students, Department of Information Science and Engineering*
[4] *Asst. Professor, Department of Information Science and Engineering*
[1][2][3][4] *Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India*

ABSTRACT

Detection of lanes is an essential module for autonomous vehicles and advanced driver assistance systems (ADAS). Many state of the art methods for lane detection have been suggested in recent years. Although, these techniques focus on identifying the lane from a single frame, and they usually provide arguably dissatisfying performance in dealing with certain extreme situations such as degradation of the lane line, large shadows, significant occlusion of vehicles, noisy inputs of images, etc. Practically, lanes are supposed to be on-road continuous line structures. Hence, a lane that cannot be precisely detected in the live frame can be extrapolated from the information of previous frames. Therefore, we have used multiple frames from a continuous driving scenario to approach lane detection, and for this reason, a hybrid architecture- combination of a convolution neural network (CNN) and a recurrent neural networks (RNN). In an attempt to train the model for optimum robustness, we intend to perform comprehensive experiments on two massive datasets.

Key Terms: *DCNN, LSTM, lane detection, autonomous driving.*

1. INTRODUCTION

Our understanding of real-time driving scenarios has become increasingly realistic, given the substantial developments in high-precision optical sensors and electronic sensors, high-precision computer vision and effective machine learning algorithms. Amidst various other features of autonomous driving vehicles, road lane detection is the principal and most significant one. The vehicle will realise where to move when the position of the lanes are obtained, thereby avoiding the risks of overstepping into other lanes. As reported in the relevant works, there are a number of modern approaches presented with smooth and sophisticated performance. They include lane detection with geometric models [16],[18], some of which include such techniques focused on deep machine learning[5],[12]. Some also map out issues related to energy minimization [17] and some use certain supervised learning strategies [19] and so on to segment the road lane. Most of the above mentioned methodologies restrict their results by detecting road lanes from a single, current driving scenario frame and result in poor performance while handling extreme driving scenes such as large shadows, substantial road lane line degradation, and significant vehicle occlusion, as depicted in top three images in Figure 1.

Considering these cases, the lane could possibly be predicted with inaccuracy or projected in the wrong direction, it can be partly detected, or it cannot even be detected. The important reason for this being, the knowledge presented by the current image frame is not nearly sufficient [19]. Considering that driving scenarios are continuous and usually identical or nearly identical between two to three immediate frames, the position of the lane lines in the next few immediate frames is nearly identical and related. By using several previous frames, it is possible to predict the location of the lane in the current frame, although the lane lines can suffer deterioration or degradation due to weathering effects and weather conditions, shadows, and occlusions due to inadequate lighting. This is the principal motivation for our team to use a series of continuous images from a driving scenario to approach lane detection.

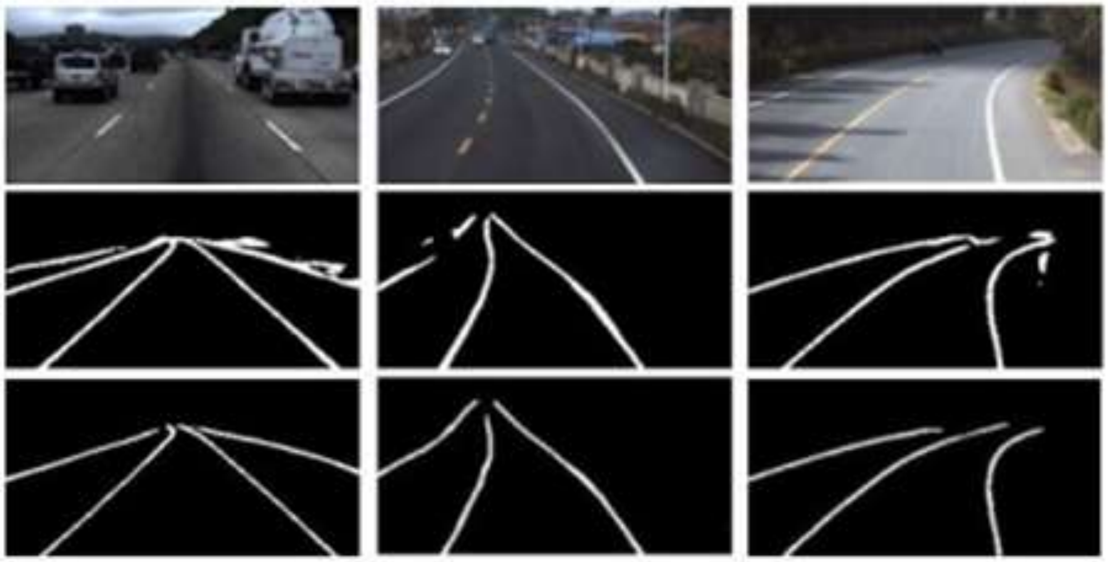

Fig -1: Lane detection in challenging situations. Middle row: Lane detection using only current frames Bottom row: lane detection using current and four previous frames as proposed by our method

2. SURVEY STUDY

Muhammad Ali Aydin et cetera all[8], presented a method where image processing techniques are used for determining lanes. The following methods are used: Image processing technique- Here we convert each video frame into HSV colour-space which differentiates various colours more accurately. Lane line determination- we use Sobel filter, Hough transform and canny edge detection for finding lanes on the roads. Find Region Of Interest (ROI): we determine ROI using skyline and various other methods. Horizon line is used to determine the required region of interest. This the line where sky and earth intersect when we see it from our naked eyes. The intersection lines and lane lines are merged to form ROI.

Thanda Aung et cetera all[6], proposed the following methods to warm drivers when they are going out of their lane. The methods used are: Smoothing: It is a process where the noise is eliminated using filters like 2d FRI filter to obtain a binary edge map and finally hough transform is used to detect edges and boundaries. Detecting line: we use hough transform for detecting lane lines which are subject to scenarios like short brakes in the lane which is caused due to noise and various other parameters. Tracking lane: we calculate distance between the lanes to find out the road width that a vehicle follows through its journey.

Fahmizal et cetera all[7], proposed a method where road lanes are detected using CNN. Here, yolo is used to implement DCNN which has two connected layers and 23 convolutional layers for detecting objects. Road-Lane Detecting: four methods are used- Warping: here, images are handled by changing perspective of input. Filtering: we filter the lane colours with non-lane lines and we pick only the range of yellow and white colours using LUV and LAB format. Detection: The non-zero values are used from the process which was done earlier to detect lanes. We also crop images to 15 sub-images and aggregate them to left and right images. De-warping: we do exactly in opposite way as we did in warping resulting in clear images.

Rama Sai Mamidala et cetera all[9], proposed “Dyanamic Approach for Lane Detection Using Google Street view and CNN” where a novel approach is being proposed for lane detection using CNN which is on the basis of SegNet decoder architecture.This architectures’s main feature is the usage of max-pooling indices in the decoders for up-sampling the low resolution feature maps which helps in retaining the frequency details of the segmented images and decreases the total number of training parameters in the decoders. To enable real time navigation, the author integrates Google APIs with the SegNet architecture. This interface helps in providing assistance for the Robotic systems.

VGPNet approach by Robotics and Computer Vision Lab(KAIST),[10]: Here, the scientists proposed a unified end to end trainable multitask network which was able to handle lane and road marking detection during extreme weather conditions and even the vanishing points of the lane. The network consists of four task modules in which each task performs a separate task: regression of the grid-box, object tracking, multi-label identification, vanishing point estimation. This technique indicated that it could detect generic marking shapes and attempt to match lines or splines to locate lanes. Classification is solved under a set of scenarios to check the effectiveness of the approach with advanced deep learning of tasks such as segmentation and classification. The accuracy is high in real-time at 20 fps.

Chanho Lee cetera all[3], proposed this method for lane detection: Lane marking detection: Triangular ROI is determined and markings of lanes are searched and then the colour of the markings are changed to grey scale, various other edge detection are performed to obtain performance and efficiency. Here the segments of the lane are used to filter using various slope filters to find the segments of line and slope values for the marked points. Lane marking Tracking: In subsequent frames we use Hough transform and canny edge detector for surviving the line components in noisy environment. The proposed method helps in both efficiency and performance in various factors and scenarios. This method removes various edges and noisy elements decreasing horizontal lines and depth of the road which are mainly affected by speeds of the car.

3. METHODOLOGY

In this method, using a sequence of continuous driving scenario images, combining Deep Convolution Neural Networks (DCNN) and Recurrent Neural Network (RNN), we present a hybrid deep neural network for road lane detection (RNN). The proposed model comprises of a DCNN from a wider perspective that incorporates different sequential images as a feedback and predicts the lane path in a svm classification manner in the current frame. In order to attain this segmentation objective, a fully deep convolution (DCNN) approach is used. It consists of a network of encoders and also a network of decoders, assuring that the final feature map is exactly the same size as the source images. From a local point of view, the encoder network’s summarised features of a Deep CNN are further explicated by a RNN. Following this, to handle the time series of encoded features, a long short-term memory (LSTM) network is used. The DRNN output should fuse the continuous input frame information and then be loaded into the network of DCNN decoders to assist in forecasting road lane routes.
We plan to build the network in the form of an encoder network -decoder network model in an effort to integrate CNN and RNN as an complete, end-to-end training network. The network architecture that is proposed is shown in Fig. 2. Both the encoder and decoder CNN are fully convolution networks. The encoder-CNN processes each of the frames with a sequence of continuous frames as input, and derives a time series of feature vectors. The feature vectors are then given to the LSTM network as sources for the lane-information prediction. In order to generate a probability distribution for the lane prediction, the LSTM output is then loaded into the decoder-CNN. The probability vector for the lane is the same scale as the source images.

3.1 Convolutional Neural Networks (CNN)

CNN is a kind of Deep Neural Network composed of several hidden layers, such as the RELU layer, the convolutionary layer, a layer for pooling and a fully related optimized layer. The weights within the convolutionary layer are primarily shared by CNN thereby reducing its memory requirements and increasing network performance. With the 3D capacities of its neurons, localized connections, and relative weights, the significant features of CNN contemplated. The convolutionl layer generates a feature vector via the convolution of various sub regions of the source images with a learned kernel. Then, via the ReLulayer, a non-linear activation function is applied to boost the convergence rate when the error is minimal. A section of the frame or feature network is selected in the pooling layer, and the pixels with the highest value between them or mean value is also preferred because the reference pixels are reduced to one scalar value by a 3x3 or 2x2 grid. This leads to an outsized decline in the sample size. In accordance with the convolution layers towards the output level, the conventional Fully-Connected (FC) layer is often used. Usually, two kinds of processes are done by the pooling layer: max pooling and mean pooling. The typical neighbourhood is calculated within the features extracted in the case of mean pooling, and within a maximum of features extracted in the case of max pooling. Mean pooling limits errors caused by the limited size of the neighbourhood and retains background info.

3.2 LSTM Network

The RNN unit in the presented network architecture considers feature vectors produced by the encoder-CNN over each image as the feedback for modelling the sequence of continuous images of driving scenarios as a time series. Various types of RNN models, such as LSTM and GRU, have been suggested to tackle the variable time-series results. An LSTM network is used in this model, which typically outclasses the conventional RNN architecture with its ability to forget insignificant details and recall only the critical characteristics by using network cells to determine whether or not a segment of information is essential. With the first unit for simultaneous feature extraction and the second for assimilation, a dual-layer LSTM is implemented.
The activations of a general ConvLSTM cell at time t can be formulated as:

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * W_{t-1} + W_{cf} \circ C_{t-1} + b_f)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * W_{t-1} + W_{co} \circ C_{t-1} + b_o)$$

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * W_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

$$H_t = o_t \circ \tanh(C_t)$$

where X_t denotes the input feature maps extracted by the encoder CNN at time t. C_t, H_t and C_{t1} , H_{t1} denote the memory and output activations at time t and $t - 1$, respetively. C_t, i_t, f_t and o_t denote the cell, input, forget and output gates, respectively. W_{xi} 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. $\sigma(\cdot)$ represents the sigmoid operation and $\tanh(\cdot)$ represents the hyperbolic tangent non-linearities. '∗' and 'o' denote the convolutio operation and the Hadamard product, respectively.

3.3 Deep Learning

It is composed of several layers of nonlinear nodes, combine computer file with a collection of weights so that assigning significance to inputs for the corresponding task the algorithm is attempting to be told in supervised and/or unsupervised behavior. The sum of the product of that input and weights is passed through the activation function of nodes. Each layer's output is fed synchronously as a feedback to the successive layer ranging from its input nodes. Major categories of descriptions corresponding to multiple points of abstraction are often used for learning.

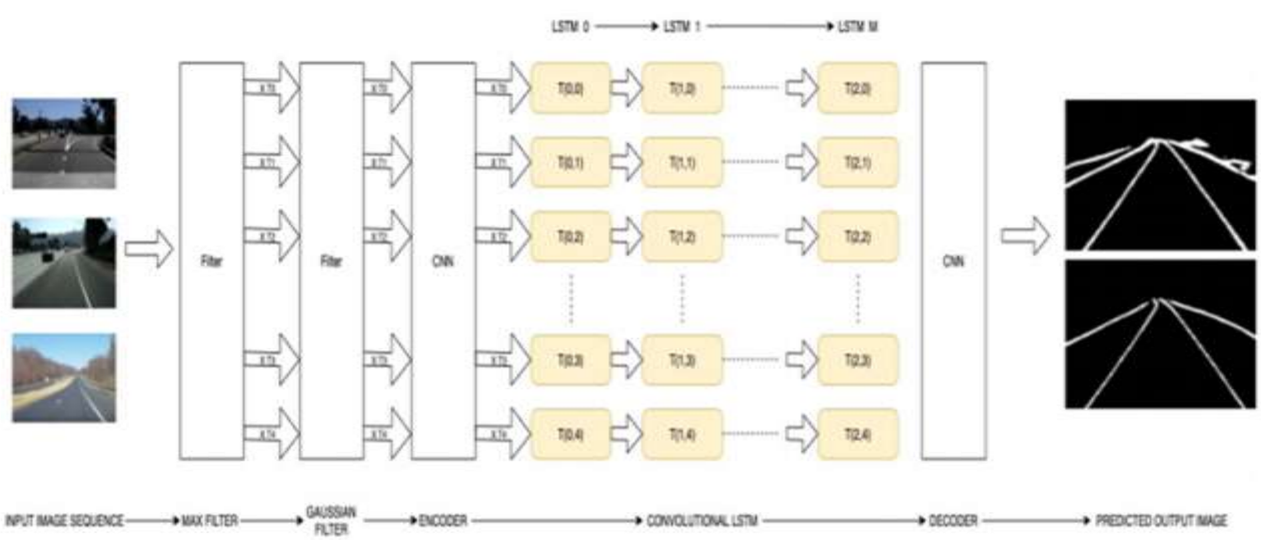

Fig -2: Architecture of the proposed network. [19]

3.4 Dataset

The dataset we plan to use the lane scenes provided by TuSimple and CULane Datasets. The TuSimple lane dataset comprises of 6,408 image sequences. These images are the front of the expressways in the United States of America. There are 20 continuous image sequences captured in one second in each series. The final image, that is, the 20th picture, is labelled with lane ground truth for each sequence. Additionally, keeping in mind the previously discussed challenges posed by Indian road lanes, we plan to create our own data set consisting of at least 20-30 image sequences, captured from a mounted camera on a car that is driven along various Indian road scenes covering a wide range of road conditions.

3.5 Flow Diagram

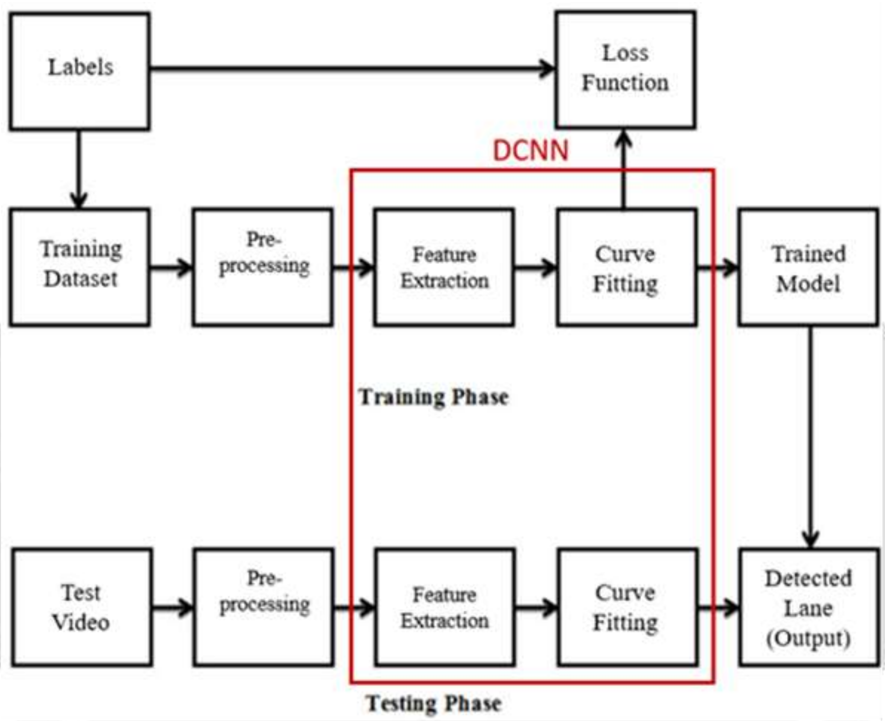

Fig -3: Flow Diagram for the proposed model.

4. RESULT COMPARISON

The below tables provide a comparison between the models proposed by various authors. It describes the accuracy, precision and recall values achieved by each of these proposed models. Thus, giving us a better overview of the results obtained from the mentioned methodologies.
Recent Studies on Road Lane Detection using CNN.

Authors	Model	Result
Shivakumar et al	D-CNN	Accuracy = 91.84 | Precision = 0.4262 | Recall = 0.8085
Jiyong et al	SCNN and ConvGRUs	Accuracy = 97.98 | Precision = 0.8674 | Recall = 0.9562
Xianpeng et al	Dual View –CNN (DVCNN)	Precision = 95.49 | Recall = 92.80
Rama Sai et al	CNN with Google Street View	Accuracy = 0.9610 | Precision = 0.9889 | Recall = 0.8732
Ze Wang et al	CNN with LSTM	TPR = 87.3% | FPR = 7.7%

Wei Wang et al	CNN with instance segmentation	ACU = 0.9785
Seokju Lee et al	VPGNet	Detection Score = 0.77

Table 1: Comparison of the results of the papers.

5. RESULTS

The CNN model is trained with both assorted driving scenes from TuSimple dataset and from Tvt dataset. The CNN model is trained with 3000 various driving scene images at 640x480 Resolution and around 133,000 classified driving scenes under shadow and occlusion conditions from CULane Dataset.

Basic measures derived from the confusion matrix:

Error rate (ERR) is calculated as the number of all incorrect predictions divided by the total number of the dataset. The best error rate is 0.0, whereas the worst is 1.0.

Error Rate=(FP+FN)/(P+N)

Accuracy (ACC) is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0.0. It can also be calculated by 1 – ERR.

Accuracy = (TP+TN)/(P+N)

Precision (PREC) is calculated as the number of correct positive predictions divided by the total number of positive predictions. It is also called positive predictive value (PPV). The best precision is 1.0, whereas the worst is 0.0.

Precision = TP/TP+FP

False positive rate (FPR) is calculated as the number of incorrect positive predictions divided by the total number of negatives. The best false positive rate is 0.0 whereas the worst is 1.0. It can also be calculated as 1 – specificity.

FP= FP/TN+FP

The performance analysis results are as follows:

Sl. No	1
Dataset	Driving Scenes from TuSimple Dataset
Data Set Split	Total=6,408 Training=3,626 Testing=2,782
Error Rate(ERR) =FP+FN/P+N	4%
Precision =TP/FP+FN	0.964
Accuracy =(1-ERR)	±96%

Table 2: Performance analysis of DCNN model for TuSimple Dataset.

Sl. No	2
Dataset	Driving Scenes from CULane Dataset
Data Set Split	Total= 133,235 Training= 88,880 Testing= 34,680
Error Rate (ERR) =FP+FN/P+N	18%
Precision =TP/FP+FN	0.827
Accuracy =(1-ERR)	±82%

Table 3: Performance analysis of DCNN model for CULane Dataset.

The output window consists of three different windows as shown in figure 3. The first window depicts the final output video of the detected lane as a live video pipeline, another window shows the corresponding live video pipeline for all the processes that the test video is subjected to. This is done to increase transparency on the process of lane detection. The third window shows the live detected lane curvature parameters. Also, the predicted position of the vehicle and the supposed directed of steering according to the detected lane is printed live every 0.5 seconds on the terminal. This can be used as an input for an external automated vehicle steering module. The model is also tested with multiple test videos with varying degrees of complexity and is observed to be robust in most cases.

Fig -4: Output visual for a test video showing all the steps in lane detection procedure.

6. CONCLUSION

It is evident that recent advances have been made in the detection of road lanes on driving scenes. Even though there are several methods that achieved significant advancements using high precision and efficient methodologies, there are still many challenges pertaining to extreme driving conditions that have to be addressed. To overcome all these challenges we are proposed an optimized and hybrid combination of Deep CNN and RNN. The reason we choose CNN is that it can extract the spatial from the data using kernels, which other networks are not capable of. The proposed method uses a combination of DCNN and RNN to predict road lanes using a continuous sequence of frames as an input. Through vigorous experimentation and testing, our model has achieved a significant accuracy of 96% and a precision of 0.964. We have also demonstrated the step by step inner workings of our model in our output, for a better visual understanding of the lane detection process by our model.

7. REFERENCES

[1] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. chun Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in Advances in neural information processing systems (NIPS), 2015, pp. 802–810.

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

[3] Lee, C. and Moon, J.H., 2018. Robust lane detection and tracking for real-time applications. *IEEE Transactions on Intelligent Transportation Systems*, *19*(12), pp.4043-4048.

[4] Li, J., Mei, X., Prokhorov, D. and Tao, D., 2016. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE transactions on neural networks and learning systems*, *28*(3), pp.690-703.

[5] Zhang, J., Deng, T., Yan, F. and Liu, W., 2020. Lane Detection Model Based on Spatio-Temporal Network with Double ConvGRUs. *arXiv preprint arXiv:2008.03922*.

[6] Aung, T. and Zaw, M.H., 2014, March. Video based lane departure warning system using Hough transform. In *International Conference on Advances in Engineering and Technology* (pp. 29-30).

[7] Nugraha, B.T. and Su, S.F., 2017, October. Towards self-driving car using convolutional neural network and road lane detector. In *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)* (pp. 65-69). IEEE.

[8] Kamçı, S., Aksu, D. and Aydin, M.A., 2019. Lane Detection For Prototype Autonomous Vehicle. *arXiv preprint arXiv:1912.05220*.

[9] Mamidala, R.S., Uthkota, U., Shankar, M.B., Antony, A.J. and Narasimhadhan, A.V., 2019, October. Dynamic approach for lane detection using Google street view and CNN. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)* (pp. 2454-2459). IEEE.

[10] Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H. and So Kweon, I., 2017. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 1947-1955).

[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. Garc´ıa. 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

[17] and Vision Computing, 22:269–280, 04 2004.

[18] 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.

[19] 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.

[20] 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.