2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

An efficient encoder-decoder CNN architecture for reliable multilane detection in real time

Shriyash Chougule[1], Asad Ismail[2], Ajay Soni[1], Nora Kozonek[2], Vikram Narayan[2], and Matthias Schulze[2]

***Abstract*— Multilane detection system is a vital prerequisite for realizing higher ADAS functionality of autonomous navigation. In this work, we present an efficient convolutional neural network (CNN) architecture for real time detection of multiple lane boundaries using a camera sensor. Our network has a simple encoder-decoder architecture and is a special two class semantic segmentation network designed to segment lane boundaries. Efficacy of our network stems from two key insights which are at the foundation of all our design decisions. Firstly, we term a lane boundary as a *weak class* object in the context of semantic segmentation. We show that the weak class objects which occupy relatively few pixels in the scene, also have a relatively low detection accuracy among the know segmentation methods. We present novel design choices and intuitions to improve the segmentation accuracy of weak class objects, which in turn reduces computation time. Our second insight lies in the manner we depict the ground truth information in our derived dataset. Instead of annotating just the visible lane markers, we accurately delineate the lane boundaries in the ground truth for challenging scenarios like occlusions, low light and degraded lane markings. We then leverage the CNN's ability to concisely summarize the global and local context in an image, for accurately inferring lane boundaries in these challenging cases. We evaluate our network against ENet and FCN-8, and found it performing notably better in terms of speed and accuracy. Our network achieves an encouraging 46 FPS performance on NVIDIA Drive PX2 platform and it has been validated on our test vehicle in highway driving conditions.**

I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) is experiencing rapid adoption and growth in automotive systems [1]. A recent study [2] estimated that ADAS functions are effective in reducing head-on and single-vehicle crashes, and minimizing driver injury risks. Among these safety-critical functionality, capability of autonomous navigation is most sought after function in ADAS. The ability to accurately and reliably detect ego lane and side lanes lie at the core of autonomous driving, and also serves other critical driving assistance tasks like lane keeping, lane departure warning, and path planning [8]. A vision based lane detection approach provides a low cost solution, but is required to perform robustly in complex driving scenarios. Extracting lane signature from road surface becomes challenging due to varying road appearance. Low light situations like sunset, dawn and night add to the challenge. And the varying road texture due to shadows casted by trees, vehicles and buildings demands extreme robustness as well. Other inevitable situations like worn-out lane markings, weakly marked lane boundaries (using periodically placed reflectors) and occlusions due to vehicles is very common driving scenario. Also, due to the road geometry, the side lane signatures are relatively feeble than ego lane in a given image. Thus estimating side lane boundaries is relatively difficult and rarely explored problem. Model based approaches use traditional computer vision techniques to devise specialized hand-crafted features. Such solutions usually works under a controlled environment and are likely to fail in complex driving scenarios. Convolutional neural networks (CNN) have demonstrated a superior performance in pattern recognition [10,11,12]. An computer vision approach, devised around a CNN, has the potential to bring about a robust solution to autonomous driving.

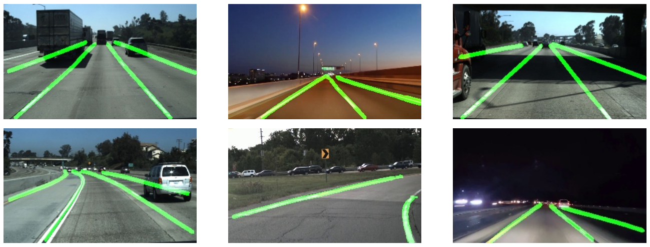


Fig. 1: Examples of multilane detection by our network.

In this work, we intent to exploit the CNN's ability of recognizing complex patterns for gaining robustness in challenging driving situations (alongside other motivations listed in Section 3). Towards that direction we investigate architectures of renowned semantic segmentation networks (ENet [22], SegNet [20] and FCN [21]) to assess their potential for segmenting lane boundaries. We discover that these networks have inherent low sensitivity to objects that occupy relatively few pixels in a image, and we term such objects (like lane markings and poles) as *weak class* objects. With these acquired insights, we choose to optimize the encoder-decoder CNN architecture for segmenting lane boundaries as described in Section 4. Our design decisions place emphasis on achieving better speed and accuracy in detecting weak class objects. In order to train our network for segmenting lane boundaries, we derive a dataset from the published TU Simple dataset (lane detection challenge dataset [25]) as detailed in Section 4. In Section 5, we compare our network against the ENet and FCN-8 [21] for the task of lane boundary segmentation, where we demonstrate that our network is notably more reliable and faster than the rest. Examples of multilane detection by our network are shown

[1] Authors are with Visteon Corporation, Pune, India. {schougu1, ajay.soni}@visteon.com

[2] Authors are with Visteon Corporation, Karlsruhe, Germany. {aismail2, nkozone2, vnaraya4, matthias.schulze}@visteon.com

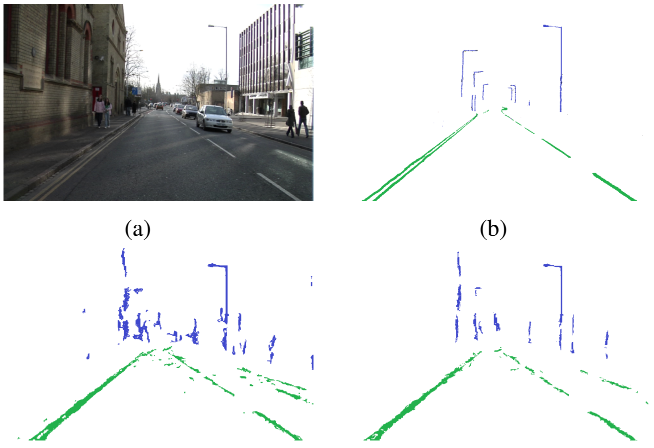
978-1-5386-4451-5/18/$31.00 ©2018 IEEE

in Fig.1. In summary, our work has following contributions:

- An efficient network for reliably detecting multiple lane boundaries in real time, without employing any post processing and tracking framework.
- We provide network design choices and intuitions to significantly improve segmentation accuracy of objects that occupy relatively few pixels in the scene.
- Our work exemplifies that an effective training dataset can be derived from the moderate TU Simple dataset (lane detection challenge dataset [25]), and together with transfer learning, this derived dataset is adequate for effectively training a segmentation network.

II. Related Work

Most of the proposed methods for lane detection use hand-crafted features in a model driven approach. An adaptive lane feature extraction method [3] detects changes in lighting condition using HSV histogram. Inverse perspective mapping (IPM) is used to discover lanes in [16] by line fitting through Hough transform and using splines. In [4] a steerable filter is devised for feature extraction, which is applied on equally spaced horizontal bands in an IPM of input frame. Prior information about vehicle position is made use of in [5] to improve false detection in [4]. A feature derived by fusing LiDAR and vision data statistics [6] is used to classify the crub points along the road side which in turn is used for lane detection. Tracking frameworks based on Kalman filter [17] have demonstrated robustness against noisy feature measurements. Superparticle [7] uses multiple particle filters for ego lane detection. Lanequest [9] demonstrate energy efficient solution to identify vehicle's current lane position using inertial sensors in a smartphone, but it cannot identify road curvature ahead. An efficient particle filter variant [14] is shown to identify road geometry ahead in real-time. Recent methods [11, 12] incorporating CNN, outperform model based methods in detecting lane markers. A CNN used as a preprocessing step in a lane detection system [10], helps enhancing edge information by suppressing noises and edges from obstacles. DeepLanes [13] detects immediate sideward lane marking with laterally mounted camera system. Although DeepLanes achieve real time performance with high accuracy, it cannot detect road turning ahead. Multilane detection in [8] makes use of CNN and regression to identify line segments that approximate a lane boundary, it requires large resolution image to work with which hampers the speed. Vision based multilane detection system [15] combines GPS information and map data. A closely related method to our work estimates ego lane in an end-to-end fashion [18] using SegNet, it is promising but has low segmentation accuracy because of which the detected lane boundaries appear fragmented and splattered.


(a)

(b)

(c)

(d)

Fig. 2: Detected lane markings and poles appear fragmented due to poor segmentation accuracy (a) Camvid dataset example. (b) Retained ground truth labels for poles and lane markings. (c) SegNet segmentation result. (d) ENet segmentation result.

III. Motivation

From the literature review, we observe that the lane detection problem has been pursued for last three decades [1]. It is an indication that the lane detection has always been a vital and desired functionality for autonomous systems. Such continual of pursuit also implies a dearth of practically feasible methods. Also it can be observed that substantial work has been done to address ego lane detection problem, whereas multilane detection is widely perceived as a next frontier. Given these background, the motivation for our work is driven by following points: (a) Explore the seldom addressed problem of multilane detection. (b) Exploit the CNN's potential to concisely summarize global and local context in an image, with an intent to provide robustness in challenging driving conditions like occlusion and low light. (c) Reliably detect lane boundaries at the road turnings. (d) Satisfy the real-time performance requirement. (e) Avoid intricacies of the devising post processing stage and tracking framework.

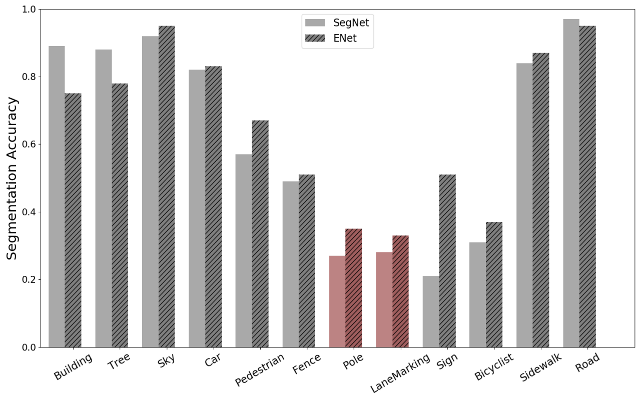


Fig. 3: Performance of SegNet and ENet on Camvid dataset. Lane marking and pole class have relatively low segmentation accuracy.

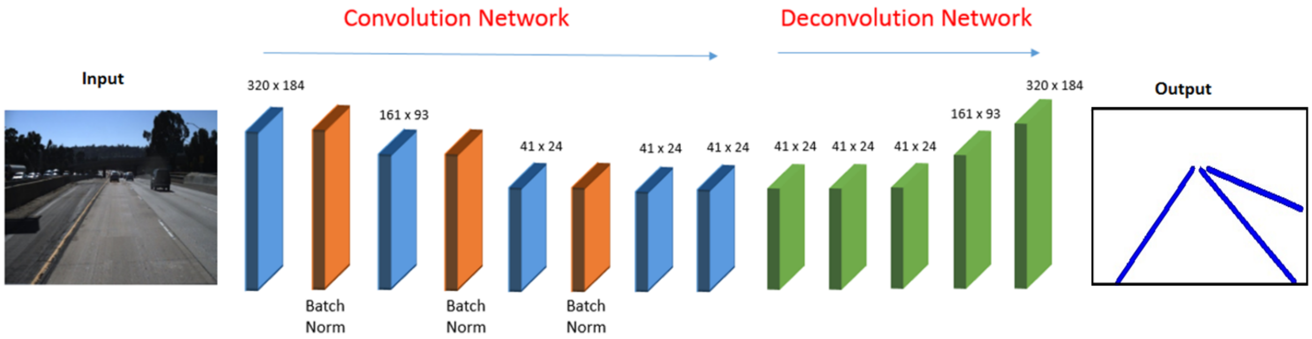


Fig. 4: Illustration of our encoder-decoder CNN architecture. Convolutional network (or encoder section) consist of convolutional and batch normalization layers which are shown by the blue and orange slabs respectively. Deconvolutional network (or decoder section) consist of deconvolutional layers shown by green slabs.

IV. Multilane Detection Network

A. *Segmentation of weak class objects*

Lane markings and poles have peculiar shape of being an elongated and thin object, usually occupying less than 5% of the total pixels in a driving scene image. We term these objects as *weak class* objects. We analyze two prominent semantic segmentation networks like SegNet [20] and ENet [22], to evaluate their potential for segmenting weak class objects like lane boundaries. We summarize their performance on Camvid dataset [27], which contain examples of an urban driving scenario. As can be seen in Fig. 3, the segmentation accuracy for lane markings (road markings) class and pole class is low in comparison with remaining classes. As a consequence of poor segmentation accuracy, the segmented lane markings appear fragmented due to many false positive and false negative as shown in Fig. 2. Inferring lane boundaries from these fragmented lane markings demands designing of a tedious post-processing stage, followed by a tracking framework. Fragmentation also hinders the prospect of estimating turns and curvature of the road ahead, which is a key information for critical driving assistance tasks (such as path planning and lane keeping). Post processing and tracking can be avoided if the lane boundaries can be directly inferred by a network in an end-to-end fashion, as done in [18] using SegNet. Although end-to-end approach is promising, the fragmentation problem persists because the lane boundaries can still be identified as weak class objects (elongated and thin). Thus network architectures like SegNet and ENet are not appropriate for lane boundary segmentation task. In the following subsections we elaborate our network design decisions to boost the segmentation accuracy of weak class object like lane boundaries and minimize fragmentation.

B. *Encoder-Decoder architecture*

With a view to improve the segmentation accuracy of weak class objects like lane boundaries, we also analyze FCN [21] architecture alongside SegNet and ENet. FCN is the first semantic segmentation network derived from an image classification network. Wherein the image classification network is a sequence of several convolutional layer that ends with a fully connected layer. In FCN, the fully connected layers of image classification network are replaced by fully convolution layers, followed by a single deconvolutional layer of fixed size receptive field. As a consequence of the fixed size receptive field, segmentation accuracy of weak class objects suffers and such objects appears to be fragmented after segmentation. Use of gradually graded deconvolution layers instead of a single deconvolution layer is known to construct denser and much precise segmentation mask [23]. Thus the segmentation accuracy of weak object class like lane boundaries, can greatly benefit by these graded deconvolutional layers. So we choose to optimize a symmetrical encoder-decoder architecture [23] where the encoder section (or convolutional network) is a sequence of convolutional layers, followed by the decoder section (or deconvolutional network) involving deconvolutional layers as shown in Fig. 4. The encoder section behaves as a feature extractor whereas the decoder section constructs a segmentation map from the extracted features.

C. *Compact structure*

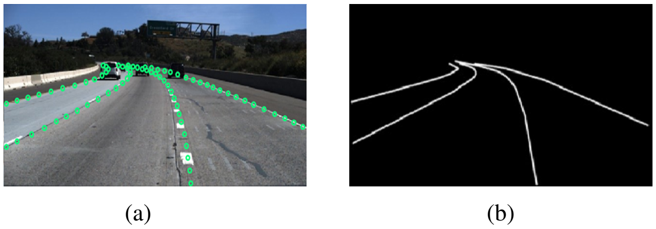
Semantic segmentation networks are fundamentally multi-class classifiers, where the network architecture is tailored for obtaining higher average accuracies over all classes (number of classes in the popular dataset of Camvid [27] and Cityscape [24] are 12 and 19 respectively). In recent years, a practice of continually adding more layers to the network with expectancy of improving the accuracy has been established. Outcome of this trend are even deeper networks, capable of learning higher order nonlinearities required for complex tasks. Such deep networks are computationally expensive and adds to the execution time, which is undesirable. Lane boundary segmentation can be viewed as a special case of two class semantic segmentation, which can be regarded as a simpler task compared to the multi-class segmentation task. Thus use of a network that is customized for a multiclass semantic segmentation (like SegNet, ENet and FCN) to solve the lane segmentation problem, creates a sense of underutilization of the base architecture. With this intuition we strip down the original 39 layers encoder-decoder network [23] to 10 layers network. Where the later has 5 convolutional layers in the encoder section (or convolutional network) and 5 deconvolution layers in the decoder section (or deconvolutional network). We also insert batch normalization layers [19] in between first four convolutional layers (as shown in Fig. 4) to bring in regularization effect. Such compact structure of encoder-decoder network significantly cut down the computations, and yet is accurate enough for segmenting lane boundaries as we demonstrate in Section 5.

D. No pooling operations

It is well known in the computer vision domain that for any given method, lowering the input image resolution provides execution speed up at the cost of accuracy. ENet leverages this fact and achieves impressive speedup by employing a strategy of early down sampling, done by its first two bottleneck layers [22]. Instead of going for early down sampling strategy, we choose to work directly with a low resolution input image to uplift the execution speed. And to counter the accuracy loss because of lowering the input image resolution, we choose to remove the max pooling layers and corresponding unpooling layers from the original deconvolution network. Max-pooling is ubiquitous procedure used to retain most active neurons, which translates in decreased features dimensionality as well as introduces some degree of translation invariance. However, in our problem we found that loosing spatial information by using max pooling layer is indeed not desirable. We recognize that presence of pooling layers in the original network, gradually weaken the localization accuracy which affects the weak class objects the most. Without pooling layers our network produces much finer segmentation masks, and saves the computation time as well. We also removed the fully connected layers from the original network and found out it has negligible effect on the accuracy but improves the speed of the network considerably.

E. Dataset generation

To train our network for lane segmentation task, we derive our dataset from TU Simple's lane detection challenge dataset [25] which contains examples of highway driving scenario. The challenge dataset contain 3626 images annotated with lane boundaries. An image in the challenge dataset may have 3 to 5 annotated lane boundaries, depending upon the scene. The ground truth information for each lane boundary is represented by a list of points, described in terms of image coordinates. These images are of 1280 x 720 resolution and recorded in normal weather conditions. We decide to derive a new dataset as the number of images in the challenge dataset is too low for effective learning, as well as the format of ground truth information is not suitable for training a segmentation network. For a lane boundary segmentation task, the ground truth has to be a binary image where pixels with label 1 denote the lane boundary region and the pixels with label 0 denote non-lane boundary region. We derive a dataset (lane segmentation dataset) which has the new ground truth information in suitable format, and also perform data argumentation to generate sufficient examples for training our segmentation network.



(a) (b)

Fig. 5: (a) Example from TU Simple dataset, where ground truth information for lane boundary consist of a list of points shown as small green circles. (b) Binary image used as ground truth in derived dataset, where lane boundaries are shown by white curves.

For generating the new ground truth, we make use of the pixel co-ordinate information (for each lane boundary) from the original dataset shown in Fig. 5(a) as small green circles. We fit and plot a second order polynomial using these pixel co-ordinates to obtain a binary image, which contains the lane boundaries delineated by the white curves as shown in Fig. 5(b). During data augmentation we generate additional images, by cropping different size windows form each of the challenge dataset image (and from its corresponding ground truth binary image) and resizing them to 320 x 184 resolution. On these rescaled images, we carry out operations like rotations, flip around vertical axis, changing the image intensities to reflect low lightning condition, adding Gaussian noise and salt-and-pepper noise of varying levels. After data augmentation process, the generated lane segmentation dataset contains 95000 training images and 1000 validation images.

F. Network training

For training our encoder-decoder network we utilize transfer learning procedure [26], which involves borrowing of weights from a network that is trained on a different dataset. We decided to build our encoder section (convolutional network) by reusing and fine-tuning the pretrained weights of the FCN-AlexNet [21] along with the pretrained weights of ENet [22]. Both these networks were originally trained on Cityscape dataset [24]. For convolutional layer number 2 to 5 in our network, we used the first four convolutional layers of FCN-AlexNet (refer Table 1). However, the first convolutional layer of FCN-AlexNet has a large kernel size (11x11) and longer stride (4x4) which reduces sensitivity of detecting finer features. In order to obtain precise prediction of the lane boundaries, it is therefore necessary to use a smaller kernel size and short stride in the beginning. With regard to this, we decided to insert an additional convolutional layer at the beginning (first convolutional layer in our network). So we use the first convolutional layer of ENet along with its pretrained weights, which has a smaller kernel size (3x3) and shorter stride (2x2) allowing the network to detect finer features. By inserting an additional layer at the beginning, the

TABLE I: Network description.

Layer name	Kernel size	Stride	Padding	Details
Conv1	(3x3)	(2x2)	1	borrowed from ENet's 1st conv layer
Conv2	(11x11)	(4x4)	5	weights are learned (based on FCN-ALexNet's 1st conv layer)
Conv3	(5x5)	(1x1)	2	borrowed from FCN-ALexNet's 2nd conv layer
Conv4	(3x3)	(1x1)	1	borrowed from FCN-ALexNet's 3rd conv layer
Conv5	(3x3)	(1x1)	1	borrowed from FCN-ALexNet's 4th conv layer
Deconv1	(3x3)	(1x1)	1	weights are learned
Deconv2	(3x3)	(1x1)	1	weights are learned
Deconv3	(5x5)	(1x1)	2	weights are learned
Deconv4	(11x11)	(4x4)	5	weights are learned
Deconv5	(10x10)	(2x2)	5	weights are learned

pretrained weights of the first FCN-AlexNet convolutional layer are no longer relevant and therefore has to be learned from the scratch. During fine tuning the borrowed weights get adapted for lane boundary segmentation task. Unlike the convolutional layers, all the deconvolutional layers in the decoder section are learned from the scratch during the fine tuning process. We use the following loss function during training process:

$$Loss = \frac{1}{2N} \sum_{n=1}^{N} \|\hat{y} - y)\| \qquad (1)$$

Where N is the number of images in a training batch, $\hat{y}$ is the predicted output and y is the ground truth. We trained the network on the derived lane segmentation dataset for 776000 iterations. We use Adam optimizer with the base learning rate of 1e-4, with momentum of 0.9 and sigmoid as learning rate decay policy.

V. Experimental results

We evaluate our network by comparing it with FCN-8 [21] and ENet [22], which are trained for lane boundary segmentation task using our derived dataset (lane segmentation dataset). FCN-8 is well known semantic segmentation network, and to the best of our knowledge ENet is currently renowned to be the fastest semantic segmentation network. We did not include SegNet in our comparison since performance of ENet is shown to better than SegNet [22].

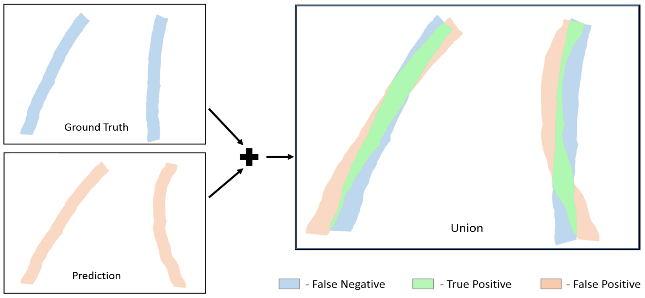


Fig. 6: Illustration of metric parameters. Overlaying prediction image over ground truth image reveals three distinct regions, identified as false positive (red color), false negative (blue color) and the intersection region as true positive (green color).

A. Quantitative analysis

In order to gauge the lane detection accuracy of our network we calculate Dice coefficient and Jaccard index (Intersection of Union or IoU), which are well known metrics for evaluating a segmentation task. The Dice coefficient also known as F1 score combines both precision and recall into one metric, where precision and recall are defined as follows:

$$Precision = \frac{TP}{TP + FP} \qquad (2)$$

$$Recall = \frac{TP}{TP + FN} \qquad (3)$$

Where TP, FP and FN represents true positive, false positive and false negative respectively. In our case, these three entities manifest themselves as pixel count in three distinct regions as illustrated in Fig. 6. The Dice coefficient which combines both precision and recall is defined as:

$$Dice = 2 * \frac{Precision * Recall}{Precisoin + Recall} \qquad (4)$$

The second metric we used for comparison is Jaccard index also commonly known in the field of computer vision as Intersection of Union (IoU), and is defined as:

$$Jaccard = \frac{TP}{TP + FP + FN} \qquad (5)$$

Dice coefficient and Jaccard index are positively correlated, but quantitatively Jaccard index penalizes an instance of bad classification more than Dice coefficient. We compute average values of these two metrics using predictions over 1000 validation images, as summarized in Table II. Our network has higher Dice coefficient and higher Jaccard index in comparison with other two networks, implying better precision and recall values. Higher values of these two metrics also indicates a strong congruity between ground truth and predictions. We benchmark the execution speed of our network on Nvidia Drive PX2 and Nvidia GTX 1080 graphics card, with Caffe (deep learning framework) and cuDNN (deep learning library by Nvidia) at the back-end. And working on the input images of resolution 320 x 184. We summarize the average execution time per frame and the corresponding average FPS (frames per second) in Table III, where our network's execution speed is notably better than the FCN-8 and ENet. These favorable results corroborate our intuitions and network design choices.

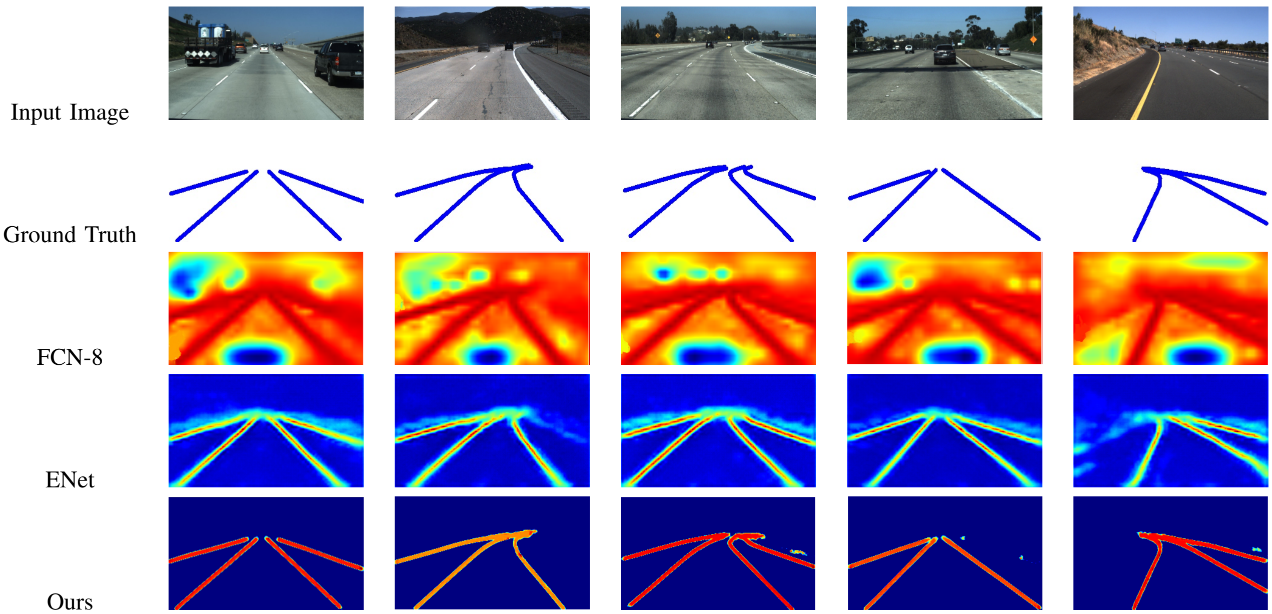


Fig. 7: Segmentation map shows the network's confidence in classifying pixels as lane boundary region. They are shown as heat maps where dark red pixels represent higher confidence end whereas dark blue pixels indicates lower confidence end. Segmentation maps of our network have negligible variance and consistent confidence level along the lane boundary length.

TABLE II: Lane boundary segmentation accuracy

Model	Dice Coefficient	Jaccard Index (IoU)
FCN-8	0.816	0.693
ENet	0.809	0.685
Ours	**0.886**	**0.793**

TABLE III: Speed performance

Model	Nvidia Drive PX2		Nvidia GTX 1080	
	ms	FPS	ms	FPS
FCN-8	456	2.2	36.3	27.5
E-Net	27	37	4	250
Ours	**21.6**	**46**	**1.78**	**560**

B. Qualitative analysis

For qualitative analysis we observe our network's performance on examples from TU Simple [25], Cityscape [24] and Camvid datasets [27], night driving examples, as well as we validated it on our test vehicle in highway test drive.

1) Analysis using segmentation maps: A network trained for N class semantic segmentation task will generate N class conditional probability maps, which are also known as segmentation maps. Each segmentation map is of the same resolution as the input image. Thus inspection of the segmentation maps generated by a particular network, reveals the network's sensitivity towards a specific class under consideration. We obtain segmentation maps from FCN-8, ENet and our network, using examples from our validation set. As shown in Fig. 7, these segmentation maps are displayed as heat maps where the per pixel probabilities are represented by color spectrum (from dark red representing the highest confidence end to dark blue at the lower confidence end). It can be observed that the probability distribution in segmentation maps of FCN-8 has wider spread (high variance), and thus it is more likely to misclassify lane boundary pixels. Although ENet's segmentation maps show a very low variance as compared to FCN-8, the lane boundary confidence drops sharply at turnings which hinders the detection of road curvature. Unlike ENet and FCN-8, segmentation maps from our network show a consistent confidence level along the entire length of lane boundaries and even at the road turnings. Our networks ability to generate these finer segmentation maps (having negligible variance) can be attributed to our decision of drooping pooling operation.

2) Predictions in challenging scenarios: The final segmentation mask in most of semantic segmentation networks are computed by comparing confidence scores of all the classes (two classes in our case), obtained for a given input image. For any pixel in the input image, the confidence scores at that particular pixel's location are compared. The pixel is classified to a class which has highest confidence score. Our network on the other hand directly produces segmentation mask through deconvolution layers. Fig. 8 show predictions by ENet, FCN-8 and our network on images from our validation dataset (having examples of occlusions, degraded lane markings, shadows and road turning). It can be observed that the predictions by our network closely resembles to the ground truth. ENet and FCN-8 predictions are visibly poor in comparison and appear splattered and fragmented. Fig. 10 show predictions of our network on examples form Cityscape and Camvid datasets. These datasets are collection of urban driving scenes, containing various informative road markings (like pedestrian crossing, lane direction arrows, vehicle parking slots and such other road symbols). As our derived lane segmentation dataset does

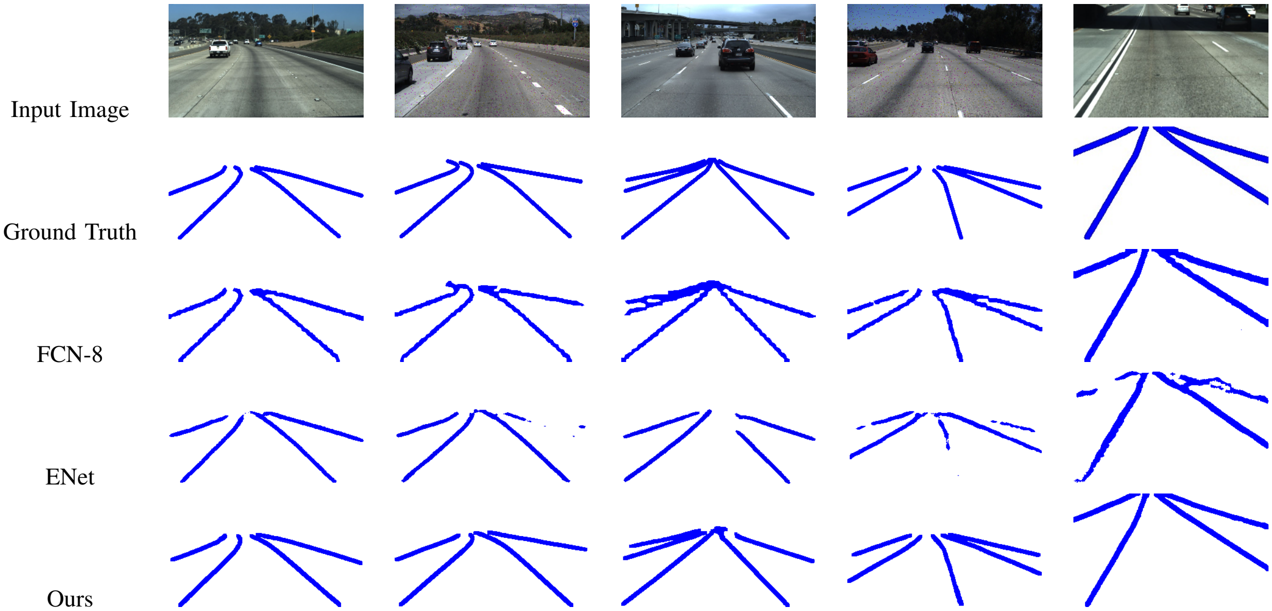


Fig. 8: Lane boundary predictions of our network closely resembles to the ground truth. Predictions by FCN-8 and ENet appear splattered and fragmented.

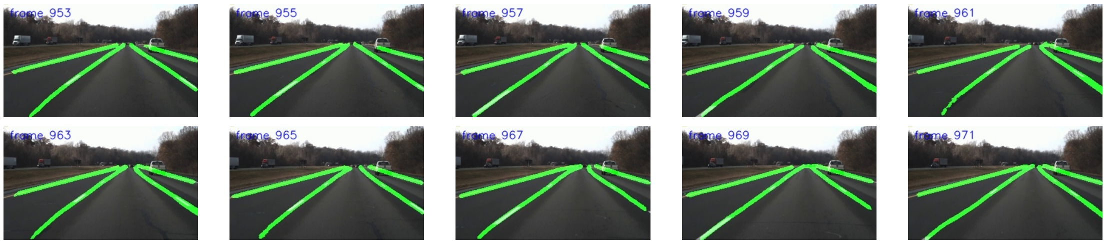


Fig. 9: A stable and consistent detection of lane boundaries can be observed over a sequence of frames (odd numbered) recorded during highway test drive.

not contain such complex examples to train our network, we observe some false classifications on the images from these two datasets (see Fig. 10). The misclassification is high especially at the junctions, cases where no lane markings are present and in the presence of informative road markings. A similar situation is noticed in night driving scenes where a accuracy loss can be observed. Although our training dataset lacked the inner city and night driving examples, the prediction results of our network are still encouraging in these situations.

3) Validation in real life driving situation: Our test vehicle is fitted with front facing point gray camera to capture the scene ahead at 30 FPS. For computational operations we have used NVidia Drive PX2 running on a linux platform. We record the scene and our network's predictions on them using rosbag (a utility form robot operating system). In Fig. 9 a sequence of odd predicted frames is shown form the recorded rosbag. A reliable and stable detection can be observed over these frames which is necessary to reduce dependency on tracking operation. Overall performance of our network was promising during the highway test driving.

VI. Conclusion

We presented an efficient encoder-decoder CNN architecture for a reliable multilane detection using a camera sensor. Our network is trained to detect both the ego lane and side lane boundaries by segmenting them from the input video frame, without using any post processing and tracking operation. We provide fresh network design choices and intuitions, chiefly considering the improvement of detection speed and segmentation accuracy of lane boundaries. We obtained encouraging results during evaluation against ENet and FCN-8, where our network's accuracy and speed were significantly better. We benchmark the execution speed of our network on Nvidia Drive PX2 and Nvidia GTX 1080 where we observe a performance of 46 FPS and 560 FPS respectively.

Our network can reliably detect lane boundaries that are represented by broken white lane markings, solid lane markings (both yellow and white) and implicit lane markings represented by periodically placed reflectors. We observe consistent detection during partial occlusions by vehicles and at the road turnings. A promising performance was observed

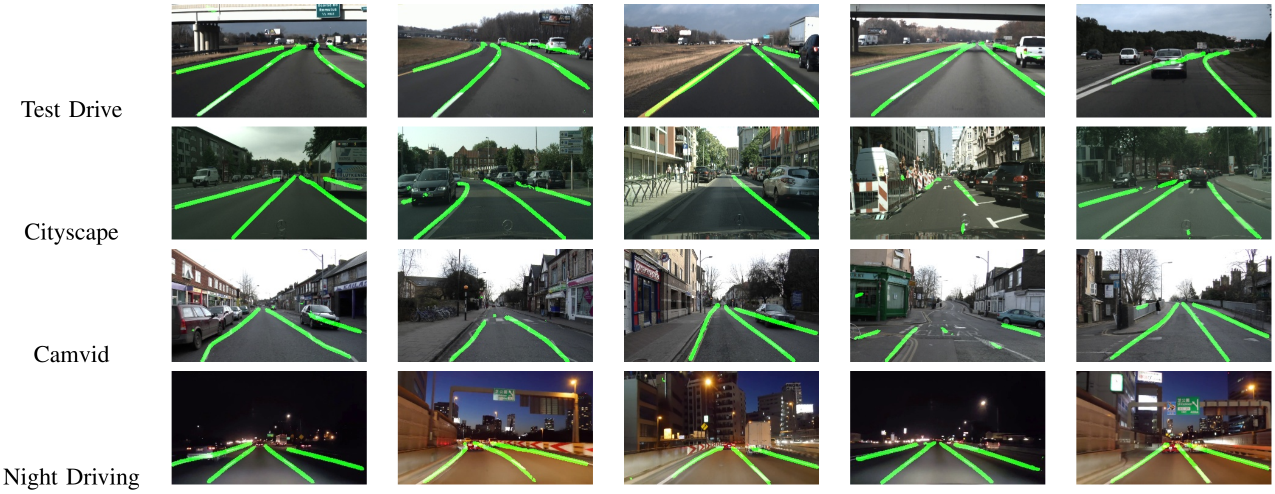


Fig. 10: First row shows detection results recorded during highway test drive. False classification due to informative road markings can be observed in urban settings of Camvid and Cityscape datasets. Accuracy loss can be observed at the road turnings and during occlusions in night driving scenes.

during our highway test driving. Although an accuracy loss is observed in night driving scenes and in the urban driving scenes where the informative road markings clutter the scene, the detection results are encouraging given that our training dataset lacked such examples. We plan to detect positional information of lane boundaries (left, right, extreme left and extreme right lane boundary) as an extension to this work, and experiment with synthetic dataset for generating challenging driving examples with an intent to add robustness.

References

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, Three Decades of Driver Assistance Systems: Review and Future Perspectives, IEEE Intell. Transp. Syst. Mag., vol. 6, no. 4, pp. 6-22, 2014.

[2] S. Sternlund, J. Strandroth, M. Rizzi, A. Lie, and C. Tingvall, The effectiveness of lane departure warning systems: A reduction in real-world passenger car injury crashes, Traffic Inj. Prev., vol. 18, no. 2, pp. 225-229, 2017.

[3] V. S. Bottazzi, P. V. Borges, and B. Stantic. Adaptive regions of interest based on hsv histogram for lane marks detection. Robot Intelligence Technology and Applications 2, 274:677-687.

[4] R. K. Satzoda and M. M. Trivedi, Vision-based Lane Analysis: Exploration of Issues and Approaches for Embedded Realization, 2013.

[5] R. K. Satzoda and M. M. Trivedi, Efficient lane and vehicle detection with integrated synergies (ELVIS), in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 708-713.

[6] Q. Li, L. Chen, M. Li, S. L. Shaw, and A. Nuchter, A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios, IEEE Trans. Veh. Technol., vol. 63, no. 2, pp. 540-555, 2014.

[7] B. S. Shin, J. Tao, and R. Klette, A superparticle filter for lane detection, Pattern Recognit., vol. 48, no. 11, pp. 3333-3345, 2015.

[8] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, Recent progress in road and lane detection: A survey, Machine Vision and Applications, vol. 25, no. 3. pp. 727-745, 2014.

[9] H. Aly, A. Basalamah, and M. Youssef, LaneQuest: An accurate and energy-efficient lane detection system, in 2015 IEEE International Conference on Pervasive Computing and Communications, PerCom 2015, 2015, pp. 163-171.

[10] J. Kim, J. Kim, G. J. Jang, and M. Lee, Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection, Neural Networks, vol. 87, pp. 109-121, 2017.

[11] M. Park, K. Yoo, Y. Park, and Y. Lee, Diagonally-reinforced Lane Detection Scheme for High-performance Advanced Driver Assistance Systems, jsts.org, vol. 17, no. 1, pp. 79-85, 2017.

[12] J. Li, X. Mei, D. Prokhorov, and D. Tao, Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene, IEEE Trans. Neural Networks Learn. Syst., vol. 28, no. 3, pp. 690-703, 2017.

[13] A. Gurghian, T. Koduri, S. V Bailur, K. J. Carey, and V. N. Murali, DeepLanes: End-To-End Lane Position Estimation using Deep Neural Networks, Comput. Vis. Pattern Recognit., pp. 38-45, 2016.

[14] M. Nieto, A. Corts, O. Otaegui, J. A.-J. of R.-T., and U. 2016, Real-time lane tracking using Rao-Blackwellized particle filter, Springer.

[15] D. Cui, J. Xue, S. Du, and N. Zheng, Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration, in IEEE International Conference on Intelligent Robots and Systems, 2014, pp. 4958-4964.

[16] M. Aly, Real Time Lane Detection in Urban Streets, Intell. Veh. Symp., pp. 1-3, 2008

[17] M. T. Smith, Robust lane detection and tracking with RANSAC and kalman filter Amol Borkar, Monson Hayes, Image Process., pp. 3261-3264, 2009.

[18] J. Kim and C. Park, End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2017, vol. 2017 July, pp. 1194-1202.

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561, 2015.

[21] J. Long, E. Shelhamer, and T. Darrell, Fully convolutional networks for semantic segmentation, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07 12 June, pp. 3431-3440, 2015.

[22] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147, 2016.

[23] H. Noh, S. Hong, and B. Han, Learning deconvolution network for semantic segmentation, in Proceedings of the IEEE International Conference on Computer Vision, 2015, vol. 2015 Inter, pp. 1520-1528.

[24] M. Cordts et al., The Cityscapes Dataset for Semantic Urban Scene Understanding, cv-foundation.org, 2016.

[25] http://benchmark.tusimple.ai/#/t/1

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, How transferable are features in deep neural networks, papers.nips.cc, 2014.

[27] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 2008.