DeepLanes: End-To-End Lane Position Estimation using Deep Neural Networks

Alexandru Gurghian, Tejaswi Koduri, Smita V. Bailur, Kyle J. Carey, Vidya N. Murali
Ford Research and Innovation Center
Palo Alto, California, 94304
{agurghia, tkoduri, sbailur, kcarey10, vnariyam}@ford.com

Abstract

Camera-based lane detection algorithms are one of the key enablers for many semi-autonomous and fully-autonomous systems, ranging from lane keep assist to level-5 automated vehicles. Positioning a vehicle between lane boundaries is the core navigational aspect of a self-driving car. Even though this should be trivial, given the clarity of lane markings on most standard roadway systems, the process is typically mired with tedious pre-processing and computational effort. We present an approach to estimate lane positions directly using a deep neural network that operates on images from laterally-mounted down-facing cameras. To create a diverse training set, we present a method to generate semi-artificial images. Besides the ability to distinguish whether there is a lane-marker present or not, the network is able to estimate the position of a lane marker with sub-centimeter accuracy at an average of 100 frames/s on an embedded automotive platform, requiring no pre- or post-processing. This system can be used not only to estimate lane position for navigation, but also provide an efficient way to validate the robustness of driver-assist features which depend on lane information.

1. Introduction

Modern cars incorporate an increasing number of advanced driver assist features such as lane keeping, automatic emergency braking, automatic lane following on highways. Fully autonomous driving cars are the main focus of research in all automotive and high-tech industries. A key enabler for these technologies is environmental perception, for which a wide range of sensors are employed. Camera based lane detection is one of the key components for many of these features. Allowing the car to determine its current position in the lane is a foundation for any subsequent lane departure detection and path/trajectory planning.

Many proposed methods for lane detection use traditional computer vision techniques, often incorporating highly-specialized and hand-crafted features combined in a model-based approach. This has two disadvantages: (i) Position estimation is dependent on accurate segmentation which is computationally intensive. (ii) Such a system would not lend itself to easy scalability because of road scene variations. Using a deep neural network to estimate lane positions, eliminates the need for segmentation. Another major advantage of deep neural networks is the fact that they do not rely on application specific or hand-crafted features, as they are able to implicitly learn efficient feature representation for the application at hand during training.

Virtually all approaches use a front-facing camera to perform lane detection. This camera configuration does not provide an optimal view of the lane markings, as it also captures the whole front facing scene, including all possible clutter. We propose to use a camera setup with two laterally-mounted cameras, which provide a clear and uncluttered view of the lane markings for training.

Our main contribution: The proposed deep neural network is able to reliably estimate the position of lane markings with centimeter-precision in an end-to-end approach using a classification architecture. We describe the problem formulation, the network architecture and the artificial data generation process in detail, which was employed to normalize the data distribution across the different classes.

The remainder of this paper is structured as follows: Section 2 provides an overview of recent vision based lane detection approaches. A brief summary on deep neural networks is given in 3. In Section 4 we present our approach to lane detection using laterally-mounted down-facing cameras and deep neural networks. We describe the network architecture, our labeling scheme and the artificial data generation in detail to allow reproducibility. Section 4.5 contains an extensive evaluation of our approach. In Section 5 we present a method to obtain the lane orientation using our neural network. We conclude the paper with Section 6 describing a summary and road-map for future work.

2. Related Work

Lane detection systems have been studied for more than 20 years [16], with many proposed lane detection algorithms being model-driven approaches. Popular models are splines [1], clothoids [10] or cubic polynomials [19, 24]. To reconcile the model with an input image, many existing algorithms rely on hand-crafted visual cues, like the structure tensor as employed by Loose et al. [19] and Smuda et al. [24], the bar-filter proposed by Zeng et al. [25], color based features as studied by Chiu et al. [6] or ridge features proposed by Lopez et al. [20]. Some authors combine such hand-crafted features with a traditional hough transform [26, 18]. In order to track the lane model parameter, researchers have used Particle or Kalman filters, fusing temporal information from the image and the vehicle's ego-motion [8, 15, 25]. A general overview of road or lane detection systems is given by Hillel et al. in [3].

In the recent years, several deep learning based methods for lane segmentation have been proposed [17]. Gopalan et al. evaluate the performance of neural networks in a boosting framework, combining a number of weak classifiers to detect lane markers in a sliding window approach [12]. Kim et al. propose a similar method, in which a convolution neural network is combined with the RANSAC algorithm to detect the position of lane markings [14]. Here the convolutional neural network is used merely as an image enhancement function. Huval et al. study the employment of a convolutional neural network that performs detection and classification in a single-forward pass [13]. In contrast to our approach this method only operates at 10 Hz and requires extensive post-processing to obtain the position of the lane markers. A number of deep learning based methods for road surface segmentation have been proposed by Chen et al. [5], Badrinarayanan et al.[2], Brust [4] et al. and Li et al. [17]. Although these methods seem to produce satisfactory results for pixel-wise scene segmentation, they require additional post-processing to determine the position of lane marker images.

3. Background

Deep Neural Networks have gained tremendous attention in recent years, especially since the ascent of Imagenet[9], as they have outperformed traditional machine learning approaches in challenging tasks like image classification and speech recognition. In some cases, deep neural networks have even outperformed humans at classification tasks [7]. More specifically, neural networks are computational graphs with input nodes, hidden layers and output nodes. While much of the work using deep learning has been focused on classification, there has been recent interest in extending the capability of neural networks to localizing objects in the image. Some prominent works of research include [23],[11], and [22]. However, much of this work requires image pre-processing, post-processing or is computationally intensive. We present a neural network technique that directly estimates the lane position in the image as the output, bypassing any additional processing.

Figure 1: Our setup uses laterally-mounted camera looking downward onto the lane markers

4. Lane Position Estimation

DeepLanes is a deep neural network that is designed to perform end-to-end lane detection in a simple and unified framework. In the following, we describe the camera setup, define the lane detection problem and requirements in more detail. Subsequently, we describe our network architecture and the process of generating semi-artificial data to help improve the network's performance.

4.1. Camera Setup & Pre-processing

Although virtually all proposed lane-detection systems use front-facing cameras, this camera setup might not always allow for the best possible view of the lane markings. We proposed a different location for the camera placement, to allow for a close-up and uncluttered view of the lane by using two laterally-mounted down-facing cameras as depicted in Figure 1. This setup allows us to exclude most of the scene clutter that is typically captured with front-facing cameras, thus enabling us to focus on the lane-markings only. We capture RGB images at a resolution of 240 by 360. The only pre-processing we perform is to correct the radial distortion caused by the wide-angle lens.

4.2. Problem Definition

Computer vision based solutions are used in many Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicle (AV) features which involve detection of lane positions. This is often challenging based on road conditions, shadows, spurious reflections, differences in illumination, perspective distortions based on angle of view, variations in height of the camera above the ground and many others. Additionally, there are significant differences in the appearance of lane markers based on type: double, single, broken or solid markers with a number of possible colors. While primitive computer vision techniques like adaptive thresholding of image pixels yield insufficient results, designing robust hand crafted features requires tedious mining

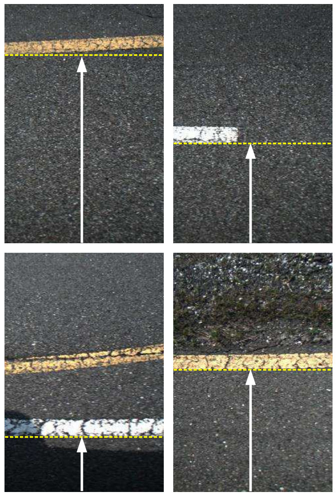


Figure 2: This Image shows our labeling scheme. The label $t_i \in [0, \ldots, 316]$ for image X_i corresponds to the row with the pixel of the lane marking that is closest to the bottom border of the image. Images which contain no lane marking are assigned to class 0. The arrows point to the image row containing the border of the marking facing the car.

and time consuming fine-tuning.

As lane markings can have variable width, specifying the exact location of a lane marking in a camera image may be ambiguous. For our camera setup, we establish a consistent definition of the lane markings position in the camera image as follows: the row of the input image that contains the pixel of the lane marking closest to the bottom border of the image, which in turn, corresponds to the car's shoulder in our camera setup. This idea is illustrated in Figure 2. Note that we also take images with multiple parallel lane markings and varying color into consideration. Consistent with this convention, in cases where parallel lane markings are visible in the image, we want to predict the position of the lane marking that is closet to the car's shoulder. Additionally, we explicitly capture and determine situations where no lane marking is visible in the image, by reserving a component in the network's output vector to handle this case.

We formulate the task of estimating the lane position as a classification problem. In the following, image i is denoted as X_i. For a given image X_i, the deep neural network computes a softmax probability output vector $Y_i = (y_0, \ldots, y_{316})$. Entry y_k in Y_i corresponds to the probability that row k in image X_i contains the position of the lane marking. We ignore a small part of the image at the top and bottom where the lane marker would only be partially visible, hence Y_i has 316 elements although the input image has a height of 360. Component y_0 in the probability vector Y_i is reserved for situations where there is no lane marking visible in the image. We denote the ground truth for image X_i as t_i.

For inferring the position of the lane marking, the input image is fed through the network and the estimated position of the lane marking e_i for image X_i is assumed to be in the image row corresponding to the entry in Y_i with the highest probability:

$$e_i = \operatorname*{argmax}_{0 \leq j \leq 316} y_j \tag{1}$$

This is an end-to-end solution because of the lack of intermediate steps. Given raw input image, the network directly yields an estimate of the lane position with respect to the vehicle. Although by this setup the network is only able to estimate the lateral displacement of the lane marker, Section 5 will present a technique to also obtain the orientation of the lane marker.

4.3. Network architecture

In the following we describe the DeepLanes network architecture, which is depicted in Figure 3. We paid attention to designing a network with minimal memory footprint and fast execution time, in order to allow for execution on embedded automotive compute platforms with scarce resources. Our neural network takes color images with a dimension of 240×360 and has the following layers.

The input is first fed through a convolutional layer, consisting of 32 filters with size of 18×18 and stride 6. Although it usually is difficult to argue or justify choices of hyperparameter in deep learning applications, for this application we chose a large filter size that roughly corresponds to half the width of a lane marking in the input image. The second convolution layer has a filter size of roughly half the previous convolution layer. Both convolution layers are followed by normalization and 3×3 pooling layer. The output of the convolution layers are fed into two fully connected layers with 2048 parameters. The dropout technique with a rate of 0.5 is applied in between the fully connected layers as a regularization measure. To obtain a probability distribution, the softmax function is applied to the output of the last fully connected layer with 317 outputs: 316 possible classes for lane positions and one class for the absence of lane marker. Rectified linear units are used as activation functions.

4.4. Artificial Data Generation

For training the network, we obtained a real world data set with over 80000 images. A view at the class histogram for the initial training set, which is depicted in Figure 4 reveals a problem with the data set: many border cases have only little representatives while the lane markings in the middle of the image dominate the histogram. We believe this is due to the natural habit of the driver trying to stay in the center of the lane while the training data was captured.

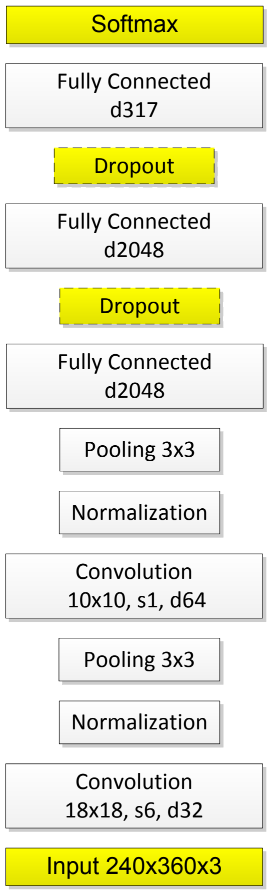


Figure 3: The DeepLanes network architecture

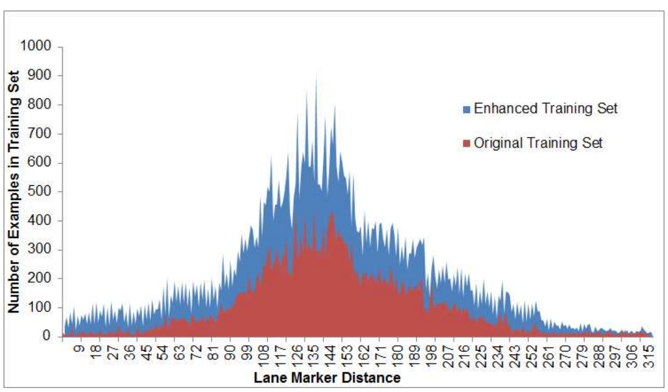
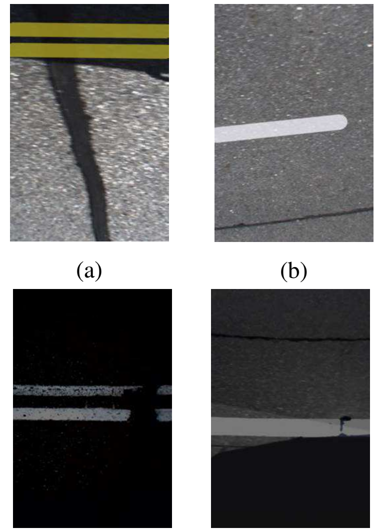


Figure 4: This plot shows the distribution of the initial training set. As only a small number of images with the lane markings at the extremeties of the image have been observed, a semi-artificial data set was needed to fill these gaps.

In order to fill these gaps in the histogram and thereby providing more diverse examples to the neural network during training, we synthesized an additional 40000 images. Mu-



(a) (b)

(c) (d)

Figure 5: This Figure shows examples of the artificially generated data. Using a real world background, various types of lane markings have been artificially placed to synthesize regular lane markings (a,b) as well as varying light conditions (c, d).

rali et al. reported a deep neural network classifier for traffic signs that has been trained on purely artificial data and generalizes well enough to classify signs from real-world images. [21]. Generating artificial data has a number of advantages. First of all, establishing a real world dataset is time consuming, cumbersome and expensive. While it is fairly easy to collect raw data recordings, it is very time consuming for humans to label the data for ground truth generation. With artificial data, the labels can be generated automatically. Besides the ability to generate arbitrary amounts of data, artificial data generation also allows us to control the class distribution. Generating artificial data for the task of lane detection with a front facing camera may be very difficult, as all types of scene clutter have to be incorporated. A major advantage of our camera setup is the fact that it allows easy generation of artificial data for augmentation.

Our data generation mechanism relies on images from the human labeled data set that contain no lane markings, on which we synthesize the lane markings We draw two numbers m, n from a uniform distribution spanning the whole range of possible lane marking positions. The randomly picked numbers are each associated with a position on the left and right borders of the image. Using the OpenCV computer vision library, we draw lines connecting the points and the line pixel values are averaged with the background pixels in order to mimic the road texture. We varied the color as well as the width of the lane markings to capture a broad variety of possible situations. Using this simple scheme, we generate a variety of scenarios including partially interrupted, parallel and scattered lane markers. Some variance in the data is induced by changing lighting conditions and shadows. This was achieved by basic image processing transformations to simulate different lighting conditions, as shown in Figure 5c. Finally, in order to mimic shadows cast on the lane markings, we reduced pixel intensities by a constant value in parts of the image, as illustrated in Figure 5d.

The final set used for training the neural network consists of $|T| = 120,000$ images, with 80000 human labeled and 40000 semi-artificial images.

4.5. Experimental Evaluation

We trained the network using the open source deep learning framework Caffe on a Nvidia Digits DevBox. The initial learning rate of $\alpha_0 = 0.5$ is decreased every four epochs by $\alpha_{i+1} = 0.8\alpha_i$. As common in deep learning applications, we used batch processing with a batch size of 128 for the gradient calculation. We trained the network for 60 epochs, which takes an average of 2.5 hours on the Digits DevBox.

Figure 6 shows grayscale versions of convolution filters that the network learned in its first layer. During training, the network learned filters that seem to respond to the typical dark-bright-dark, dark-bright and bright-dark transitions of lane markings.

For evaluating the network's performance, we used a validation data set V with $|V| = 39,000$ hand labeled images. Figure 7 shows a plot of the loss function and the accuracy over the training epochs. The softmax loss function reaches a value of 0.016 after 60 epochs. A simple way to measure the network's performance is the top-k accuracy. The top-k accuracy counts a classifier output as correct if the ground truth is among the top k predicted classes, i.e. the k components of Y_i with the highest probability. Our network reaches a final top-1 accuracy of 89.28% and a top-5 accuracy of 98.55%. For our application, the top-k accuracy measure can be deceiving, as it does not address the variance of top-5 predicted lane positions. We therefore setup an additional performance measure in Figure 8. The histogram in Figure 8a analyses in how many cases the predicted lane marking position e_i deviates less than $k \in \{0, \ldots, 9\}$ pixel from the ground truth t_i. Using the Iversion bracket $[P]$, which evaluates to 1 if P is true and to 0 if P is false, we define this measure as:

$$E_k = \frac{1}{|V|} \sum_i |e_i - t_i| [|e_i - t_i| \leq k] \quad k \in \{0, \ldots, 9\} \quad (2)$$

The evaluation shows that $E_2 = 97.85\%$ of the estimated lane positions deviate 2 or less rows from the ground truth and $E_5 = 99.04\%$ deviate 5 or less image rows. Histogram 8b shows the total error distribution $e_i - t_i$ for all images in the training set.

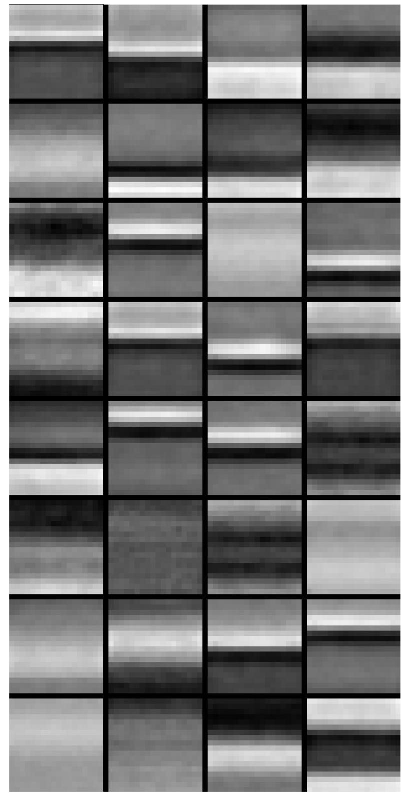


Figure 6: Filters in the first convolutional layer

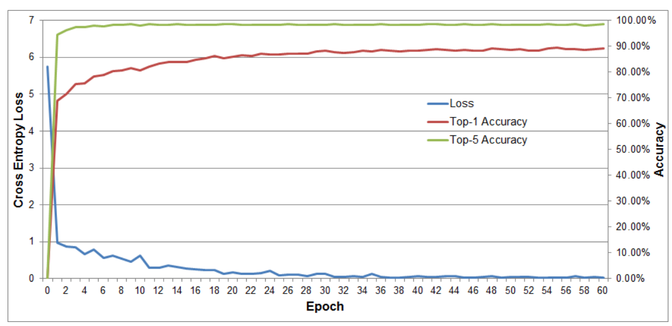
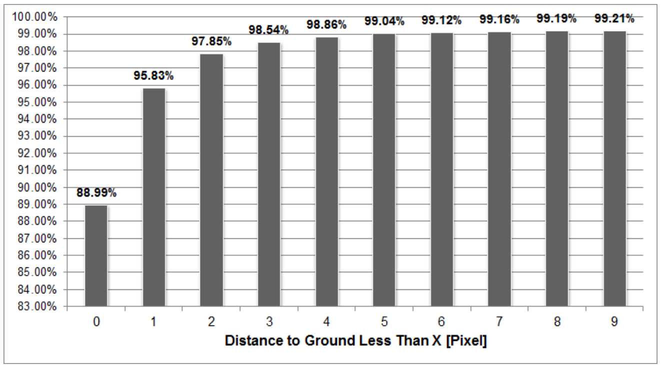
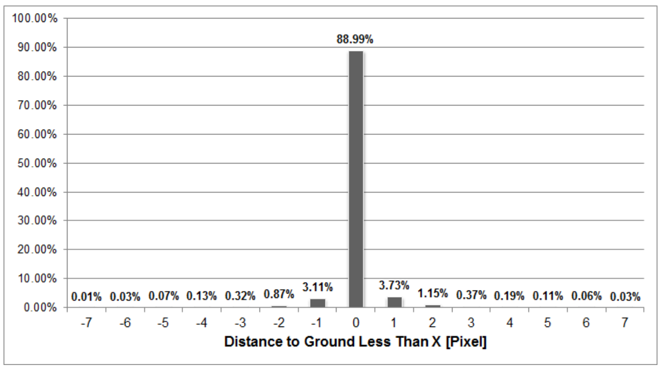


Figure 7: This plot shows the loss as well as the top-1 and the top-5 accuracy during the training of LanesNet. Note that top-1 accuracy only reaches 90% because every deviation from ground truth is penalized for misclassification, although the predicted position is of centimeter precision.

Given that for the camera setup we used for obtaining the evaluation data, the real-world distance between each image pixel only corresponds to a few millimeters, the ability to estimate the lane position can therefore be regarded as highly precise. To illustrate this, we plot a selection of images with $|e_i - t_i| > 5$ in Figure 9. Although the predicted lane position is not on the inside of the marker facing the car's shoulder, the estimated position is still very close



(a) Portion of evaluation set classified within various error bounds



(b) Histogram of deviation between ground truth and network output

Figure 8: Figure (a) shows the fraction of the data set that was classified within certain error bounds $|e_i - t_i| < k$ with $k \in \{0, \dots, 9\}$. Our network is able to predict the image row containing the lane marking in 97.85% of the images within a precision of 2 pixel, in 99.04% of the cases with 5 pixel precision. Figure (b) shows the histogram $e_i - t_i$ for all 39000 images in the evaluation set.

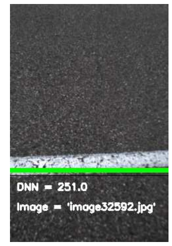
to the actual lane.

In the following, we also investigate the network's performance with respect to the Mean Absolute Error (MAE), which is defined in Equation 3.

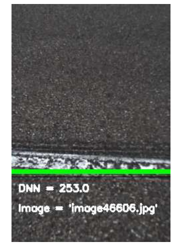
$$MAE = \frac{1}{|V|} \sum_i |e_i - t_i| \tag{3}$$

For our network, the naive Mean Absolute Error was 1.182. The standard definition given above does not consider that output y_0 is designed to classify images that contain no lane marking. In cases where $t_i = 0$ and $e_i \neq 0$ or vice versa, no sensible distance between estimated position e_i and true position t_i can be defined. In order to take these circumstances into consideration, we also report the Mean Absolute Error only for the cases where $t_i \neq 0 \wedge e_i \neq 0$:

$$MAE_2 = \frac{1}{\sum_i [t_i \neq 0 \wedge e_i \neq 0]} \sum_i |e_i - t_i| [t_i \neq 0 \wedge e_i \neq 0] \tag{4}$$



(a)



(b)

(c)

(d)

Figure 9: This figure shows examples images with a deviation from the ground truth larger than 5, i.e. $|e_i - t_i| > 5$.

The evaluation shows that the DeepLanes network has an MAE_2 of 1.59, i.e. the error is on average less than two pixel. The reason for $MAE_2 > MAE$ is due to the fact that MAE_2 does not count correctly classified images that contain no lane markings, therefore the denominator is smaller in MAE_2.

As y_0 in the network output vector Y_i is reserved for cases where the input image contains no lane marking, we also investigated the ability of the network to distinguish between images with and without a marker. To quantify the performance, we calculated the network's precision and calculated *Recall* values on these scenarios. We define true positive as $t_p = \sum_i [e_i = 0 \wedge t_i = 0]$, false negative to be $f_n = \sum_i [e_i \neq 0 \wedge t_i = 0]$ and false positive as $\sum_i [e_i = 0 \wedge t_i \neq 0]$. Recall is then defined in equation 5 and precision in equation 6.

$$P(e = 0 | t = 0) = \frac{t_p}{t_p + f_n} \tag{5}$$

$$P(t = 0 | e = 0) = \frac{t_p}{t_p + f_p} \tag{6}$$

The evaluation shows that LanesNet has a precision of 98.96% and a recall of 99.9% in detecting the presence of lane markings in the input image.

4.6. Execution Time

We evaluate the runtime of a forward pass of DeepLanes for the Nvidia DIGITS DevBox as well as on the Nvidia DrivePX. As we propose using two cameras, we evaluate the average forward pass duration for a single image as well

(a) (b) (c)

Figure 10: This figure shows the slicing technique for obtaining the lane's orientation. Figure 10a shows the sliced image and Figure 10b the network's output for each individual slice. Figure 10c shows the result for an image with a partial lane marking.

as for two images. As Table 1 shows, DeepLanes can run at an average of 100 frames/s on the DrivePX and at 500 frames/s on the DevBox, establishing real-time capability running in concurrence with other potential heavy duty algorithms.

Platform	1 Image	2 Images
Nvidia Digits DexBox	1.7 ms	1.75 ms
Nvidia DrivePX	9.4 ms	10.5 ms

Table 1: Average Forward-pass duration the DeepLanes network

5. Estimating Lane Orientation

The presented network is able to robustly estimate the row in the input image containing the lane marker nearest to the car. This information translates to the lateral distance of the car with respect to the current lane. Additionally we extract the lane orientation information as well. The lane marker can be approximated as a straight line if the region of interest in the image is small enough. As illustrated in Figure 10, we divide an input image into five slices that are resized to meet the network's input size. Estimating the position of the lane marking for each slice individually yields a set of points that can be used to calculate the lane orientation. Figure 10a and 10b show the resulting points on the individual slices. The approach can be scaled to achieve arbitrary precision by reducing or increasing the number of slices.

6. Conclusion and Future Work

We present a classification approach using a convolutional neural network that allows estimating the position of lane marker for input image with respect to the baseline of the image, when the image is taken from a laterally-mounted down-facing camera on a vehicle. Our unified framework approach is a simple, end-to-end solution that does not depend on tedious pre-processing, post-processing or hand-crafted features. As the evaluation shows, we are able to estimate the lane position in 99% of the cases with less than five pixel error in real-time on an embedded automotive platform. Our speedy and scalable technique can be applied for realtime navigation as well as robustness testing of driver-assist or automated driving features. In our future work, an application to the front-facing camera using projective geometry and exploitation of temporal and ego-motion information is contemplated.

References

[1] M. Aly. Real Time Detection of Lane Markers in Urban Streets. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 7–12, June 2008. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 2

[3] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz. Recent Progress in Road and Lane Detection: A Survey. *Machine Vision and Applications*, 25(3):727–745, 2012. 2

[4] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler. Convolutional Patch Networks with Spatial Prior for Road Detection and Urban Scene Understanding. *arXiv preprint arXiv:1502.06344*, 2015. 2

[5] X. Chen and Y. Qiao. Road Segmentation via Iterative Deep Analysis. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2640–2645, Dec 2015. 2

[6] K.-Y. Chiu and S.-F. Lin. Lane Detection using Color-based Segmentation. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 706–711, June 2005. 2

[7] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012. 2

[8] R. Danescu and S. Nedevschi. Probabilistic Lane Tracking in Difficult Road Scenarios Using Stereovision. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):272–282, June 2009. 2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2

[10] C. Gackstatter, P. Heinemann, S. Thomas, and G. Klinker. Stable Road Lane Model Based on Clothoids. In *Advanced Microsystems for Automotive Applications 2010*, pages 133–143. Springer, 2010. 2

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE conference*

Figure 12: This figure shows lane marker positions estimated by the DeepLanes network. The network is able to detect lane markers that are in the shadow as well as parallel, broken and scattered markers.

on computer vision and pattern recognition, pages 580–587, 2014. 2

[12] R. Gopalan, T. Hong, M. Shneier, and R. Chellappa. A Learning Approach Towards Detection and Tracking of Lane Markings. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1088–1098, Sept 2012. 2

[13] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng. An Empirical Evaluation of Deep Learning on Highway Driving. *CoRR*, abs/1504.01716, 2015. 2

[14] J. Kim and M. Lee. Robust Lane Detection based on Convolutional Neural Network and Random Sample Consensus. In *Neural Information Processing*, pages 454–461. Springer, 2014. 2

[15] Z. Kim. Robust Lane Detection and Tracking in Challenging Scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):16–26, March 2008. 2

[16] K. Kluge and S. Lakshmanan. A Deformable-Template Approach to Lane Detection. In *Intelligent Vehicles '95 Symposium., Proceedings of the*, pages 54–59, Sep 1995. 1

[17] J. Li, X. Mei, and D. Prokhorov. Deep Neural Network for Structural prediction and Lane Detection in Traffic Scene. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–14, 2016. 2

[18] G. Liu, F. Wrgtter, and I. Markeli. Combining Statistical Hough Transform and Particle Filter for Robust Lane Detection and Tracking. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 993–997, June 2010. 2

[19] H. Loose, U. Franke, and C. Stiller. Kalman Particle Filter for Lane Recognition on Rural Roads. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 60–65, June 2009. 2

[20] A. López, J. Serrat, C. Cañero, F. Lumbreras, and T. Graf. Robust Lane Markings Detection and Road Geometry Computation. *International Journal of Automotive Technology*, 11(3):395–407, 2010. 2

[21] V. N. Murali, A. Micks, M. J. Goh, and D. Liu. Generation and Usage of Virtual Data for the Development of Perception Algorithms Using Vision. Technical report, SAE Technical Paper, 2016. 4

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *arXiv preprint arXiv:1506.02640*, 2015. 2

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv preprint arXiv:1312.6229*, 2013. 2

[24] P. Smuda, R. Schweiger, H. Neumann, and W. Ritter. Multiple Cue Data Fusion with Particle Filters for Road Course Detection in Vision Systems. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 400–405, 2006. 2

[25] Z. Teng, J. H. Kim, and D. J. Kang. Real-Time Lane Detection by Using Multiple Cues. In *Control Automation and Systems (ICCAS), 2010 International Conference on*, pages 2334–2337, Oct 2010. 2

[26] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong, and H. Chen. A Novel Lane Detection based on Geometrical Model and Gabor Filter. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 59–64, June 2010. 2