

End-to-End Deep Learning of Lane Detection and Path Prediction for Real-Time Autonomous Driving

Der-Hau Lee and Jinn-Liang Liu

Abstract—Inspired by the UNet architecture of semantic image segmentation, we propose a lightweight UNet using depthwise separable convolutions (DSUNet) for end-to-end learning of lane detection and path prediction (PP) in autonomous driving. We also design and integrate a PP algorithm with convolutional neural network (CNN) to form a simulation model (CNN-PP) that can be used to assess CNN’s performance qualitatively, quantitatively, and dynamically in a host agent car driving along with other agents all in a real-time autonomous manner. DSUNet is 5.16x lighter in model size and 1.61x faster in inference than UNet. DSUNet-PP outperforms UNet-PP in mean average errors of predicted curvature and lateral offset for path planning in dynamic simulation. DSUNet-PP outperforms a modified UNet in lateral error, which is tested in a real car on real road. These results show that DSUNet is efficient and effective for lane detection and path prediction in autonomous driving.

Index Terms—Deep learning, convolutional neural network, depthwise separable convolutions, lane detection, path prediction, autonomous driving.

I. INTRODUCTION

Lane detection is an essential task of advanced driver assistance systems (ADAS) in modern vehicles and autonomous driving systems in self-driving cars for lane keeping assistance, departure warning, and centering [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. There are two major paradigms, namely, classic computer vision and deep learning, for lane detection [9]. The classic computer vision paradigm requires intensive feature engineering, road modeling, and special case handling and is thus not robust enough to deal with seemingly infinite driving situations, environments, and unexpected obstacles [1]. The deep learning paradigm has shown considerable progress to alleviate these difficulties in recent years [9].

Path prediction (lane following) is also a crucial task of ADAS and self-driving vehicles for active driving assistance and lateral and longitudinal controls [4], [5], [6], [7]. It depends on various factors of vehicles and road and traffic environment such as ego vehicle’s speed and steering angle, road geometry, and the other closest in-path vehicle [4], [5], [6], [7]. It also involves perception, sensory, and control systems in the ego vehicle [5], [4].

UNet [11] is an encoder-decoder convolutional neural network (CNN) [12] for semantic segmentation. It achieved remarkable success in many competitions [13] due to its

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2115-M-007-011-MY2.

D. Lee was with the Department of Electrophysics, National Chiao Tung University, Hsinchu 300, Taiwan.

J. Liu is with the Institute of Computational and Modeling Science, National Tsing Hua University, Hsinchu 300, Taiwan (e-mail: jinnliu@mail.nd.nthu.edu.tw, website: <http://www.nhcue.edu.tw/~jinnliu>).

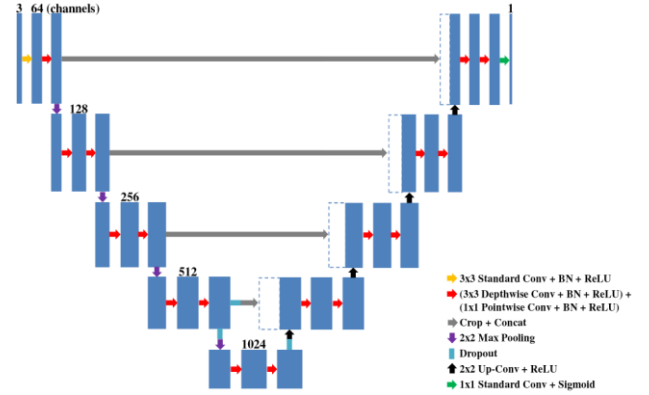


Fig. 1. DSUNet replaces 17 3x3 standard conv layers in UNet [11] with 17 3x3 depthwise and 17 1x1 pointwise conv layers shown by red arrows in the architecture with 3 dropouts by light-blue bars. Other operations remain the same. The ratios of DSUNet/UNet are 40/23 conv layers, 6.01/31.04 M parameters, 9.56/62.51 B MACs, and 45/28 FPS in inference.

elegant architecture of symmetrical contracting (encoding) and expanding (decoding) sub-networks with skip connections that combine semantic (global) and appearance (local) information from both sub-networks [14].

The accuracy and efficiency of CNNs are equally important in a variety of applications such as robotics, mobile devices, and self-driving cars [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. The concept of depthwise separable convolutions [25] (a depthwise convolution followed by a pointwise convolution [16]) has been used to construct many efficient CNNs [15], [16], [17], [18], [19], [21], [22]. We apply this concept to UNet [11] (DSUNet) for semantic road segmentation and lane detection and integrate DSUNet (or other CNNs) with path prediction (PP) algorithm to form a simulation model (CNN-PP) that can be used to assess CNN’s performance qualitatively, quantitatively, and dynamically in a host agent car driving along with other agents all in a real-time autonomous manner [24], [26].

The small segmentation network DSUNet not only has real-time performance with merits on energy and memory consumption but also attains sufficient accuracy in dynamic traffic. We show that DSUNet is 5.16x lighter and 1.61x faster than UNet with comparable accuracy.

II. RELATED WORK

Fig. 1 illustrates the architecture of both the original UNet [11] and DSUNet. DSUNet replaces 17 3x3 convolutional (conv) layers in UNet with 17 3x3 depthwise and 17 1x1 pointwise conv layers shown by red arrows in the figure with 3 dropouts by light-blue bars. The batch normalization and padding method of [2] are used in DSUNet. The input and

output channels of DSUNet/UNet are 3/1 and 1/2, respectively, due to different input images (color/black and white) and output loss functions (weighted cross entropy/softmax). Other operations remain the same. The ratios of DSUNet/UNet are 40/23 conv layers, 6.01/31.04 M parameters, 9.56/62.51 B multiplication accumulation operations (MACs) with the same I/O of DSUNet, and 45/28 frames per second (FPS) in inference on a PC with Intel i9-9900K and an NVIDIA RTX 2080ti.

In the past few years, UNet have been modified or extended for its use in various applications [13] such as 3D volumetric segmentation [27], 3D stroke lesion segmentation [28], retinal vessel segmentation [29], [30], and photoacoustic imaging [31]. Bruls et al. first applied UNet to road marking detection with batch normalization after every convolutional layer [2]. Zou et al. replaced the bottom layer of UNet with a convolutional long short-term memory network to form UNet-ConvLSTM, where UNet features of continuous multiple frames are inputs to LSTM in time series for feature learning and lane prediction [9]. UNet-ConvLSTM improves inference accuracy significantly, especially for road images with worn-up or occluded lane markings.

In addition to CNN's performance in accuracy, its computational complexity is a critical concern in self-driving cars [20], [23]. Depthwise separable convolutions [25] separate the operations of spatial and channel correlations [16] using single-channel $k \times k$ and 1×1 filters [15], respectively, and thus greatly reduce parameters [15], [16]. This idea inspired a variety of efficient models such as SqueezeNet, Xception, MobileNet, and ShuffleNet derived from AlexNet, GoLeNet, or ResNet for classification or object detection [15], [16], [17], [18].

From UNet for semantic segmentation, Beheshti and Johnsson designed Squeeze-UNet [21] by replacing the contracting and expanding layers with fire modules inspired by SqueezeNet [15]. The ratios of Squeeze-UNet/UNet are 0.8 to 0.98 in accuracy for five classes (car, road, tree, building, and sky), 1/12 in model size MB, 1/3.2 in MACs, and 0.83 and 0.48 in inference and training time, respectively. The main differences between our and their model are the ratios (DSUNet/Squeeze-UNet) 0.46 and 1.33 in MACs and the speed of inference, respectively, i.e., DSUNet is 1.33x faster. We use standard depthwise separable convolutions [15], [16], [17] instead of fire modules that repeatedly squeeze (using 1×1 filters) and expand (mixing 1×1 and 3×3 filters) conv layers.

Gadosey et al. also applied DS convolutions with weight standardization and group normalization to UNet (SD-UNet) for semantic segmentation of biomedical images, where SD is short for stripping down [22]. Their results show that the ratios of SD-UNet/UNet are 0.97/0.98 for accuracy, 3.9/31 M parameters, 7.8/62.04 floating point operations, and 91/56 in inference speed. The ratios of DSUNet/SD-UNet are 1 for accuracy, 1.53 in parameters, 1.22 in operations, and 0.99 in speed. SD-UNet is much lighter than DSUNet in computational complexity. Nevertheless, the performance of these two models in accuracy and speed is almost the same

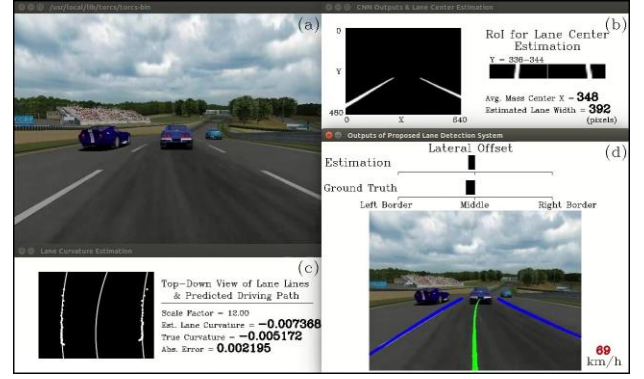


Fig. 2. Real-time simulation architecture of lane detection and path prediction by DSUNet-PP from a host car's view in TORCS traffic: (a) input image, (b) lane detection, (c) line clustering and perspective transformation, and (d) lateral offset estimation, inverse perspective transformation, line (blue) fitting, and driving path (green) prediction

on the baseline of UNet for two different applications.

Path prediction models are generally polynomials of lane centerline having few coefficients determined by heading and lane curvature without [6] or with considering vehicle's transient dynamics such as yaw rate and lateral and longitudinal velocities [5]. Cudrano et al. [6] proposed an integrated NNPP model that uses vehicle's heading and lateral displacement calculated from the lane detection output of a modified UNet to determine a cubic polynomial. They tested the model in a real car driving on two racetracks (one is more curvy and the other is less) without other cars and with a maximum speed of 54 km/h. They tested PP algorithm with results but did not present the overall architecture of NN-PP model, the details of the modified UNet, and its performance on lane detection.

We proposed a deep learning model in [24] that combines multi-task CNN and control algorithms for multiple actions in autonomous driving (steering, acceleration, braking, lane changing, and overtaking). We also modified The Open Racing Car Simulator (TORCS) [26], [24], [32] on which the model was implemented for a host car (Host) efficiently inferring and stably driving with other autonomous cars (Agents) in dynamic TORCS traffic [24], [26]. We extend here our TORCS to include lane detection and path prediction in order to test and evaluate the overall performance of deep learning, lane detection, and path prediction algorithms in the simulator before deploying them to real cars in real world traffic.

A. Contributions

Scarce literature exists on lightweight CNN and path prediction algorithms for self-driving cars to perform lane detection, road recognition, and lane following in dynamic traffic [6], let alone the overall performance of these algorithms in a real-time autonomous manner. The contributions of the present work are summarized as follows:

(i) We propose DSUNet, a lightweight CNN, that extends the original UNet from classification and object detection to semantic segmentation and improves UNet by

5.12x, 6.54x, and 1.61x in model size, MACs, and FPS, respectively.

(ii) We propose an end-to-end CNN-PP model for simulating self-driving cars and evaluating the dynamical performance of CNN and PP algorithms in real time.

(iii) We show that DSUNet-PP outperforms UNet-PP in both static and dynamic mean average errors of predicted curvature and lateral offset (vehicle center to lane center [33]) for path planning in dynamic traffic of self-driving agents with lane changes and a max speed of 70 km/h. (iv) We show that DSUNet-PP outperforms the modified UNet of [6] in lateral error on different (real-world [6] vs. curvier artificial) roads by different (real [6] vs. artificial) cars.

III. DEEP LEARNING OF LANE DETECTION AND PATH PREDICTION

Fig. 2 illustrates the architecture of our deep learning, lane detection, and path prediction algorithms implemented in TORCS from Host's view with several Agents in dynamic traffic. The architecture consists of (a) images from Host's front-facing camera, (b) gray lane images obtained from a CNN (DSUNet for example in Fig. 2) and their region of interest (RoI) for lane localization and centering, (c) a perspective mapping (PM) of lane lines and their clustering from camera's view (CV) to top-down view (TDV) for path prediction and curvature estimation, and (d) the inverse PM lane lines (blue) and prediction path (green) with lateral offset estimation from TDV back to CV. More details about these four sub-figures and corresponding methods are given below.

To handle the unbalanced numbers of positive (lane line) and background samples, we use the weighted cross entropy loss [34]

$$L = - \frac{N}{P+N} \sum_{y_c=1} \log(\sigma(x_c)) - \frac{P}{P+N} \sum_{y_c=0} \log(1 - \sigma(x_c)) \quad (1)$$

for training CNNs, where x_c is a sample score, y_c is the corresponding label, σ is the sigmoid function, and $P(N)$ is the number of positive (negative) samples in a batch of images.

A. Lateral offset estimation

Cameras mounted on vehicles vary in height and lateral location so do lane images [10]. We first resize the output gray (probability) images to the original size 640x480 and then choose the region of interest (RoI shown in Fig. 2b) of gray images in the range of $y = 336 \sim 344$ for lane centering by calculating the centroid coordinate x . The coordinates of left ($x_l = 0 \sim 320$) and right ($x_r = 321 \sim 639$) parts of gray images are x_l and x_r , respectively. The lane center is estimated as $(x_l + x_r)/2$ and the lane width is $x_r - x_l$. Since the image center is at $x = 320$, the distance from Host to the lane center is estimated by $\Delta_0 = 320 - (x_l + x_r)/2$.

However, this approach tends to overestimate the true distance to the lane center since RoI is ahead of Host and depends on camera's position. We use $\Delta = \alpha \Delta_0$ with $\alpha = 0.6$ as

an estimated offset to the true lane center shown in Fig. 2d. This lane centering is thus a CNN-based method.

B. Driving path prediction

The following path prediction algorithm comprises wellknown or advanced post-processing methods that are particularly suitable for output images having local and global properties to deal with complexity, scalability, and homography issues [8], [10].

Path Prediction Algorithm:

Step 1. We use DBSCAN [1] to cluster detected line segments (dots in Fig. 2c).

Step 2. A perspective mapping [8], [10] then transforms semantic lanes from camera's view to top-down view space to which quadratic polynomials [3] apply for fitting parallel lane lines (Fig. 2c) by using recursive least squares [6] and imposing parallelism.

Step 3. Another quadratic polynomial $f(x)$ for the desired driving path is uniquely determined by three conditions, i.e., the starting (x_s) and final (x_f) points of the middle line (Fig. 2c) of two parallel lines and its tangent at x_s as that of the parallel lines. We thus obtain the TDV lane curvature $\kappa_0 = f''(x_s)/[1 + f'(x_s)]^{3/2}$ at x_s in 1/m unit. The true curvature $\kappa = \beta \kappa_0$ in the real world space with a scale factor $\beta = 12$ is then used to calculate its error compared to the ground truth curvature of TORCS track as shown in Fig. 2c.

Step 4. Finally, the inverse PM transforms these three lines back to the CV space for visual verification and lateral estimation shown in Fig. 2d.

In Step 1, we use five consecutive gray images to obtain averaged line segments for possible occluded or worn-out markings. Higher-order polynomials and corresponding PM transformations are generally needed in Steps 2 and 3 for curvy or up-and-down roads in real world [10].

IV. EXPERIMENTAL SETUP

Lane segmentation results predicted by a CNN are expressed in pixel-wise true positive, true negative, false positive, and false negative values which are paired with ground truth values of lane lines in images. Its performance is measured in accuracy, precision, recall, and F1 score [9].

The image data of this work consist of artificial (TORCS) and real-world (LLAMAS [35] and TuSimple [36]) images summarized in Table 1, where TuSimple includes images augmented by mirroring in training phase. The ratio between the numbers of training and test images is about 10. We annotate TORCS images (collected by Host driving on Tracks 1 to 6 [26]) to binary images of ego-lane lines, which are similar to those annotated from real-world images [35]. A second test set of 3,200 images from curvy Track 7 [26] is used to evaluate the overall performance (Figs. 2b-2d) of the integrated CNN-PP model from Host's perspective in real-time and dynamic traffic with other Agents.

UNet and DSUNet are trained on separate TORCS, LLAMAS, TuSimple, and TuSimple-mirror images with Adam optimizer, a batch size of 1, momentum 0.9, and learning rate 10^{-4} and 10^{-5} for first 75 and last 25 epochs, respectively. The trained models are evaluated on the first test set for their

performance of lane detection. We then evaluate the static and dynamic performance (as shown in Fig. 2) of the integrated

TABLE I

IMAGE DATA OF THIS WORK			
Dataset	TORCS	LLAMAS	TuSimple
No. of images	42747	22714	6408
Labels	ego-lane lines		multi-lane lines
Source	this work	[35]	[36]

TABLE II

PERFORMANCE OF UNET AND DSUNET FOR LANE DETECTION ON TESTING DATA

TORCS				
	Accuracy	Precision	Recall	F1 score
UNet	0.995	0.933	0.911	0.921
DSUNet	0.990	0.902	0.833	0.858
LLAMAS				
	Accuracy	Precision	Recall	F1 score
UNet	0.986	0.789	0.910	0.843
DSUNet	0.984	0.768	0.902	0.825
TuSimple				
	Accuracy	Precision	Recall	F1 score
UNet	0.974	0.873	0.758	0.809
DSUNet	0.971	0.817	0.786	0.796
TuSimple-mirror				
	Accuracy	Precision	Recall	F1 score
UNet	0.974	0.821	0.826	0.822
DSUNet	0.967	0.750	0.841	0.789

CNN-PP model in TORCS for Host driving with 6 Agents on Track 7 having a total length of 3919 m. All cars drive autonomously by default control algorithms [24] with the maximum speed of 70 and 67 km/h for Host and Agents, respectively, as displayed in Fig. 2d.

V. RESULTS

Table 2 shows that DSUNet is comparable to UNet in four measures for lane detection on almost all images (up to 96% in average F1) in the first test dataset as illustrated in Fig. 3, for example.

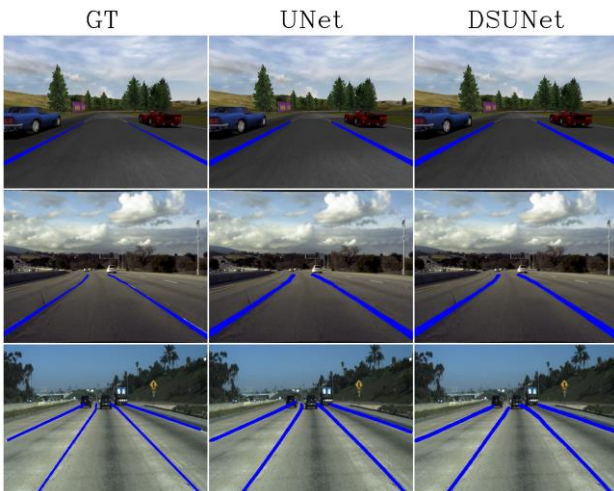


Fig. 3. Comparison of UNet (second column) and DSUNet (third column) in lane detection with ground truth (first column) on TORCS (first row), LLAMAS [35] (second row), and TuSimple [36] (third row) images

TABLE III

PERFORMANCE OF UNET-PP AND DSUNET-PP IN STATIC AND DYNAMIC ERRORS (sMAE/dMAE) WITH THE PERCENTAGE (%) OF AVAILABLE FRAMES (Avail) ON TRACK 7 [26]

	UNET-PP	DSUNET-PP
sMAE/dMAE		
κ (1/m)	0.0048/0.0052	
Δ (m)	0.0046/0.0049	
	0.1909/0.1125	
	0.1771/0.1018	
	Avail (%)	
κ	100/100	100/100
Δ	95.68/94.06	83.50/82.06

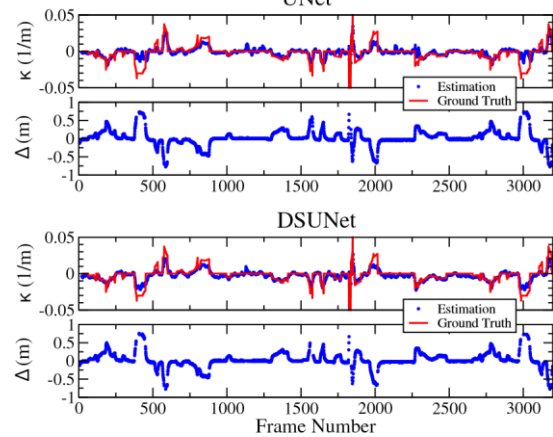


Fig. 4. Estimated curvatures (κ) and distances to lane center (Δ) in blue by UNet-PP and DSUNet-PP with ground-truth in red on a sequence of frames from Track 7

Table 3 shows the overall performance of UNet-PP and DSUNet-PP measured in the static and real-time dynamic mean absolute errors (sMAE/dMAE) [24] of predicted curvature κ (in 1/m) and lane center distance Δ (m) on the path traversed by Host in TORCS traffic with other Agents. Here, dMAE means that MAE is dynamically calculated while Host is in motion as shown in Fig. 2c in contrast to sMAE from recorded images. The percentage (%) of available frames (Avail) [6] from Host's videos for predicting Δ is less than 100 due to shadows or occlusions on the road (RoI in Fig. 2b is not available) and is worse in dynamic than in static conditions.

DSUNet is surprisingly better than UNet in both sMAE and dMAE for both κ and Δ although it is much smaller in model size. Its available frames for calculating Δ dMAE have been reduced by 19.03%. We attribute its better performance to more missing frames for calculating Δ as these obscure frames would increase its Δ MAE if they were accounted for. Δ dMAE is smaller than Δ sMAE for the same reason. Δ and κ MAEs are correlated in CNN-PP model.

Table 3 thus shows important effects of CNN's perception in motion on path planning in autonomous driving. Our model performs better than that of [6] (cf. Table 3), i.e., 10.18 vs. 45.3 cm in Δ dMAE in centered driving style on different (artificial vs. real-world) roads by different (artificial vs. real) cars at

different speeds (70 vs. 54 km/h). Cudrano et al. [6] also modified UNet and integrated it in a perception pipeline that includes Δ calculations but without using learning methods for lateral corrections like our lateral offset estimation.

Fig. 4 shows the ground-truth (red) and estimated (blue dotted) curves of κ and Δ in sMAE with respect to the frame number of a video from Track 7 recorded by Host driving alone in the middle lane, where each point of the curves is an average of 11 values of κ or Δ from consecutive frames. The κ and Δ curves are reciprocal in peaks and valleys as in their units. The curvature (κ) at current frame (time) is estimated by the one-dimensional Kalman filter [38] using consecutive 15 noisy data points as curvature measurements, which are numerical values from Step 3 in the PP algorithm, where the covariance of measurement and process noises is 0.01 and 0.0001, respectively. DSUNet-PP outperforms UNet-PP in deviation from ground-truth values on several curvy sections (near frame numbers 400, 2200 and 3000, for example).

The simulation track is more curvy than real or experimental roads [10] by one order of magnitude. The minimum radius of general roads is about 130-160 m [37] corresponding to a maximum curvature of 0.00625-0.0076 m⁻¹. Table 3 and Fig. 4 thus postulate that DSUNet-PP may reduce the dynamic error of its lateral (Δ) prediction to 5 cm on real roads in future studies. Two simulation videos are available on our website for a qualitative and quantitative review on the dynamic performance of UNet-PP and DSUNet-PP from Host driving on Track 7 with other Agents and lane changes in real time.

VI. CONCLUSION

Based on UNet and depthwise separable convolutions, we proposed a lightweight DSUNet for lane detection, road recognition, and path prediction (PP). We also proposed a model that integrates CNN and PP algorithms and developed a simulator to evaluate the dynamic performance of the model qualitatively as well as quantitatively from a host car driving autonomously with other cars and lane changes. Our results show that (i) DSUNet improves UNet by 5.12x, 6.54x, and 1.61x in model size, MACs, and FPS, respectively, (ii) DSUNet-PP outperforms UNet-PP in both static and dynamic mean average errors of predicted curvature and lateral offset for path planning in dynamic traffic, and (iii) DSUNet-PP outperforms the modified UNet of [6] in lateral error on different (artificial vs. real-world [6]) roads by different (artificial vs. real) cars.

REFERENCES

- [1] Huval, B., et al.: An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716 (2015)
- [2] Bruls, T., Madder, W., Morye, A.A., Newman, P.: Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data. IEEE International Conference on Robotics and Automation (ICRA), pp. 1863-1870 (2018)
- [3] Garnett, N., et al.: 3D-LaneNet: End-to-end 3D multiple lane detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2921-2930 (2019)
- [4] Kim, W., et al.: Vehicle path prediction using yaw acceleration for adaptive cruise control. IEEE Transactions on Intelligent Transportation Systems. 19, 3818-3829 (2018)
- [5] Lin, C.F., Ulsoy, A.G., LeBlanc, D.J.: Vehicle dynamics and external disturbance estimation for vehicle path prediction. IEEE Transactions on Control Systems Technology. 8, 508-518 (2000)
- [6] Cudrano, P., et al.: Advances in centerline estimation for autonomous lateral control. In: IEEE Intelligent Vehicles Symposium (IV), pp. 14151422 (2020)
- [7] Badue, C., et al.: Self-driving cars: A survey. Expert Systems with Applications. 165, 113816 (2021)
- [8] Li, X., Li, J., Hu, X., Yang, J.: Line-CNN: End-to-end traffic line detection with line proposal unit. IEEE Transactions on Intelligent Transportation Systems. 21, 248-258 (2020)
- [9] Zou, Q., et al.: Robust lane detection from continuous driving scenes using deep neural networks. IEEE Transactions on Vehicular Technology. 69, 41-54 (2020)
- [10] Lu, P., et al.: SUPER: A novel lane detection system. IEEE Transactions on Intelligent Vehicles. 6, 583-593 (2021)
- [11] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234-241 (2015)
- [12] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science. 313, 504-507 (2006)
- [13] Liu, L., et al.: A survey on U-shaped networks in medical image segmentations. Neurocomputing. 409, 244-258 (2020)
- [14] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440 (2015)
- [15] Iandola, F.N., et al.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv preprint arXiv:1602.07360 (2016)
- [16] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251-1258 (2017)
- [17] Howard, A.G., et al.: MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
- [18] Zhang, X., Lin, M., Sun, J.: ShuffleNet: An extremely efficient convolutional neural network for mobile devices. arXiv preprint arXiv:1707.01083 (2017)
- [19] Santos, A.G., et al.: Reducing SqueezeNet storage size with depthwise separable convolutions. In: International Joint Conference on Neural Networks (IJCNN), pp. 1-6 (2018)
- [20] Hussain, R., Zeadally, S.: Autonomous cars: Research results, issues, and future challenges. IEEE Commun. Surv. Tutor. 21, 1275-1313 (2019)
- [21] Beheshti, N., Johnsson, L.: Squeeze U-Net: A memory and energy efficient image segmentation network. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1495-1504 (2020)
- [22] Gadosey, P.K., et al.: SD-Unet: Stripping down U-net for segmentation of biomedical images on platforms with low computational budgets. Diagnostics. 10, 110 (2020)
- [23] Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G.: A survey of deep learning techniques for autonomous driving. J. Field Robotics. 37, 362386 (2020)
- [24] Lee, D.-H., et al.: Deep learning and control algorithms of direct perception for autonomous driving. Applied Intelligence. 51, 237-247 (2021)
- [25] Sifre, L.: Rigid-motion scattering for image classification. Ph. D. thesis, Ecole Polytechnique, Palaiseau, France (2014)
- [26] Chen, C., Seff, A., Kornhauser, A., Xiao, J.: DeepDriving: Learning affordance for direct perception in autonomous driving. In: IEEE International Conference on Computer Vision (ICCV), pp. 2722-2730 (2015)
- [27] Cicek, O., et al.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: MICCAI 2016: Medical Image Computing and Computer-Assisted Intervention, pp. 424-432 (2016)
- [28] Rodriguez-Sanchez, A., Tureckov, A., A.: Isles challenge: U-shaped convolution neural network with dilated convolution for 3d stroke lesion segmentation. In: International MICCAI Brainlesion Workshop, pp. 319-327 (2018)
- [29] Xiao, X., Lian, S., Luo, Z., Li, S.: Weighted Res-UNet for highquality retina vessel segmentation. In: 9th International Conference on Information Technology in Medicine and Education (ITME), pp. 327331 (2018)
- [30] Li, R., Li, M., Li, J., Zhou, Y.: Connection sensitive attention U-NET for accurate retinal vessel segmentation. arXiv preprint arXiv:1903.05558 (2019)

- [31] Guan, S., Khan, A.A., Sikdar, S., Chitnis, P.V.: Fully dense UNet for 2-D sparse photoacoustic tomography artifact removal. *IEEE Journal of Biomedical and Health Informatics*. 24, 568-576 (2020)
- [32] Wymann, B., et al.: TORCS: The open racing car simulator. (2000) [33] Jung, C.R., Kelber, C.R.: A lane departure warning system using lateral offset with uncalibrated camera. In: *Proceedings IEEE Intelligent Transportation Systems*, pp. 102-107 (2005)
- [34] Wang, X., et al.: ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 3462-3471 (2017)
- [35] Behrendt, K., Soussan, R.: Unsupervised Labeled Lane Markers Using Maps. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 832-839 (2019)
- [36] TuSimple Competitions for CVPR2017. [https:// github.com/TuSimple /tusimple-benchmark](https://github.com/TuSimple/tusimple-benchmark). Accessed 11 Aug 2021
- [37] Fitzpatrick, K.: Horizontal Curve Design: An Exercise in Comfort and Appearance. *Transportation Research Record*. 1445, 47-53 (1994)
- [38] Welch, G., Bishop, G.: An introduction to the Kalman filter. Univ. North Carolina, Chapel Hill, NC, USA, Lecture (2001)