2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops

End-to-End Ego Lane Estimation based on Sequential Transfer Learning for Self-Driving Cars

Jiman Kim, Chanjong Park
Software Center, Samsung Electronics
Seoul R&D Compus, Republic of Korea
{jiman14.kim, cj710.park}@samsung.com

Abstract

Autonomous cars establish driving strategies using the positions of ego lanes. The previous methods detect lane points and select ego lanes with heuristic and complex postprocessing with strong geometric assumptions. We propose a sequential end-to-end transfer learning method to estimate left and right ego lanes directly and separately without any postprocessing. We redefined a point-detection problem as a region-segmentation problem; as a result, the proposed method is insensitive to occlusions and variations of environmental conditions, because it considers the entire content of an input image during training. Also, we constructed an extensive dataset that is suitable for a deep neural network training by collecting a variety of road conditions, annotating ego lanes, and augmenting them systematically. The proposed method demonstrated improved accuracy and stability on input variations compared with a recent method based on deep learning. Our approach does not involve postprocessing, and is therefore flexible to change of target domain.

1. Introduction

Deep learning understands the world by analyzing the context of a scene, then focusing on important objects and observing them at a hierarchy of levels, from narrow with high resolution, to broad with low resolution. Therefore, when understanding a scene, deep learning is relatively insensitive to variations of environmental condition, and is inexpensive to redesign, to respond to different targets. For deep learning to achieve high accuracy, it needs a large amount of high-quality data. Therefore, much of the progress in deep learning, specifically in supervised deep-network learning, can be attributed to the availability of huge image datasets such as ImageNet [42], ActivityNet [18], MS COCO [32], Open Images [25], YouTube-8M [1], and YouTube-BB [39]. Recently, in the field of autonomous driving, various datasets have been published including CamVid [9], KITTI [16], TORCS [10], GTA5 [40], Cityscapes [12], and SYNTHIA [41]. Those datasets focus on high-level scene understanding by semantic segmentation. Semantic segmentation is essential for an autonomous driving intelligence to understand the complex situations of a scene. Segmenting objects, including static and moving objects, in a scene means that we can simultaneously detect and classify all objects; the results can be used to analyze the properties of objects and the relations between them. This paper presents a method to adopt abundant semantic information for ego lane estimation using newly constructed datasets and networks developed for high-level scene segmentation (Fig. 1). This method is effective because considering the complete scene, rather than parts of it, reduces the deep networks sensitivity to problems such as occlusion by vehicles and pedestrians, rough road condition, blurred lane markings, low illumination at the evening, and other variations of road surface due to environmental conditions.

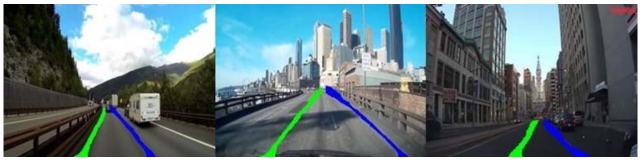

Figure 1. The proposed method transfers the learned representations of a deep network through sequential representation domain change and reduction. The transferred network extracts left and right ego lanes directly and separately from input image without any postprocessing.

The contributions of this paper are: 1) an end-to-end estimation process of left and right ego lanes from an input image directly and separately using sequential transfer learning, without any postprocessing, 2) a semi-automatic ego lane annotation tool to reduce the effort required to construct a new dataset for our end-to-end approach, and 3) a large dataset construction with extensive data augmentation to train a deep neural network. We present related work in

2160-7516/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPRW.2017.158

Section 2, details of our approach in Section 3, experimental results in Section 4 and a conclusion in Section 5.

2. Related Work

Model-driven approaches. Many papers have reviewed lane detection based on model-driven approaches [50, 19, 27]. Most of the methods involve stages such as preprocessing, feature extraction, line fitting and lane-parameter estimation. The problem of designing and combining hand-crafted features with the model is very important. Mainly three features (edge, color, texture), are defined by four types of model [27]. The first type finds strong edge components in an image and uses the combination of a Sobel filter and Hough Transform [51, 3, 5, 31, 13, 28] to analyze the strong edge direction. The accuracy of the original Hough Transform was improved by the Statistical Hough Transform that uses a multiple kernel density to describe the distribution of the Hough variables without edge preprocessing [33, 34, 35]. Some researchers used IPM to change the point of sight and found all straight lines [2, 6, 33, 7]. The second type observes the change of feature values. To detect lanes, these method detect a large change of intensity [24, 22] or measure the positive and negative second derivatives of edge components [30]. The third type analyzes the primary color or direction of lane components. These methods segregate pixels by exploiting primary color information of lanes [11, 52, 8], or extract lanes by clustering straight lines that point in similar directions and removing outliers [47]. The fourth type defines models with specific shapes, then performs model fitting on a feature image. For example, various models that represent straight or curved lines (e.g., linear, parabolic, B-spline), are each matched to a sub-window that has feature values extracted on geometric constraints [53, 46, 48, 49]. All of these model-driven approaches use hand-crafted features that is elaborately designed dependent on target's properties and need heuristic and complex postprocessing with strong geometric assumptions to determine the positions of ego lanes. Therefore, if the target is change, the design of features should be modified.

Approaches based on deep learning. To overcome limitations of model-driven approaches, recent research has adopted deep learning, specifically convolutional neural networks (CNNs), for lane detection. One method extracts lane candidate regions and uses the RANSAC algorithm to remove outliers and to perform line fitting [23], but because the CNN uses an edge image as input, the method's accuracy is directly affected by that of the edge-detection algorithm, which is sensitive to intensity variations and occlusion; in the paper, the CNN was only to extract features, and did not consider full context of an input image. Another method predicts two end points of a local lane segment in a sliding window by regression using a CNN [20]; it uses local context of a scene by considering occlusion cases to obtain ground truth. Another combines multi-task CNN and RNN to detect lane boundaries [29]. To select ego lanes and separate left and right lanes, the two previous methods performed postprocessing including DBSCAN, line clustering, and heuristic selection. A different approach uses use two laterally-mounted down-facing cameras to estimate the position of lanes with sub-centimeter accuracy [17], but because of the orientation of the cameras the method cannot exploit all of the information in the scene. Another approach adds an expansion network to a CNN and trains the network end-to-end for estimation of ego lane [38]. To achieve best trade-off between segmentation quality and runtime, several architecture refinements were added, but the method cannot estimate exact left and right ego lanes (two side-curves) because it consider a region that is surrounded by the two side-curves. Our proposed method extracts the left and right ego lanes directly and separately from an input image and utilizes all information in the front road scene by an end-to-end technique to improve accuracy.

3. Semantic Ego Lane Estimation

3.1. Problem Redefinition

Model-driven methods detect lane markings by observing a large change of feature values in a sub-region surrounding each pixel. Previous deep-learning based methods also analyzed the existence possibility of lane markings by considering the sub-region surrounding a pixel. These methods can be defined as a point-detection problem on local context. The results are lane segments; they should be clustered into groups that share similar properties such as position and direction, then classified as two ego lanes by additional heuristic steps. Methods based on deep learning solve the challenges of the lane detection more efficiently than model-driven methods. But, because methods based on deep learning train the CNN on an edge image, rather than on the original image [23], or consider the context only within a sub-image rather than the entire image [20], or estimate lane positions in a fully-connected layer using the information missed by numerous convolution and pooling layers [29, 17], they cannot fully exploit the information included in driving situation. We overcame these limitations by redefining the point-detection problem as a region-segmentation problem. This change of perspective, from point to region and from sub-context to full context, reduces the sensitivity of our approach to occlusion, degraded markings and various road conditions. For example, if a scene consists of roads and background, we can estimate the positions of ego lanes, even if they have poor texture. In our approach, the results of segmentation into left and right ego lane regions can be used to adjust the driving direction di-

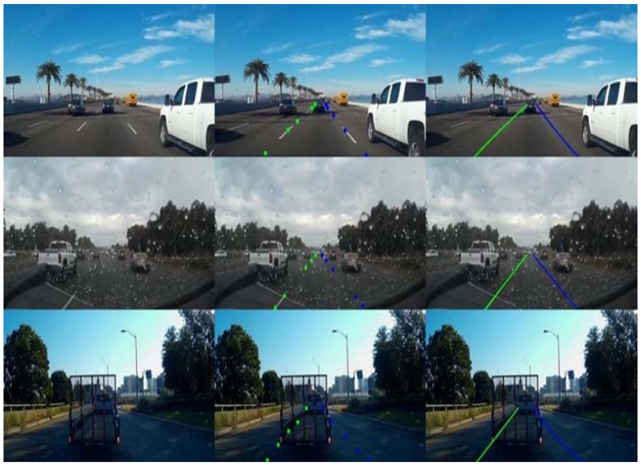

Figure 2. Semi-automatic interface to annotate left and right ego lanes. Original image (left). Annotation of lane points (center). Fitted curves (right).

rectly because they involve the information already. Also, we can use abundant scene segmentation datasets as pre-knowledge for lane region segmentation.

3.2. Dataset Generation of Ego Lanes

KITTI is very useful and famous dataset to evaluate a lot of functions for autonomous driving cars. Among various categories, lane dataset consists of 95 training and 96 test images. We wanted to construct and share a larger dataset that included more various highway and urban road conditions. So, we constructed a new dataset to segment ego lanes through end-to-end estimation. In advance, to reflect various road conditions, we downloaded black box videos of America, Europe and Asia from Youtube site. Their images includes various curved road as well as straight road. We also use Grand Theft Auto V (GTA5) and TORCS games to consider more various situations with less effort. The set of real data consists of 10,680 images (50 video clips, 5 hours 56 minutes, 5.43 GB). The set of virtual data consists of 960 images (2 videos, 32 minutes, 2.13 GB). Because high-quality game simulations describe real world concretely with a variety of scenarios that cannot be generated in real situations, recent research has used them very actively [10, 40].

To remove duplicated images and select representative scenes, we collected 4,000 images by sampling at an appropriate frame interval. The frame interval varies from 30 to 100 to obtain the same number of images from each video clip. We developed a semi-automatic annotation tool to change the collected images into training data, then used Matlabs interface to annotate left and right ego lanes for each image (Fig. 2). After marking the two end points of a left ego lane, we additionally marked three middle points in which the line direction changes between two end points. We selected the upper endpoint considering that all points will be fitted using a second-order polynomial curve. The right ego lane was also marked using the same process. To ensure the quality of the ground truth points, cross-checks were done through multiple people, and if the current frame had occlusion cases, the previous and next frames were referenced. Then two second-order polynomial curves were drawn using ten points, then the original image and four versions of it were saved for verification; these were: (1) a binary image with lane points, (2) a dilated binary image with lane regions, (3) a dilated color image with different colors on the left and right ego regions, and (4) a dilated gray image with different indexes on the left and right ego regions. The images used for training deep network were the original RGB image and image (4). We use the dilated lane region, not the line image, to represent the lane width as one context in our training process because a real lane is a region with a width value. We randomly selected 25% of images (1,000 images) as test data, and used the rest (3,000 images) as training data. To include a variety external environmental conditions, we performed extensive training data augmentation by scaling, blurring, translation, rotation, noise, and illumination. We used Matlab's various functions and parameters; imresize(0.8 1.4), imgaussfit(-1.0 2.5), pixel shift(-6 12), imrotate(-2 3), imnoise(gaussian, poisson, salt-pepper, speckle), and imadjust((-1.0,-1.0) (0.3 0.9)). We produced at most 30 versions of each image (i.e., 6 techniques by 5 parameters); the result was 90,000 image pairs for deep network training.

3.3. Learning Semantic Ego Lanes

Network Architecture In the region-segmentation problem, if the number of categories increases, the problem becomes one of scene segmentation. Recently, various networks have been proposed for pixel-level scene segmentation. FCN [36, 43] uses networks that consist of only convolution and pooling layers by eliminating the fully-connected layer from AlexNet [26], VGG-net [44], and GoogLeNet [45]. Two groups [37, 4] generated a segmented image with the same size of input image by adding upsampling networks. In this paper, we used SegNet [4]; it was mainly trained and evaluated with road scene data and has shown fast processing for real-time autonomous driving. The network consists of a convolution network (i.e., an encoder), which extracts features by hierarchical abstraction, and a deconvolution network (i.e., a decoder), which reconstructs a segmented image by upsampling. The convolution network has same structure as the first 13 convolution layers of the VGG15 network, and generates feature maps. To solve the gradient vanishing and exploding problem and to reduce the number of iterations taken for loss convergence in training process, it also includes batch normaliza-

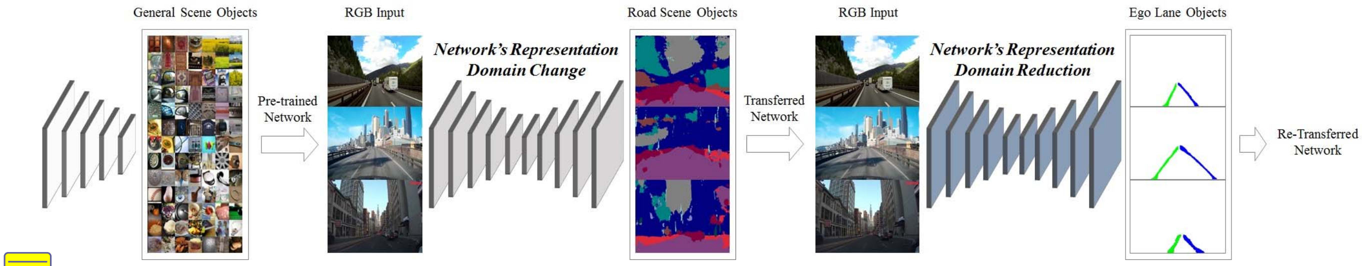

Figure 3. Overall procedure of the proposed sequential end-to-end transfer learning. VGG-net pre-trained on ImageNet dataset (first network) is symmetrically concatenated as a deconvolution network. The modified network is trained to segment each class component through representation domain change from general scene objects to road scene object (second network). Lastly, the transferred network is trained to extract left and right ego lanes by representation domain reduction from road scene objects to ego lane objects (last network).

tion [21] technique and uses a rectified linear unit (ReLU) as an activation function. The deconvolution network has deconvolution layers that correspond to the convolution layers and performs upsampling. The main idea of the method is to use max-pooling indices memorized in each pooling layer of convolution network. After passing the final deconvolution layer, they use trainable multi-class soft-max classifier to categorize each pixel.

Network Training We used two techniques for our end-to-end training and inference that consider the full content of a scene. The first technique was to acquire the contexts of various objects in a scene, then to transfer the learned representations of the deep network over two stages. Transfer learning [14] is a method that can use the representation ability obtained from huge amounts of another domain's data when our target domain does not have enough training data. For road scene segmentation, we performed a fine-tuning on published datasets, such as CamVid, Cityscapes, and GTA5. To train the network, we used stochastic gradient descent method with the same hyper-parameters with SegNet [4] including learning rate, momentum, and loss function. This process changes the feature representation domain of a network from general scene objects included in the ImageNet dataset, to road scene objects (Fig. 3, gray network). The second technique was to use the data obtained from our annotation tool and image amplification, to reduce again the domain from road scene objects to left or right ego lanes (Fig. 3, blue network). The input for this second transfer learning consists of 480x360 RGB images, and the output consists of ground truth images that are indexed into three categories: left ego lane region, right ego lane region, and background region, each labeled with a unique integer starting from zero. For both of two transfer learning, we applied a class balancing technique to assign the weight differently in the loss function according to the ratio of class frequency. During this second transfer learning using the same hyper-parameters with the first transfer learning, the time required to converge a cross-entropy loss is much shorter and the network's region segmentation ability increases because the number of target categories is much less than the number of categories of road scene.

To overcome challenges [50, 19, 27] such as occlusion, shadow, degradation, illumination, print quality, weather conditions, road geometries, and extraneous objects, the design of the ground truth data is also important. If lane markings are invisible for various reasons, human drivers recognize the whole context, estimate ego lanes intuitively, and drive on the correct path. Applying the same principle, we generated the ground truth data of ego lanes. If ego lanes were briefly invisible, we used our intuition obtained from the previous scenes to annotate the estimated region. In this way, by using data generated based on the logic that humans use while driving, our deep network can recognize ego lane under various road-surface conditions.

4. Experiments

We evaluated the accuracy of ego lane recognition. Each test image represents a unique road scene. The evaluation and analysis was performed from three distinct viewpoints: 1) network's representation ability dependent on different datasets used in the first transfer learning, 2) network's inference accuracy dependent on various data augmentation ratios used during the second transfer learning, and 3) network's identification stability in a variety of input variations.

Two measurements were used for these three experiments. One is composed of *region-based* precision and recall to evaluate the region segmentation accuracy of ego lanes. These measures are defined as

$$precision = \frac{TP}{TP+FP}, \quad recall = \frac{TP}{TP+FN}, \qquad (1)$$

where TP is the number of True Positives (i.e., the number of ego lane pixels correctly classified), FP is the number of False Positives (background pixels classified as ego lanes), and FN is the number of False Negatives (ego lane pixels classified as background). Because KITTI dataset and our

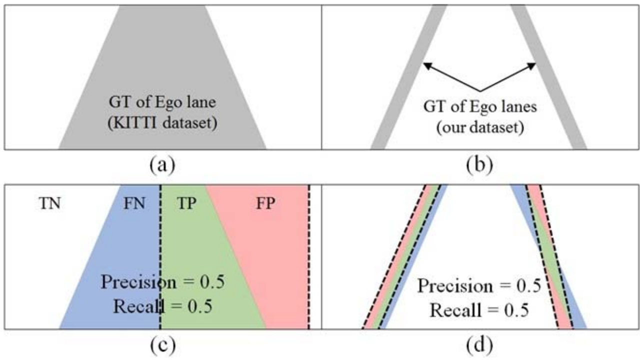

Figure 4. Definitions of ego lane of KITTI (a) and our dataset (b). Examples of TP, TN, FP, FN, region-based precision, and recall for KITTI (c) and our dataset (d). Dash lines mean the boundaries of estimated result.

paper use different meaning of 'ego lane', we used different metric. In KITTI dataset, ego lane means a wide region that is surrounded left and right ego lanes (Figure 4(a)). So, although the two side curves are not exact, precision and recall values are not low if the overall region is extracted well (Figure 4(c)). In our paper, ego lanes mean two narrow regions corresponding two side curves themselves (Figure 4(b)). The evaluation using these definition makes us to estimate more exact two side-curves (Figure 4(d)). Accurate estimation of left and right ego lanes (curves) is very important for lane departure warning, lane change assistance, forward collision avoidance (advanced driver assistance system), and self-driving (autonomous driving).

Another measure is *line-based* accuracy to evaluate the direction estimation of ego lanes (Figure 5). This statistic is defined as

$$accuracy = \frac{R_{GT} \cap R_{L,R}}{R_{L,R}}, \qquad (2)$$

where R_{GT} means the ground truth area of ego lane regions and $R_{L,R}$ means the area that was identified to be left or right ego lanes, and we ignored the lower 8% of an input image because ego lanes are occasionally occluded by the bonnet of a car. We evaluated line-based accuracy because, unlike the proposed method that extracts left and right ego lanes directly and separately, the existing methods separate ego lanes by applying postprocessing after detecting all lanes. Thus, the method cannot define FP and FN for their naive outputs before postprocessing. This second metric is very reasonable because real lane has a little width. That is, after we fit the extracted two ego lanes into 2nd polynomial curves, we computed the overlapped ratio between the fitted line and ground-truth region. Then, if the fitted curve is included within the width, the result is decided as reasonably good.

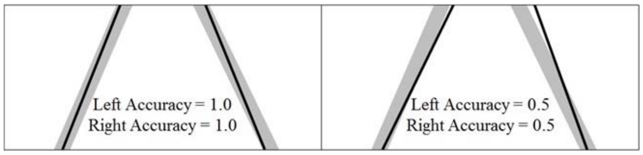

Figure 5. Examples of the line-based accuracy.

Dataset	Images	Resolution	Classes
CamVid	701	960x720	32 (11)
Cityscapes (fine)	5,000	2,048x1,024	30
GTA5	24,966	1,914x1,052	19

Table 1. The overview of published datasets for semantic scene segmentation.

4.1. Importance of Sequential Domain Change

Many published datasets (Table 1) are suitable for the first transfer learning. We used them to analyze the effect of different datasets on network's representation ability. For CamVid dataset, in common with [4], we used 11 classes for the first transfer learning and it is reasonable because it has much less number of images among datasets. Cityscapes has highest resolution and includes very various kinds of cities with the most classes. GTA5 has much more images than other datasets with middle resolution and classes. We performed transfer learning to change the feature representation domain from general scene objects to road scene objects, and measured the scene segmentation accuracy. We did not use all datasets together, because they have different numbers of categories, so we applied median frequency balancing [15] for each dataset. The method achieved the highest scene segmentation accuracy in the fewest iterations on the CamVid dataset (Fig. 6, left), because this dataset has fewer categories and less number of images than the other datasets. Cityscape 5,000 and GTA5 24,966 need more iterations to obtain enough accuracy. But this is just the scene segmentation result on different datasets, not the region segmentation result of ego lanes, so we performed the second transfer learning using the same ego lane dataset that we collected (Fig. 6, right). After 20,000 iterations, there is no difference between ego lane segmentation accuracy of five datasets; this result means that the accuracy of the first transfer learning is not directly related to one of the second transfer learning in terms of the accuracy. The important thing is that increasing the number of categories considered in the first transfer learning improves the network's representation ability at the second transfer learning with much less iterations. If we perform the second transfer learning without the first transfer learning, we need much more iterations to achieve the comparable high accuracy(Fig. 6, right (None)).

Also, we measured the precision, recall, and F measure

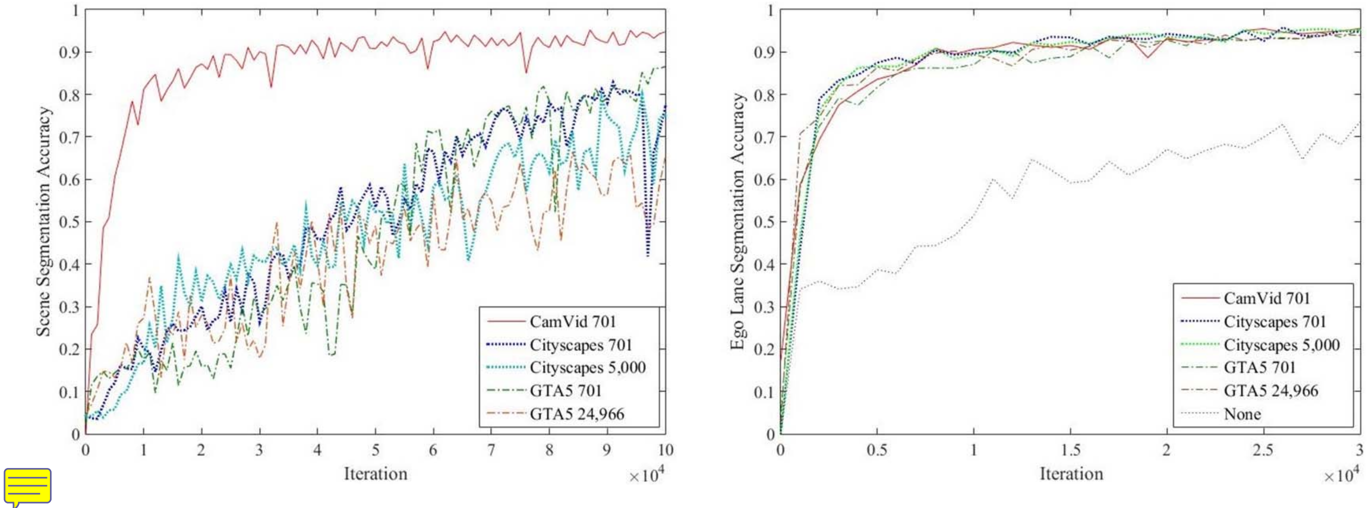

Figure 6. (Left) Scene segmentation accuracy during 100,000 iterations at the first transfer learning. The numbers, 701, 5,000, and 24,966 mean the number of images that is used in training. (Right) Lane segmentation accuracy during 30,000 iterations at the second transfer learning. The accuracy converges with much less iterations compared with the first transfer learning.

Dataset	Precision	Recall	$F_{0.5}$ measure
CamVid 701	0.37	0.77	0.41
Cityscapes 701	0.39	0.58	0.42
Cityscapes 5,000	0.43	0.54	0.45
GTA5 701	0.33	0.78	0.37
GTA5 24,966	0.31	0.69	0.35

Table 2. Precision, recall, and $F_{0.5}$ measure of averaged total ego lanes. Because precision is more important than recall in driving lane estimation (false detection is much more dangerous than missed detection), we used $\beta = 0.5$.

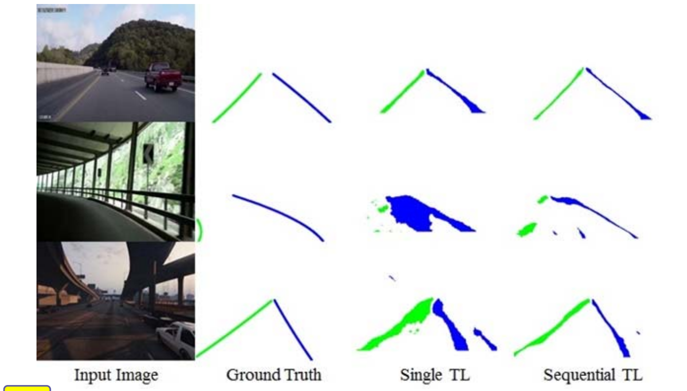

Figure 7. Comparison of ego lane estimation without(single TL)/with(sequential TL) the first transfer learning. Especially, in shadow, tunnel, and overpass, the network with the first transfer learning recognizes well the left and right ego lanes.

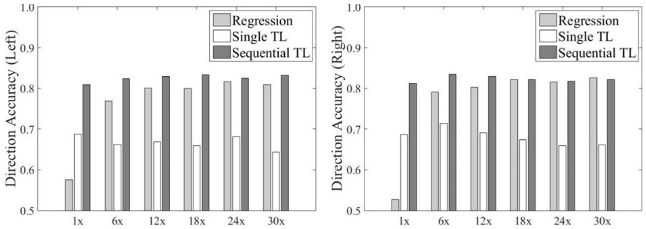

Figure 8. Comparison of direction estimation accuracy with augmented training data from regression, single TL network (without the first transfer learning), and our sequential TL network. (Left) left ego lane. (Right) right ego lane. 1x, 6x, 12x, 18x, 24x, and 30x, represent that set Nx consists of N x 10^3 images.

of ego lane regions using test data (Table 2). The deep network trained using Cityscapes 5,000 (full) dataset showed the most reasonable results when we consider the trade-off between precision and recall values. Because the number of our ground-truth pixels is much less than KITTI dataset, there are big gaps of precision and recall values. In our definition of ego lanes, if the direction of detected two ego lanes is a little different with the ground-truth, then precision and recall decrease rapidly. For various road conditions, sequential transfer learning (TL) on Cityscapes 5,000 shows accurate and stable estimation results (Fig. 7), so we selected this network for following experiments.

4.2. Extensive but Reasonable Data Augmentation

We measured the direction estimation accuracy to evaluate the amplification effect of our training data on network's inference performance. From Sec 4.1, we selected the network that is trained using Cityscapes 5,000 as the base model and performed the second transfer learning by fine-tuning for different data augmentation ratios. We compared our proposed method with a recent regression method based on deep learning [20]; this method predicts two end

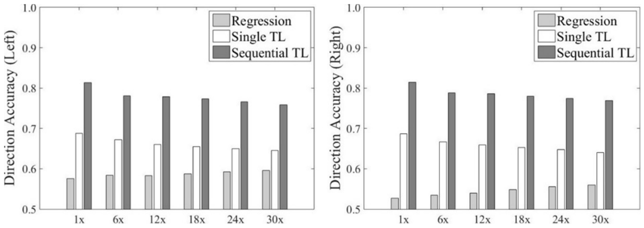

Figure 9. Comparison of direction estimation accuracy with augmented test data from regression and our network. Left ego lane (left). Right ego lane (right).

points of lane segments in each sliding window and clusters them using DBSCAN. Because we compare the ego lane direction, we selected among the clustered lanes the two that are closest to the center of an input image. The reason that we select the method is it generates lane curves not a wide region that is surrounded by curves. At the 1x augmentation ratio (no augmentation) or small amount of training data, the proposed method obtained much higher accuracy (Fig. 8) because it considers all information in a scene, rather just parts of it. Also, the proposed method achieved higher direction estimation accuracy for most cases. Regression method was more closely related to training data augmentation than our method, because the local information of a sub-region is much more affected by the augmentation techniques than the full context of an image. In either method, direction estimation did not provide better than using the 18x augmentation ratio because unnecessary images were added to training data in 24x and 30x. This result implies the existence of reasonable augmentation ratio with only the data that affect the accuracy; more training data augmentation than the reasonable criterion do not help improve the accuracy.

4.3. Robustness For Road Variations

In the previous Section, we evaluated the augmentation effect of training dataset on the fixed test dataset. Lastly, we evaluated the robustness of the ego lane recognition on various input road conditions on the fixed training dataset (no augmentation). From Section 4.2, we selected three models (1x) for regression, single TL, and sequential TL methods. We amplified the test data instead of the training data by applying image-processing techniques including scaling, blurring, translation, rotation, noise, and illumination (Sec. 3.3). The input variations represent diverse outputs from a camera mounted in a vehicle, such as low-resolution images, installation-position change, installation-angle change, noise caused by a faulty camera, and low illumination during the evening. Using five parameters of each amplification technique, we set six sets of test images: 1x, 6x, 12x, 18x, 24x, and 30x, where set Nx consists of N x 10^3 images. For each amplification ratio, we compared the direction estimation accuracy (Fig. 9). The accuracy of three methods was little affected ($< 4\%$) by the input variations. Our proposed sequential TL method achieved higher direction estimation accuracy (left ego lane, right ego lane, full ego lanes) than the regression model and single TL method in all sets regardless of input variations, even when the camera state or the external environmental condition changed severely. Especially, our proposed method represented better estimation results under rough road conditions and low illumination environments (Fig. 10), where the straight road is often seen because the proportion of the highway images is higher than the urban images. These results indicate that ego lane segmentation stabilizes when all information in the scene is considered.

5. Conclusion

We proposed a method of semantic ego lane estimation to train a deep network based on sequential end-to-end training and to recognize left and right ego lanes without postprocessing directly and separately. The method uses two transfer learning steps. The first step changes the network's representation domain from a general scene to a road scene; the second step reduces the target from road objects in general, to left and right ego lanes in particular. Because this sequential domain change is based on region segmentation that considers the full context of a scene, the proposed method recognizes ego lanes with low sensitivity to road conditions. Also, the end-to-end approach reduces re-design and re-optimization during data modification and eliminates the possibility that postprocessing generates errors. By modification and extension of our target data, the proposed approach can support more detailed information for to support driving situation decisions or to establish driving strategies.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv:1609.08675*, 2016.

[2] M. Aly. Real time detection of lane markers in urban streets. *In IVS*, 2008.

[3] A. Assidiq, O. Khalifa, R. Islam, and S. Khan. Real time lane detection for autonomous vehicles. *In ICCCE*, 2008.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation.

[5] A. Borkar, M. Hayes, M. Smith, and S. Pankanti. A layered approach to robust lane detection at night. *In CIVVS*, 2009.

[6] A. Borkar, M. Hayes, and M. T. Smith. Robust lane detection and tracking with ransac and kalman filter. *In ICIP*, 2009.

[7] A. Borkar, M. Hayes, and M. T. Smith. Polar randomized hough transform for lane detection using loose constraints of parallel lines. *In ICASSP*, 2011.

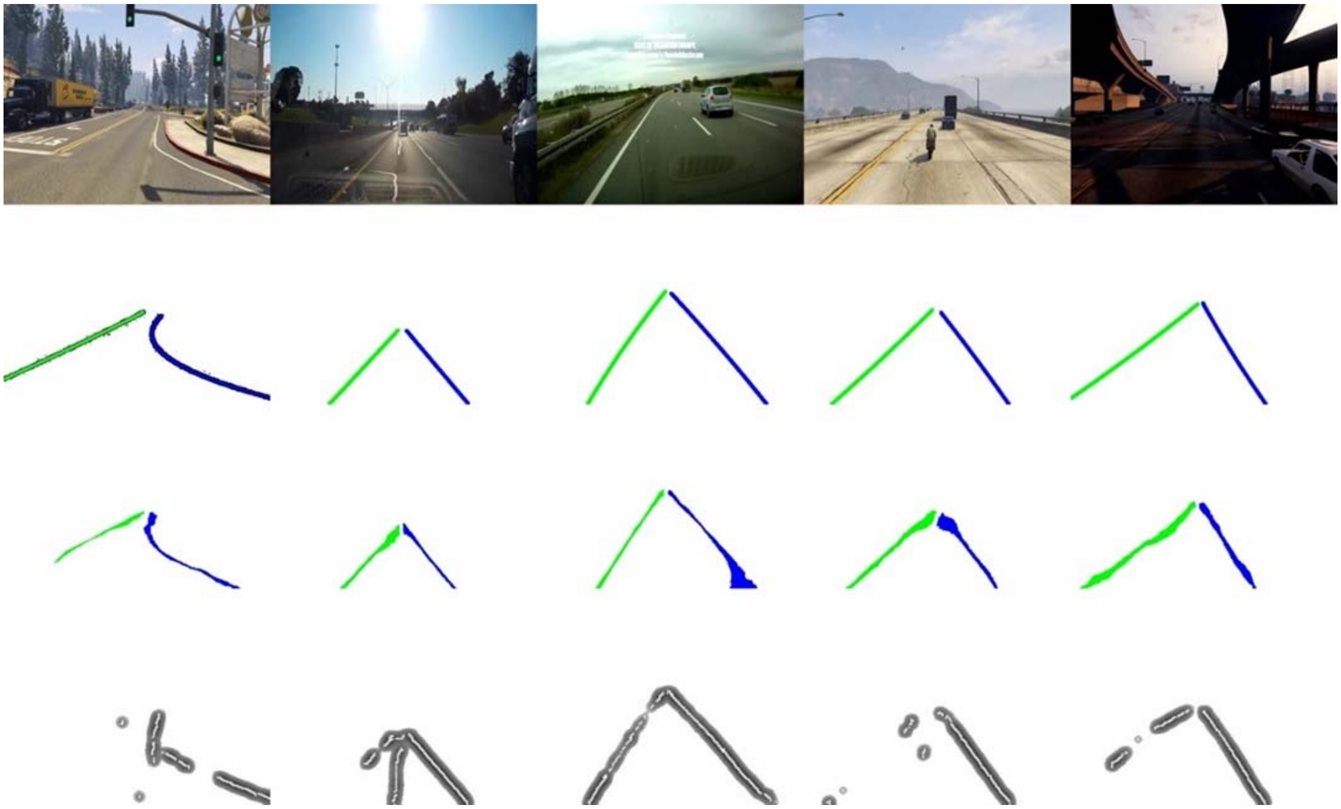

Figure 10. Example of semantic ego lane estimation results with augmented test data; these were: (first row) input image, (second row) ground-truth ego lanes, (third row) estimated left/right ego lanes by our method without post-processing, and (fourth row) estimated lane points by regression method, where shaded regions are just for separation between lane points and background. To estimate left and right ego lanes as a second-order polynomial curve, the regression method needs additional postprocessing such as line clustering and ego lane separation based on geometric assumptions.

[8] 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.

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

[10] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: learning affordance for direct perception in autonomous driving. *In ICCV*, 2015.

[11] H. Y. Cheng, C. C. Yu, C. C. Tseng, K. C. Fan, J. N. Hwang, and B. S. Jeng. Hierarchical lane detection for different types of roads. *In ICASSP*, 2008.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *In CVPR*, 2016.

[13] P. Daigavane and P. Bajaj. Road lane detection with improved canny edges using ant colony optimization. *In ICETET*, 2010.

[14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: a deep convolutional activation feature for generic visual recognition. *In ICML*, 2014.

[15] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *In ICCV*, 2015.

[16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *In CVPR*, 2012.

[17] 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. *In CVPRW*, 2016.

[18] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *In CVPR*, 2015.

[19] A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, 25:727–745, 2014.

[20] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. C. Yue, F. Mujica, A. Coates, and A. Y. Ng. An empirical evaluation of deep learning on highway driving. *arXiv:1504.01716*, 2015.

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

[22] H. Jung, J. Min, and J. Kim. An efficient lane detection algorithm for lane departure detection. *In IV*, 2013.

[23] J. Kim and M. Lee. Robust lane detection based on convolutional neural network and random sample consensus. *In ICONIP*, 2014.

[24] Z. Kim. Robust lane detection and tracking in challenge scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):16–26, 2008.

[25] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, A. Gupta, D. Narayanan, C. Sun, G. Chechik, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *https://github.com/openimages*, 2016.

[26] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *In NIPS*, 2012.

[27] A. M. Kumar and P. Simon. Review of lane detection and tracking algorithms in advanced driver assistance system. *International Journal of Computer Science Information Technology*, 7(4):65–78, 2015.

[28] Y. C. Leng and C. L. Chen. Vision-based lane departure detection system in urban traffic scenes. *In ICARCV*, 2010.

[29] J. Li and X. Mei. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[30] C. W. Lin, H. W. Yang, and D. C. Tseng. A robust lane detection and verification method for intelligent vehicles. *In IITA*, 2009.

[31] Q. Lin, Y. Han, and H. Hahn. Real-time lane detection based on extended edge-linking algorithm. *In ICCRD*, 2010.

[32] T. Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar. Microsoft coco: Common objects in context. *In ECCVW*, 2016.

[33] G. Liu, F. Worgotter, and I. Markeli'c. Combining statistical hough transform and particle filter for robust lane detection and tracking. *In IV*, 2010.

[34] G. Liu, F. Worgotter, and I. Markeli'c. Lane shape estimation using a partitioned particle filter for autonomous driving. *In ICRA*, 2011.

[35] G. Liu, F. Worgotter, and I. Markeli'c. Stochastic lane shape estimation using local image descriptors. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):13–21, 2013.

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *In CVPR*, 2015.

[37] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *In ICCV*, 2015.

[38] G. L. Oliveira, W. Burgard, and T. Brox. Efficient deep models for monocular road segmentation. *In IROS*, 2016.

[39] E. Real, J. Shlens, S. Mazzocchi, and X. Pan. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *arXiv:1702.00824*, 2017.

[40] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: ground truth from computer games. *In ECCV*, 2016.

[41] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. *In CVPR*, 2016.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[43] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *In CVPR*, 2015.

[46] M. Tan, B. Paula, and C. R. Jung. Real-time detection and classification of road lane markings. *In SIBGRAPI*, 2013.

[47] T. T. Tran, H. M. Cho, and S. B. Cho. A robust method for detecting lane boundary in challenging scenes. *Information Technology Journal*, 10(12):2300–2307, 2011.

[48] S. C. Tsai, B. Y. Huang, Y. H. Lin, C. W. Lin, C. S. Tseng, and J. H. Wang. Novel boundary determination algorithm for lane detection. *In ICCVE*, 2013.

[49] J. Wang, T. Mei, B. Kong, and H. Wei. An approach of lane detection based on inverse perspective mapping. *In ITSC*, 2014.

[50] S. Yenikaya, G. Yenikaya, and E. Duven. Keeping the vehicle on the road - a survey on on-road lane detection systems. *ACM Computing Survey*, 46(1), 2013.

[51] Y. U. Yim and S. Y. Oh. Three-feature based automatic lane detection algorithm (tflda) for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 4(4):219–225, 2003.

[52] H. Zhao, Z. Teng, H. H. Kim, and D. J. Kang. Annealed particle filter algorithm used for lane detection and tracking. *Journal of Automation and Control Engineering*, 1(1), 2013.

[53] 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 IV*, 2010.