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

Road Lane Semantic Segmentation for High Definition Map

Wonje Jang, Jhonghyun An, Sangyun Lee, Minho Cho, Myungki Sun and Euntai Kim*

Abstract— **High Definition map (HD Map) is an important part of autonomous driving vehicle. Most conventional method to generate HD map requires expensive system and post-processing of observed data. In this paper, we propose automatic HD map generating algorithm using just monocular camera without further human labors. The proposed algorithm detects road lane from image and classifies the type of road lane at pixel-level with Fully Convolutional Network (FCN) which outperforms the other semantic segmentation methods. The segmentation results are used to extract lane features, and the features are used for loop-closure detection. Final map is generated with graph-based Simultaneous Localization and Mapping (SLAM) algorithm. The experiment is done with monocular camera mounted on mobile vehicle. In this paper, final map generated by proposed method is compared with aerial view data. The results show that the proposed method can generate reliable map that is comparable to real roads even only the low-cost sensor is used.**

I. Introduction

The HD map generation is an essential technique for autonomous driving. This technique can be used to cover three challenges for autonomous driving. First, it helps a vehicle to localize itself with high precision in a map. Second, it can be used to solve the problem of recognition or prediction to events occurring beyond the reach of onboard sensors. Third challenge concerns the vehicle's capability to perform path planning in accordance with the needs of passengers. Until now, maps in vehicles have been used mainly navigation purposes. However, the resolution of these maps is not precise enough for autonomous driving.

Data for HD maps are commonly obtained using the Mobile Mapping System (MMS), typically fitted with a range of photographic, radar, laser, LiDAR or any number of remote sensing systems. GPS combined with camera systems allow acquire precise data for HD map. The data include road signs, traffic lights, lanes, road marks and area surrounding the road while driving. However, the mapping system squanders so much resource because the sensors mounted in MMS cost a lot and the data from MMS are analyzed manually. There is no research have been done to generate map with low-cost sensor.

W. Jang, J. An, S. Lee, M. Cho and E. Kim* is in the School of Electrical and Electronic Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, Korea (email: [jangwj1256; jhonghyen; hello072; minho8849; etkim] @yonsei.ac.kr)

M. Sun is in Hyundai Mnsoft, Hyundai Motor bldg., 74, Wonhyo-ro, Yongsan-gu, Seoul 04365, Korea (email: mk77@hyundai-mnsoft.com)

This research was supported by 'Machine Learning-based Development of Lane Extraction Algorithm from Mobile Mapping System(MMS)' project funded by the HYUNDAI-MnSOFT Corporation.

*Corresponding author

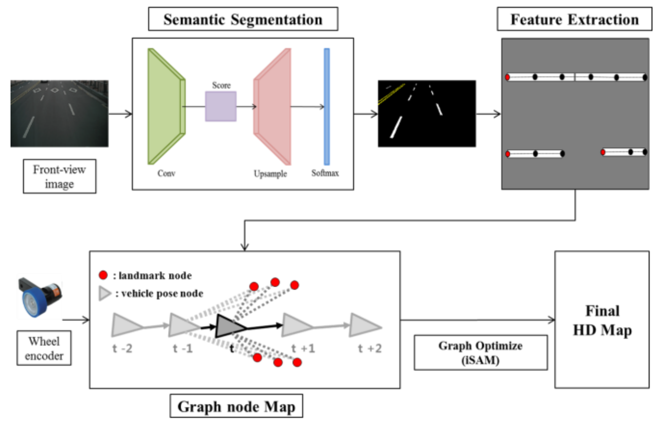


Figure 1. System architecture of HD Map reconstruction system.

In this paper, we proposed an entire system that automatically creates HD maps from data obtained using a monocular camera. Based on the odometry data obtained from the wheel encoder of the vehicle, we estimate the position of the car and detect road lane from the obtained image. With recent advances in image processing, there were many studies about lane detection with image processing[1], [2] or deep learning[3], [4], [5]. Most researches are detecting offset of lane for vehicle localization or drivable region for advanced driver assistance systems(ADAS)[3]. There are little researches to use lane detection result as landmark for map[1]. However, these researches used only image processing to detect road lane from image. In this paper, the location and type of the lane are detected through image semantic segmentation using deep learning. The datasets obtained for training were manually labeled at the pixel level, and we used semantic segmentation networks to produce impressive performance for mapping. The details about the acquisition of dataset and segmentation are given in sections II - III. Using the road lane data extracted from the image, we update a graph map by comparing with the map data created in the previous step. The graph map is used to restore the correct high definition map and vehicle trajectory by optimizing the error through graph optimization. Details of the entire system of HD map generation are described in detail from section III - V.

In this paper, we make three contributions:

- We design a mapping system using low-cost sensor instead of using expensive sensors such as DGPS in MMS. The system presented in this paper produces maps in a cheaper and faster way than current manual system without sacrificing required performance and quality.

- Proposed method is to detect the exact position, size, and type of road lane by applying semantic segmentation method. Unlike the conventional method that estimates the offsets of road lane on an image through image processing and linear regression, the proposed method offers the type of lane and exact pixel coordinate of lane.

- We collect our own dataset by pixel-wise annotating lanes for training and test. We created our own annotation tool for dataset which is simple and fast for road lane annotation, error correction and class complements.

II. SYSTEM OVERVIEW

In this paper, we propose an algorithm to generate HD map with only image data. The overall system is shown in Fig. 1. The entire algorithm consists of three steps, the road lane segmentation step, lane feature extraction step and graph based SLAM step.

At the beginning of the algorithm, the proposed method performs image semantic segmentation. For training and validating road lane segmentation, benchmark dataset need to be built up. We collected our own dataset with own annotation tool. Three semantic segmentation networks which are states of the art are applied and the outperforming network is adopted for the proposed algorithm. The Fully Convolutional Network is used for road lane segmentation. After segmentation step, it is necessary to extract the feature points of each road lane to reduce the memory of map. In the feature extraction step, the global coordinate of road lanes is restored from camera coordinate. The details of segmentation step and feature extraction step are described in Section III.

Local SLAM and global SLAM are executed with the extracted features and vehicle position. Error of local lane map is calculated with comparing the current lane map with lane map generated in previous step and the map is updated to optimize the local error. Even though errors are optimized through local SLAM, errors are still accumulated at several situations, such as intersections where there are no lanes to compare. To minimize the global error, stop line is used as landmarks for vehicle trajectory correction. The vehicle poses and relative position of landmark are fed into the pose graph for optimal trajectory and landmark pose. Details are given in Section IV.

III. SEMANTIC SEGMENTATION FOR LANE EXTRACTION

A. Dataset Collection

In order to perform image segmentation for various type of lane, we need dataset where each pixel is labeled with types of lanes or non-lane. To this end, we collected data in various circumstances and carried out pixel-wise annotation with our own tool. Examples of our dataset and other segmentation datasets are shown in Fig. 2. Fig. 2 (a), (b) are examples of the Cityscapes dataset[6] and the KITTI dataset[7], respectively. Although these datasets are for driving scenes, all road regions are annotated as a single road segment, and it is not suitable for our purpose. On the other hand, our dataset focuses on pixel-wise semantic segmentation of all type of lanes as shown in Fig. 2 (c)-(d).

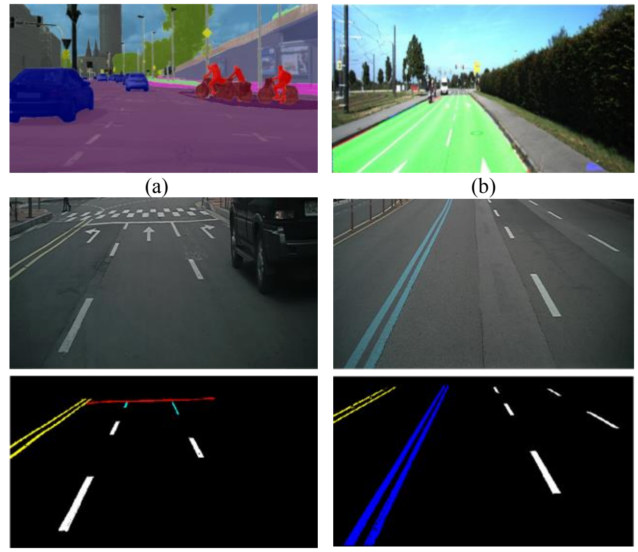

(a) (b)
(c) (d)

Figure 2. Example of existing image segmentation dataset. (a) Cityscapes dataset[6]. (b) KITTI dataset[7]. (c), (d) are examples of annotation image of our dataset.

In the existing annotation tools for segmentation dataset, labelers should annotate all corner points of each object to form a polygon including the object. However, this approach cannot handle curved lanes efficiently since they have a lot of corner points and it require careful efforts to cover this type of lane precisely. To overcome this problem, we created our own tool where labeler can annotate each lane by picking only 2 points around the lane.

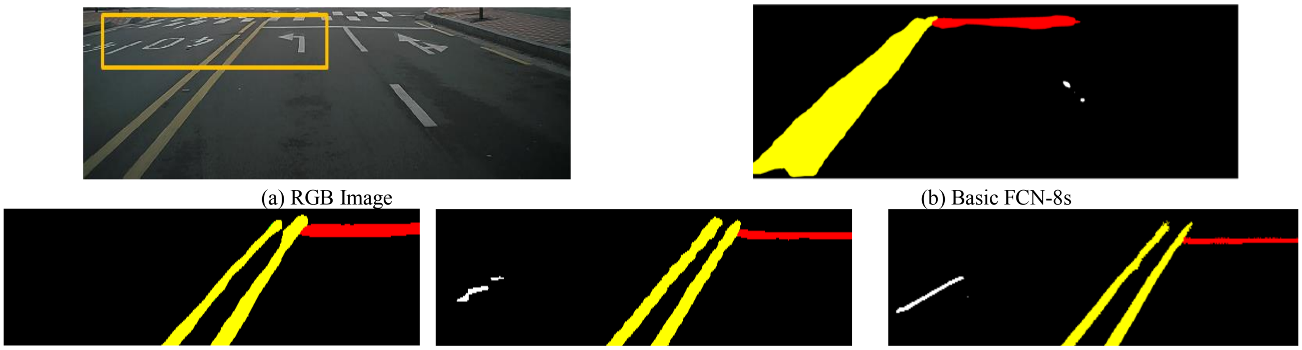
Details on our annotation tool that were made to target the dataset's focus are following. First, frontal view image from monocular camera is transformed into bird's eye view form with inverse perspective mapping (IPM)[8]. The reason for doing this is that it addresses the difficulty of annotating lanes near vanishing point. Second, binary image is generated from the bird's eye view using Top-Hat filter. Motivated by the fact that lanes are brighter than the dark asphalt, pixels that are brighter than threshold in gray scale set to 1 and other pixels set to 0. Then when a labeler picks 2 points, one for the top-left corner and one for bottom-right corner of bounding box of each lane, all pixels of having value 1 in the bounding box are annotated.

In this paper, the total number of image obtained for the experiment is 13,493. The obtained images include situations such as a bright day, a cloudy day, a road with many vehicles. There are seven classes of lane in collected dataset: single-yellow lane, single-white lane, double-yellow lane, double-blue lane, dash white lane, lane of intersection and stop line. Among the annotated data, 9452 images are used as training data and 4041 images are used as test data and validation data for performance check.

B. Semantic Segmentation

a. Basic Network

In this paper, we use image semantic segmentation to detect road lane. We used three semantic segmentation methods to apply semantic segmentation to lane detection: Segmentation Network (Seg-Net)[9], FCN[10] and Pyramid Scene Parsing Network (PSP-Net)[11].



(a) RGB Image (b) Basic FCN-8s

(c) SegNet (d) PSPNet (e) FCN-8s

Figure 3. Semantic segmentation example for each network. (a) is test image and (b) shows result of basic network result without weighted loss function. (c) ~ (e) show results of weighted networks. The result is an enlargement of the orange box area of (a).

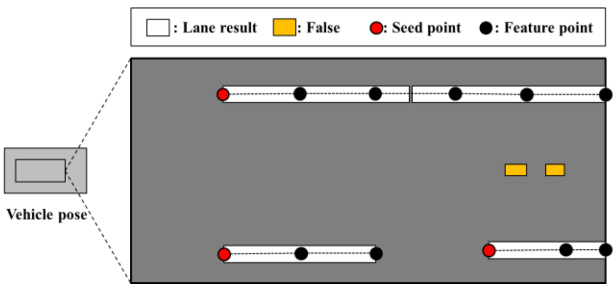


Figure 4. Feature extraction from top view image of semantic segmentation result. This process extracts key points from each road line segment.

The previous image segmentation method is to detect pixel-wise objects in image, such as buildings, road area, vehicles, pedestrians, trees and so on. Road lane, which is the detection target of this paper, occupies only a small area of the whole image, so the class imbalance problem occurs. Pixels not belonging to any road lane classes are classified as background class, and the average number of pixels of background class per image exist about 100 times more than road lane pixels. In order to reduce bias, we set region of interest (ROI) except for the unnecessary area and use them as training set and test set. However, the performance of network is inaccurate, as shown in Fig. 3 (b), because the ratio of classes is still too biased even when we set ROIs.

b. Weighted Network

Our loss function which is designed to reduce the performance degradation due to the class imbalance problem is presented. To compute the total loss values, we use a weighted sum of losses for each class instead of just averaging these values. The class-dependent weights are determined by an update equation as follows:

$$\lambda_{c,t+1} = \lambda_{c,t} + \left\{ \sum_{d} \frac{\lambda_{ct}}{\lambda_{d,t}} \frac{n_{c,d}}{n_{c,c}} - \sum_{j} \frac{\lambda_{d,t}}{\lambda_{c,t}} \frac{n_{d,c}}{n_{d,d}} \right\} \quad (1)$$

where $c \in [0, 7]$ is the class index; $n_{c,d}$ denotes the number of pixels of the actual class c but misclassified as class d. $\lambda_{c,t}$ is weight of class c at t epoch. The initial value $\lambda_{i,0}$ is determined by inverse of the ratio of the number of pixels in each class to the background pixel while the initial weight of the background class ($i = 0$) is just fixed as 1.

Fig. 3 shows the results of the proposed weighted network method applied to FCN, PSP-Net, Seg-Net. It can be seen that the weighted networks (Fig.3 (c)-(e)) shows the better performance than the result of basic network without updating weights (Fig (b)). Among the networks, FCN-8s is used for the mapping system since the network with the most accurate performance is FCN-8s. The detail about experiment setup is in experiment section, Section V.

C. Road Lane Feature Extraction

Considering the limitation of computational cost, we extract the feature points from each lane segment instead of using all pixels from network. To this end, the lane segment pixels are transformed from the camera coordinate to the global coordinate using inverse perspective mapping(IPM). The lanes in bird's eye view image have characteristic that road lanes are parallel to the direction of vehicle and have rectangular shape. The process of extracting lane feature points is shown in Fig. 4.

The nearest pixel from vehicle is set as a seed point of lane (red points in Fig. 4). The points which are connected to seed point are selected as points from same lane. However, in case of a real road situation, many lanes are erased or interrupted, so it is difficult to judge whether the lane is connected or not by 4-connectivity or 8-connectivity method. Since the direction which the column increases in the image coordinate is the direction of the lane, 4-connectivity is determined by comparing the reference pixel (x_i, y_j) with the pixel $(x_i, y_{j+1,2,3,4})$, not the conventional 4-connectivity method that compares reference pixel (x_i, y_j) with the pixels $(x_{i\pm1}, y_{j\pm1})$. Therefore, the proposed method can compute connectivity of lane even if the lane is interrupted or erased. The feature points of lane (black points in Fig. 4) are extracted at a certain distance from seed point. The distance is set as 40 pixels which is about 0.4m. The final feature point is extracted from end of lane.

IV. HIGH DEFINITION MAP RECONSTRUCTION

A. Local Map Matching

Global coordinate of road lane is calculated with vehicle pose coordinate at the time. The vehicle pose is obtained with odometry data from wheel encoder sensor and it contains

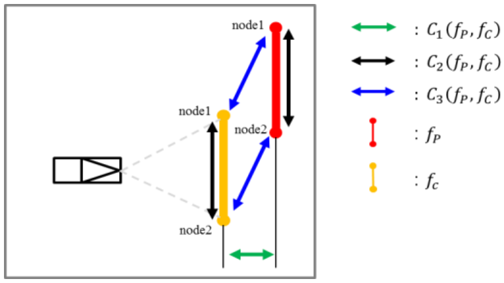


Figure 5. Pose estimation with stop line in loop-closure candidate pair

errors. Therefore, if the global coordinate of lane is determined with only vehicle position, the error is accumulated and accuracy of final map is deteriorated. Iterative Closest Point (ICP) based map matching[12] is used while updating each local map. Extracted lane features are considered as the point of the current map and compared to the point from previous map to obtain the relative pose of the current vehicle.

B. Loop-closure Matching

The problem with using ICP alone to correct errors is that they cannot handle the situation such as an intersection, where there is no lane feature. To overcome the limit of ICP, Loop-closure detection and graph optimization is used with ICP and we use stop line as a landmark for graph optimization. The reason why we choose stop line as landmark is that it is lying vertical to heading angle of vehicle and has own length, so it can effectively compensate the longitudinal and lateral error of the vehicle position. When a stop line is detected, the algorithm determines whether the stop line is re-visited or not. If there is no previous data or similar candidate, the stop line is updated as a new landmark. If the previous stop line data exists, the similarity between the previous stop line which is loop-closure candidates and the currently detected stop line is calculated. The similarity between stop lines is calculated by the following equation:

$$\underset{p}{\arg\min} \sum_{i=1}^{3} C_i(feature_P, feature_C) \quad \begin{cases} C_1(f_P, f_C) = \left|\overrightarrow{dist_{P,C}}\right| \\ C_2(f_P, f_C) = \left|f_{P,\text{node1}} - f_{P,\text{node2}}\right| - \left|f_{P,\text{node1}} - f_{P,\text{node2}}\right| \\ C_3(f_P, f_C) = \left|f_{P,\text{node1}} - f_{C,\text{node1}}\right| - \left|f_{P,\text{node2}} - f_{C,\text{node2}}\right| \end{cases} \tag{2}$$

, where $C_1(f_P, f_C)$ is function that returns distance between stop line segment. The following Eq. (3) shows the distance function.

$$\overrightarrow{dist_{P,C}} = \left(f_{P,\text{node1}} - f_{C,\text{node1}}\right) + s_c\vec{u} - t_c\vec{v} \tag{3}$$

, where $f_C = f_{C,\text{node1}} + t * \vec{v}$ when f_P, f_C is linear equation of previous and current stop line. $C_2(f_P, f_C)$ is the equation for calculating the similarity of length between the stop lines, and $C_3(f_P, f_C)$ is the equation for calculating the direction similarity between the stop lines. If the result of Eq. (2) is smaller than a certain threshold, it is determined as a re-visited landmark and the pair is grouped into a loop-closure pair. See Fig. 5 for a description of the above equation.

Figure 6. Monocular camera equipped on experimental vehicle.

C. Graph SLAM

After all loop closure pairs are created as above, the relative position of the vehicle can be calculated by using the distance difference of the pair. When the relative position is calculated, the result is optimized using the Incremental Smoothing and Mapping (iSAM)[13] to minimize the following error function:

$$X^*, L^* = \underset{X,L}{\arg\min}\left\{\sum_t \left\|f\left(x_{t-1}, u_t\right) - x_t\right\|_{\Sigma_t}^2 + \sum_k \left\|h_k\left(x_{tk}, l_{jk}\right) - z_k\right\|_{\Sigma_k}^2\right\} \tag{4}$$

, where X, L is trajectory of vehicle and landmark of map. $f\left(x_{t-1}, u_t\right)$ is process model with odometry measurement u_t. Also $h_k\left(x_{tk}, l_{jk}\right)$ is measurement model which l_{jk} is landmark measured at current pose of vehicle. x_t and z_k is current pose and measurement from the camera. Above equation optimizes the distance error between the pairs in loop-closure candidates and we can obtain the corrected vehicle trajectory and landmark as the result of graph optimization. The result of the corrected map with real road data is shown in experiment section.

V. EXPERIMENT

A. Experiment Setup

The experiments are done with real, on-road data with mobile system shown in Fig. 6 with monocular camera installed behind of windshield of vehicle and wheel encoder installed to obtain odometry of vehicle. Data for training and mapping are acquired with this vehicle. The network experiment was conducted using a simulator equipped with Intel Core i7-7700, and GeForce GTX 1080 Ti, and all the processes goes offline.

B. Semantic Segmentation Performance

A total of 13,493 pieces of image were obtained on various on road situations using our experimental vehicle. We set ROI in road region, so we cropped bottom part of image. The original image size is 648×1280 and cropped size for network is 320×1280. We evaluate the performance of four networks which are SegNet, PSPNet, FCN and the weighted FCN used in this paper. In order to evaluate the performance, both the precision and recall should be considered. Therefore, the F-score, calculated by Eq. (5) including both recall performance and precision performance, is used as a performance index. The β value is weight, higher value for higher recall weight than precision and lower for opposite situation.

$$F = \frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall} \tag{5}$$

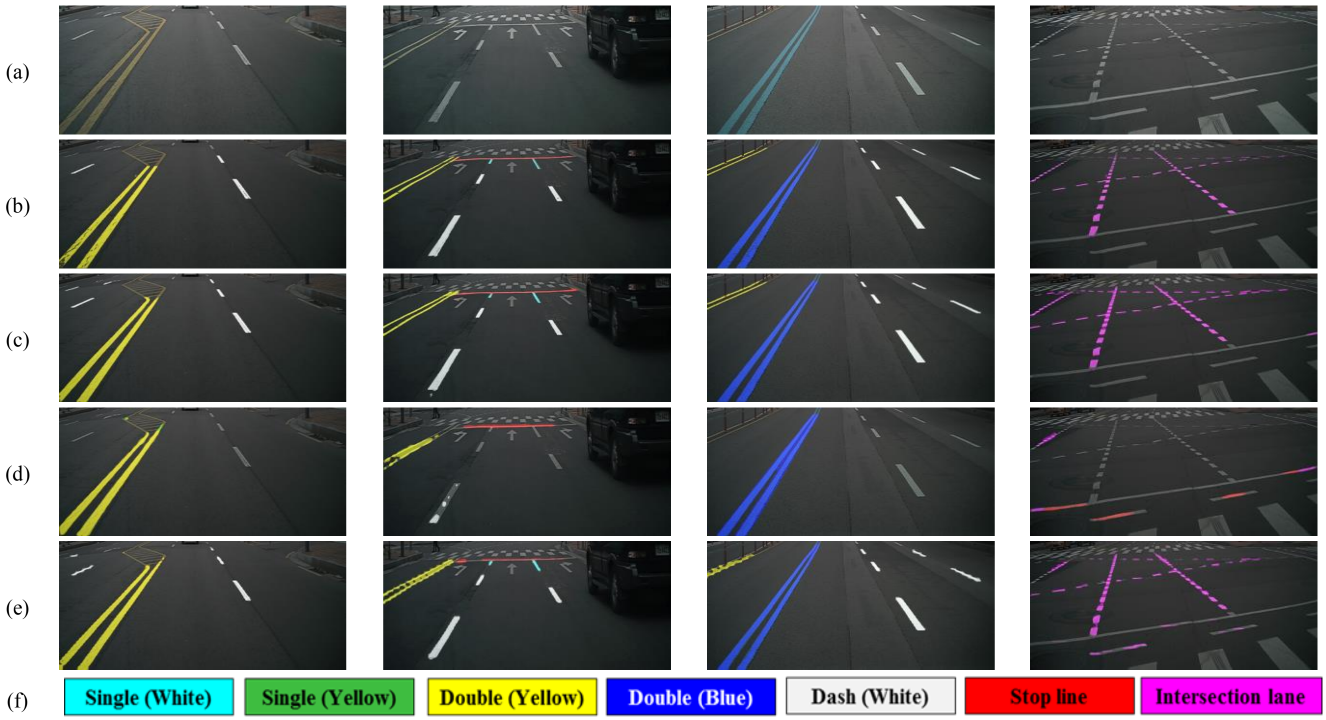


Figure 7. Result of networks. First two rows show original input(a) and ground truth(b). Each row shows result of other network, weighted FCN(c). SegNet[9](d), PSPNet[11](e) and meaning of each color(f).

TABLE I. PERFORMANCE OF SEMANTIC SEGMENTATION

	SegNet[9]	***FCN***[14]	***PSPNet***[11]	***Weighted-FCN***
Recall	0.4157	0.6238	0.9552	0.9439
Precision	0.6467	0.4021	0.6034	0.7295
F-score (beta = 1)	0.5061	0.4890	0.7396	0.8230
F-score (beta = 2)	0.4477	0.5618	0.8554	0.8915

Recall, precision and the F-score of network are shown in TABLE 1. SegNet and the FCN have poor F-score about 0.5. SegNet results miss small lane segments such as dash-lanes and lanes of the intersection and FCN cover the larger area than ground truth.

On the other hand, PSPNet and weighted-FCN show much improved performance. PSPNet detected most of the existing road lane ground truth. However, PSPNet segmented peripheral area of lane as road lane, so the precision performance is low. Weighted FCN has high recall, almost similar with PSPNet, and the highest precision and F-score.

Fig. 7 shows the results of each network and there are four different circumstances. The meaning of colors is described at the bottom of Fig. 7. Fig. 7 (b) shows the ground truth of the image. Fig. 7 (c) through (e) are the results of Weighted-FCN, SegNet, and PSPNet, respectively and meaning of color is in (f). In case of SegNet, the dotted line is not detected well. In the case of PSPNet, the lane placed far away shows a square patch, while the results of weighted-FCN are well-segmented.

The segmentation performance of each class of weighted-FCN used in the mapping algorithm is arranged in the following TABLE 2. Single yellow lane is confused with fallen leaves and also there are many blurred lanes. Also, in the case of the stop line, there was a lot of erroneous detection from back bumper of the car ahead and the characters on the road.

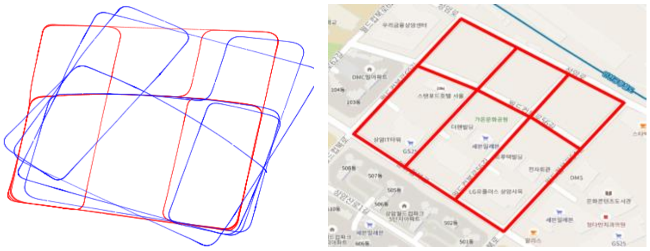
TABLE II. DETAIL PERFORMANCE OF FCN FOR EACH CLASS

Pixels / Class	**GT**	**Detected**	**True Positive**	**Recall**	**Precision**
Double (Yellow)	6,293,736	7,498,914	5,965,214	**0.9178**	**0.7955**
Double (Blue)	3,392,489	3,631,902	3,351,149	**0.9878**	**0.9027**
Dash-line (White)	8,524,116	11,981,358	8,217,144	**0.9040**	**0.7438**
Single (Yellow)	1,782,581	2,797,984	1,606,148	**0.9310**	**0.5740**
Single (White)	3,582,976	4,000,174	3,012,459	**0.8408**	**0.7531**
Intersection lane	3,328,552	4,016,123	3,261,003	**0.9797**	**0.8120**
Stop line	2,077,671	3,242,243	1,792,153	**0.9107**	**0.5836**

C. HD map reconstruction result

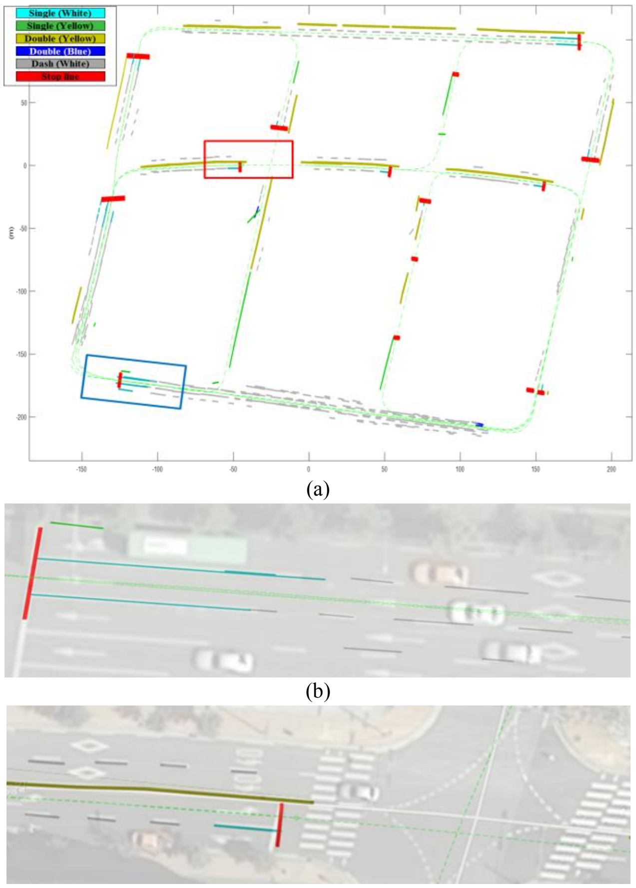
High definition map is created based on the results of lane segmentation. The area used for mapping is a region where it is difficult to create a map, as there are large number of vehicles, wide roads and narrow roads are mixed. Data for mapping was obtained by making several round trips of 400m and 400m. The data acquired with the experimental vehicle is about 15 minutes long and the total length is about 4.8 km.

Fig.8 shows trajectories of vehicle. Since the original trajectory is calculated using only odometry, there is a lot of error accumulation when the vehicle rotates in intersection.



(a) (b)

Figure 8. Trajectory of our experiment. (a) shows raw trajectory(blue) and optimized trajectory(red). (b) is ground truth map from Google Earth.



(a)

(b)

(c)

Figure 9. Final map result. (a) Final map of Sangam-dong, Mapo-gu, Seoul, Korea. (b)-(c) Detail map of blue and red part from final map.

Fig.8 (a) shows the comparison between optimized trajectory and original trajectory. The whole path is shown in the red line plot on the Google Earth image as Fig.8 (b). The final HD map generated with the optimized odometry and landmark is shown in Fig.9. Fig.9 (a) shows the entire HD map result. Five types of lanes exist in the area. In this area, many high-rise buildings and vehicles exist so, shadow of buildings and lanes occluded by cars exist.

We used low cost GPS to fix the initial vehicle pose and restored the latitude and longitude coordinates of entire map using only odometry. The generated global HD map is projected onto the aerial view image. Fig.9 (b) and Fig.9 (c) shows the details of map superimposed on the aerial view image. White single lane, stop line in Fig.9 (b) and dash-line, yellow double line in Fig.9 (c) matches exactly with aerial image.

VI. Conclusion

In this paper, we propose a system for recognizing lane information and generating a reliable map that is comparable to actual road map. A fully convolutional neural network with the new learning strategy is trained from our dataset. The result of image segmentation in the class imbalance problem shows better performance than existing state of the art. The pixels from segmentation result is converted to lane feature and used to generate HD map. The ICP is used for local SLAM and iSAM is used for global graph optimization. The result of the proposed method reconstructed map well even in environment that many vehicles exist.

In the actual experiment, the variance of pitch angle of the experiment vehicle is more influential on map performance. The proposed method estimates the distance of road lane from vehicle with position in image. So, when the positional inaccuracy of road lane segmentation results increases along with the distortion of image due to lean of vehicle, the inaccuracy of segmentation affects the final performance of the map. For the further works, if we add an algorithm to correct the pitch angle of the vehicle, it will be possible to generate more precise map than the result of this paper.

References

[1] J. Jeong, Y. Cho, and A. Kim, "Road-SLAM : Road marking based SLAM with lane-level accuracy," in *IEEE Intelligent Vehicles Symposium*, 2017, pp. 1736–1743.

[2] Y. Lu and Y. Chen, "Monocular Localization in Urban Environments using Road Markings Optimization," in *IEEE Intelligent Vehicles Symposium*, 2017, pp. 468–474.

[3] J. Dai, T. J. Liu, and H. Lin, "Road Surface Detection and Recognition for Route Recommendation," in *IEEE Intelligent Vehicles Symposium*, 2017, pp. 121–126.

[4] S. Tsutsui, T. Kerola, and S. Saito, "Distantly Supervised Road Segmentation," in *International Conference on Computer Vision*, 2017, pp. 174–181.

[5] S. Lee, N. Kim, J. Kim, and T. Lee, "VPGNet : Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition," in *International Conference on Computer Vision*, 2017, pp. 1947–1955.

[6] 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 *Computer Vision and Pattern Recognition*, 2016.

[7] J. Fritsch, T. Kuhnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 1693–1700.

[8] C. C. Lin and M. S. Wang, "A vision based top-view transformation model for a vehicle parking assistant," *Sensors.*, vol. 12, no. 4, pp. 4431–4446, 2012.

[9] V. Badrinarayanan, A. Kendall, R. Cipolla, and S. Member, "SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[10] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *Computer Vision and Pattern Recognition*, 2016, pp. 2881–2890.

[12] J. Zhu, N. Zheng, and S. Du, "Point-to-Line Metric Based Iterative Closest Point with Bounded Scale," in *IEEE Conference on Industrial Electronics and Applications*, 2009, pp. 3032–3037.

[13] M. Kaess, S. Member, A. Ranganathan, and S. Member, "iSAM : Incremental Smoothing and Mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, 2008.