

딥러닝 기반의 의미론적 영상 분할을

이용한 주행 보조 시스템

Driving Assist System using Semantic Segmentation based on Deep Learning

김정환*, 이태민*, 임준홍*

Jung-Hwan Kim*, Tae-Min Lee*, Joonhong Lim*

Abstract

Conventional lane detection algorithms have problems in that the detection rate is lowered in road environments having a large change in curvature and illumination. The probabilistic Hough transform method has low lane detection rate since it exploits edges and restrictive angles. On the other hand, the method using a sliding window can detect a curved lane as the lane is detected by dividing the image into windows. However, the detection rate of this method is affected by road slopes because it uses affine transformation. In order to detect lanes robustly and avoid obstacles, we propose driving assist system using semantic segmentation based on deep learning. The architecture for segmentation is SegNet based on VGG-16. The semantic image segmentation feature can be used to calculate safety space and predict collisions so that we control a vehicle using adaptive-MPC to avoid objects and keep lanes. Simulation results with CARLA show that the proposed algorithm detects lanes robustly and avoids unknown obstacles in front of vehicle.

요약

기존의 차선 검출 방법들은 곡률과 날씨 변화가 큰 도로 환경에서 검출률이 낮다. 확률적 허프 변환을 이용한 방법은 에지와 직선의 각도를 이용해서 차선을 검출함으로 곡선과 악천후일 때 검출률이 낮다. 슬라이딩 윈도우 방법은 윈도우로 이미지를 분할해서 검출하기 때문에 곡선 형태의 차선도 검출하지만 어퍼인 변환을 사용하기 때문에 도로의 경사율에 영향을 받는다. 본 논문에서는 다양한 외부 환경에서도 차선을 강인하게 검출하고 장애물을 회피하기 위한 딥러닝 기반의 주행 보조 시스템을 제안한다. VGG-16기반의 SegNet으로 입력 영상을 의미론적으로 분할해서 차선을 검출한다. 검출한 차선과의 이격 거리를 계산하고 안전범위를 산출해서 차량이 차선의 중앙을 주행하도록 제어한다. 또한, 전방의 미확인 물체와 충돌이 예상되면 운전자에게 경보를 주고 Adaptive-MPC로 차량을 제어해서 충돌을 회피하는 알고리즘도 제안한다. CARLA로 시뮬레이션한 결과 제안한 알고리즘은 곡률이 큰 차선과 다양한 환경에서도 강인하게 차선을 검출하고 전방의 안전범위를 계산하여 충돌을 회피하는 것을 볼 수 있다.

Key words : Lane Detection, Deep Learning, Semantic segmentation, Hough Transformation, Sliding Window

* Dept. of Electronic Engineering, Hanyang University

★ Corresponding author

E-mail : jhlim@hanyang.ac.kr, Tel : +82-31-400-4043

※ Acknowledgment

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No.2017R1D1A1B03033806).

Manuscript received Mar. 6, 2020; revised Mar. 17, 2020; accepted Mar. 18, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

최근 5년간 고속도로에서 발생한 사망사고의 원인 가운데 졸음운전이 1위이다. 졸음운전과 주시태만으로 사망하는 사고율을 낮추기 위해서 최근에는 주행 조향 보조 시스템(LKAS : Lane Keeping Assist System), 첨단 운전자 지원 시스템(ASCC : Advanced Smart Cruise Control) 등 다양한 안전 시스템들이 연구 및 개발되고 있다. 이러한 안전 시스템들은 차선을 인식하는 것에서 시작한다. 카메라로 좌·우의 차선을 인식해서 차량이 차선의 중앙을 주행하고 있는지 판단한다. 차선을 이탈할 시에는 운전자에게 경고를 주고 조향 제어를 통해서 차량을 안전범위 내에 위치시킨다. 그러므로 차선 인식 알고리즘은 자율주행에서 핵심 기술 중에 하나다.

기존에 차선을 인식하는 방법으로는 대표적으로 2가지가 있다. 첫 번째는 허프(Hough) 변환을 이용하는 것으로 차선의 특징인 직선을 검출하는 방법이다. 일반적으로 이러한 검출 방법을 사용하기 전에 인식률을 높이기 위해서 다음과 같은 전처리 과정을 수행한다. 통상적으로 입력되는 이미지는 왜곡과 노이즈(Noise)를 가지고 있기 때문에 왜곡 보정과 가우시안(Gaussian) 필터를 이용해서 제거한다. 이후 연산 수행 시간을 줄이기 위해서 관심영역(ROI : Region Of Interest)을 설정한다. 관심영역을 설정한 후에는 에지(Edge)와 색 공간을 이용한 검출 결과를 서로 교집합 하여 차선의 인식률을 높인다. 이러한 전처리 과정 후에 허프 변환을 사용해서 직교 좌표계의 표현을 극 좌표계 형식으로 나타내준다. 극 좌표계는 원점에서 떨어진 수직 거리 ρ 와 이 수직선이 x 축과 이루는 각도 θ 로 직선을 표현할 수 있고 일정한 간격으로 검출함으로써 정밀하다는 장점이 있다. 하지만 이 방법은 영상 내 모든 화소를 가지고 변환을 수행하기에 연산량이 크다는 단점이 있다. 이를 해결하기 위해서 화소를 무작위로 선택하여 연산하는 확률적 허프 변환을 수행한다. 최종적으로 허프 변환 후 직선의 각도 θ 를 이용해서 차선을 검출할 수 있다. 일반적으로 차선은 수평이 아닌 수직방향으로 있으므로 각도 θ 를 제한하여 검출하면 된다. 그러나 이러한 방법은 허용한 각도 범위 외의 차선을 검출하는데 어려움이 있다. 두 번째 방법인 슬라이딩 윈도우(Sliding Window)는 허용 각도 범위로 발생하는 단점을 극

복하기 위해서 제안된 알고리즘이다. 앞서 설명한 허프 변환 방법과 다른 점은 어파인(Affine) 변환을 통해서 입력 이미지를 Bird's-eye view 형태로 변환한 후 슬라이딩 윈도우로 차선을 검출한다는 것이다. 전체 이미지에서 설정한 관심영역 내의 영상을 히스토그램(Histogram)으로 변환한 후 지정한 윈도우의 개수만큼 영역을 분할해서 차선을 검출한다. 허프 변환과 달리 영상을 분할하여 윈도우로 검출하기 때문에 곡선 형태의 차선도 강인하게 검출해내는 장점이 있다. 하지만 어파인 변환을 사용하기 때문에 도로의 경사에 따라서 영상이 뭉개지는 현상이 발생된다. 그리고 설정되는 윈도우의 개수에 따라서 검출률이 달라지고 초당 프레임 수(FPS : Frame Per Second)에 영향을 미친다는 단점이 있다. 이 외에도 주간, 야간, 노을 등 조도가 변하는 상황과 차선의 마모와 악천후 등 다양한 주변 환경에서 인식률이 낮아지는 문제점들이 발생한다[1, 2].

본 논문에서는 이러한 단점을 해결하기 위해서 VGG-16기반의 SegNet을 이용한 의미론적 영상 분할 방법으로 차선을 검출할 것을 제안한다. 차선을 검출 후에는 차량과 차선 간의 이격거리를 계산해서 중앙을 주행하도록 제어한다. 또한 미확인 물체와의 충돌이 예상되면 운전자에게 경보를 주고 Adaptive-MPC로 충돌을 회피한다.

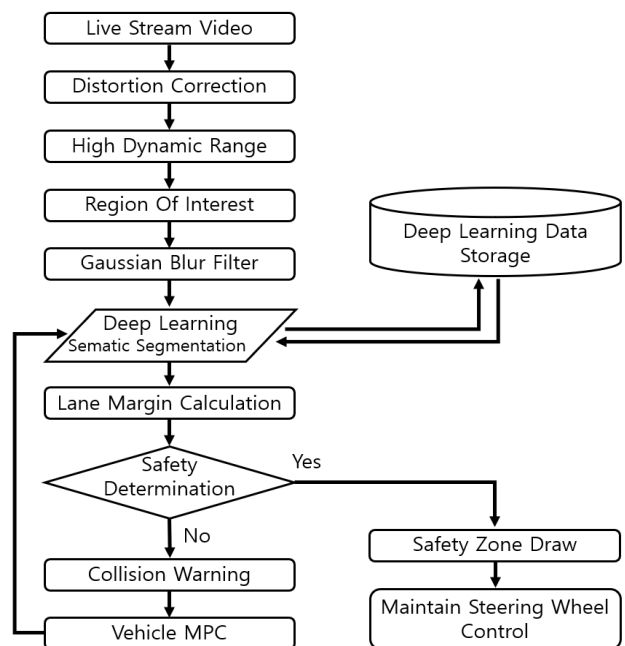


Fig. 1. Driving assist algorithm using semantic segmentation. 그림 1. 의미론적 분할을 이용한 주행 보조 알고리즘

II. 차선 검출 및 안전성 판별

카메라로부터 입력받는 영상은 광각 렌즈의 특성에 의해서 이미지가 왜곡되는 현상이 발생된다. 그러므로 카메라 캘리브레이션(Calibration)으로 카메라 내·외부의 파라미터를 구해서 왜곡을 보정한다. 왜곡을 보정한 후에는 HDR(High Dynamic Range)을 수행한다. HDR은 한 장의 이미지에서 밝고 어두움의 차이가 큰 환경이 동시에 있을 때 이를 보정해주는 알고리즘이다. 이를 이용하면 역광의 환경에서도 강인한 이미지를 얻을 수 있다. 색 보정 후에는 ROI를 설정한다. 영상의 모든 영역에 대해서 차선 검출 알고리즘을 수행하면 불필요한 영역도 포함되기 때문이다. ROI 설정으로 영상 처리 속도를 증가시킬 수 있다. 차선과 안전 범위 영역을 검출하는 것이 목적이므로 영상의 하단부분을 관심 영역으로 지정한다. 입력되는 영상은 카메라의 내·외부적인 요인에 의해서 노이즈가 포함될 수 있다. 딥러닝으로 의미론적 영상 분할을 할 때에 노이즈 신호가 증폭되어서 인식률이 저하될 수도 있다. 그러므로 본 논문에서는 ROI를 지정한 후에 입력되는 이미지에 가우시안 스무딩(Smoothing) 필터를 수행한다. 노이즈를 제거한 후에는 차선을 검출하기 위해서 딥러닝을 이용한 의미론적 영상 분할을 수행한다. 의미론적 영상 분할은 이미지 내의 객체를 픽셀 단위로 분류하는 알고리즘이다. 영상 분할로 객체의 검출과 분류를 동시에 수행할 수 있는 장점이 있다. 또한, 그림 2와 같이 검출한 객체가 경계 상자 안에 있는 것과는 달리 비정형 모양도 검출할 수 있다.

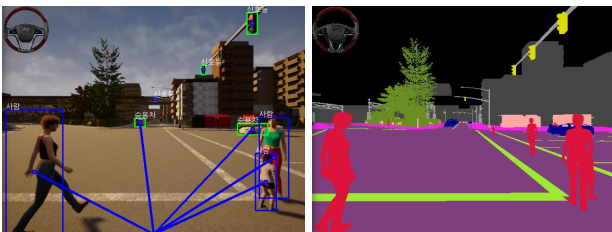


Fig. 2. Semantic segmentation algorithm comparison.
그림 2. 의미론적 영상 분할과 객체 감지 알고리즘 비교

기존의 CNN(Convolutional Neural Network) 신경망들은 분류목적으로 사용되기에 이미지 내의 공간 정보를 손실하게 된다. 일반적인 CNN은 입력

되는 객체 이미지에 설정한 크기의 커널(Kernel)로 스트라이드(Stride)하면서 컨벌루션 연산을 수행한다. 이를 통해서 특징 맵을 추출하게 되는데 이러한 특징 맵은 입력한 이미지보다 크기가 작아지게 되면서 정보의 소실을 불러온다. 이를 사전에 막기 위해서 패딩(Padding)으로 보완하지만 활성화 맵에서 풀링(Pooling)을 수행하면서 이미지가 계속해서 작아지게 되는 다운 샘플링(Down Sampling)이 수행된다. 그리고 객체 이미지를 분류하기 위해서 3차원 데이터를 1차원 배열로 변환하는 플랫텐(Flatten) 또한 수행한다. 이후 완전연결(FC : Fully-connected) 층을 통해서 1차원 배열 전체를 심층 신경망으로 연결한다. 최종적으로 소프트맥스(Softmax)를 통해서 객체 이미지를 분류하게 되는데 이러한 일련의 과정으로 인해서 이미지 내의 공간 정보가 없어지게 된다.

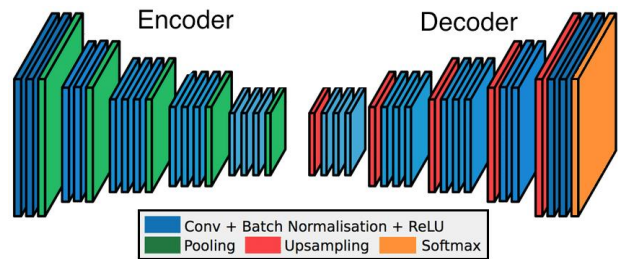


Fig. 3. Encoder-decoder structure.
그림 3. 인코더-디코더 구조

차선과 차량의 안전범위는 비정형 모양으로도 검출될 수 있으므로 본 논문에서는 이미지 내의 공간 정보를 활용해서 픽셀 단위로 검출할 것을 제안한다. 이미지 내의 공간 정보를 유실하지 않기 위해서 그림 3과 같이 기존의 CNN 구조를 인코더 - 디코더(Encoder - decoder) 구조로 변경한다. 인코더에서는 풀링을 사용하고 디코더에서는 업 샘플링(Upsampling)을 사용하는 구조이다. 이러한 모델들로는 SegNet, UNet, PSPNet, FCN(Fully Convolutional Network) 등이 있다[3]. 제안한 알고리즘에서는 SegNet 모델을

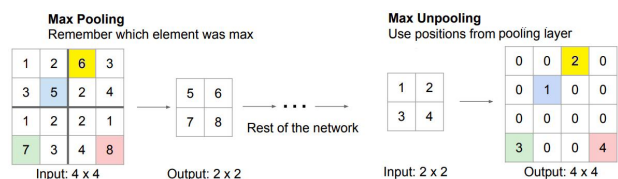


Fig. 4. Remember max-pooling index position in encoder.
그림 4. 인코더 내의 최대 풀링 인덱스 위치 기억

사용하고 활성화 함수로는 ReLU를 사용했다. SegNet은 Skip Connections을 사용하지 않는다. 대신에 그림 4와 같이 인덱스(index) 위치를 기억한다.

인코더에서 다운 샘플링을 수행할 때 최대 풀링(Max-Pooling)의 인덱스를 기억했다가 디코더에서 업 샘플링을 수행할 때에 최대 언풀링(Max-Unpooling)을 해당 인덱스로 배치시킨다. 이 방법은 특징 맵을 모두 저장해두고 업 샘플링을 하는 방식에 비해서 메모리를 효율적으로 사용할 수 있다는 장점이 있다[4, 5].

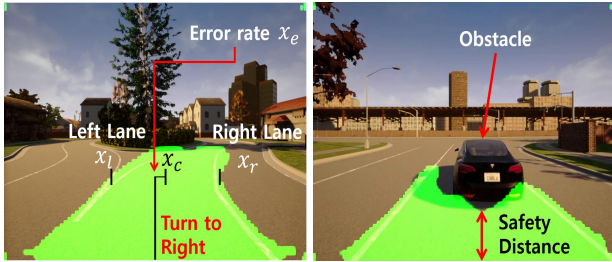


Fig. 5. Lane departure and safety determination.

그림 5. 차선 이탈 및 안전성 판별

SegNet을 이용한 의미론적 영상 분할을 통해서 좌·우 양단의 차선이 검출되면 안전 범위까지 공간에 표시된다. 그러므로 그림 5와 같이 이미지 내에 표시되는 공간의 가로·세로 길이와 차량의 중점으로부터의 거리를 계산하면 차선 이탈과 안전성을 판별할 수 있다. 카메라가 항상 차량의 중앙에 있다는 가정하에 이미지의 중점과 검출된 양단 차선의 중점을 서로 비교한다면 차량이 차선의 중앙으로부터 얼마나 이격 되었는지를 알 수 있다. 이격 정도는

$$x_e = x_c - \frac{x_r - x_l}{2} \quad (1)$$

와 같이 구할 수 있다. 여기서 x_e 는 이격 정도이고 x_c 는 차량 중앙 좌표의 x 값이다. x_l 과 x_r 은 검출된 좌·우 차선의 좌표 x 값이다.

안전 범위는 의미론적 영상 분할을 통해서 이미지에 표시되는 공간 정보로 구할 수 있다. 차량 중앙으로부터 세로의 길이가 일정 범위 이내로 짧아지면 운전자에게 충돌 경보를 알린다.

III. 주행 보조 알고리즘

차량과 차선의 이격 정도를 계산한 후에 안전성 판별을 통해서 차량을 제어한다. 본 논문에서는 차량을 제어하는 방법으로 적응형 모델 예측 제어기(Adaptive Model Predictive Control)를 사용한다. Adaptive-MPC를 수행하기 위해서 제어기의 상태 벡터를 모델링하면

$$z = [x, y, v, \phi] \quad (2)$$

와 같이 나타낼 수 있다. 여기서 x 는 차량의 x 축 위치이고 y 는 차량의 y 축 위치를 의미한다. v 는 속도, ϕ 는 요(yaw) 각도이다. 입력 벡터는

$$u = [a, \delta] \quad (3)$$

와 같다. a 는 가속도이고 δ 는 핸들조향 각도이다. MPC 제어기는 경로 추적을 위해서 비용 함수를

$$\min Q_f(z_{T,ref} - z_T)^2 + Q \sum (z_{t,ref} - z_t)^2 + R \sum u_t^2 + R_d \sum (u_{t+1} - u_t)^2 \quad (4)$$

$$\text{subject to } z_{t+1} = Az_t + Bu_t + C$$

$$|u_{t+1} - u_t| < du_{\max}$$

$$|u_t| < u_{\max}, z_0 = z_{0,ob}$$

$$v_{\min} < v_t < v_{\max}$$

$$u_{\min} < u_t < u_{\max} \quad (5)$$

와 같이 최소화 한다. 여기서 z_{ref} 는 목표 경로와 속도를 의미한다. 목표 경로는 차선의 중앙이다. 제약 조건의 대상은 조향 속도와 각도 그리고 차량의 속도와 입력값이다. 제어 대상인 차량은 비선형적인 모델을 가지고 있다. 그러므로 선형화 과정을 통해서 상미분방정식의 형태로 변환한다. 그리고 이산 시간에서 차량을 제어하기 위해서 샘플링 시간 dt 와 Forward Euler Discretization을 사용한다. 최종적으로 차량의 동역학 모델은

$$z_{t+1} = Az_t + Bu_t + C \quad (6)$$

$$A = \begin{bmatrix} 1 & \cos(\bar{\phi})dt & -\bar{v}\sin(\bar{\phi})dt \\ 0 & \sin(\bar{\phi})dt & \bar{v}\cos(\bar{\phi})dt \\ 0 & 1 & 0 \\ 0 & \frac{\tan(\bar{\delta})}{L}dt & 1 \end{bmatrix} \quad (7)$$

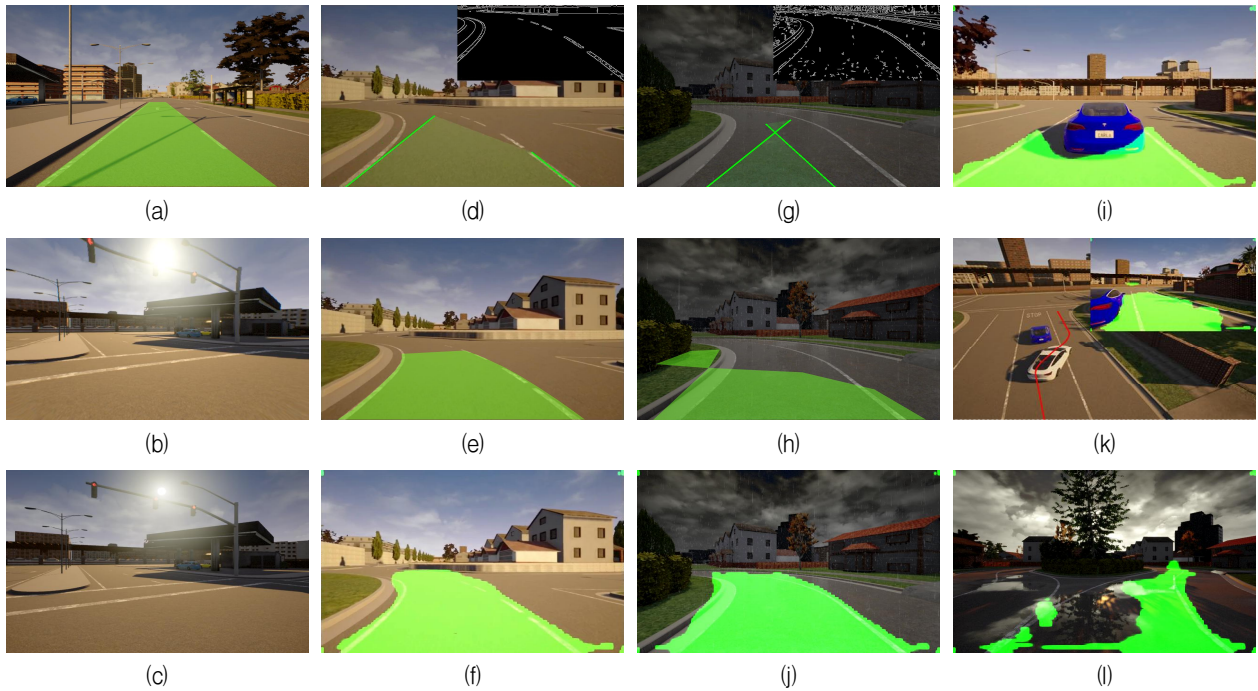


Fig. 6. Experimental results of lane detection and driving assistance using semantic segmentation based on deep learning.
그림 6. 딥러닝 기반의 의미론적 영상 분할을 이용한 차선 검출 및 주행 보조 실험 결과

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ dt & 0 \\ 0 & \frac{\bar{v}}{L \cos^2(\bar{\delta})} dt \end{bmatrix} \quad (8)$$

$$C = \begin{bmatrix} \bar{v} \sin(\bar{\phi}) \bar{\phi} dt \\ -\bar{v} \cos(\bar{\phi}) \bar{\phi} dt \\ 0 \\ -\frac{\bar{v} \bar{\delta}}{L \cos^2(\bar{\delta})} dt \end{bmatrix} \quad (9)$$

와 같이 나타낼 수 있다[6].

Table 1. Lane detection algorithms comparison.

표 1. 차선 인식 알고리즘들 비교

Parameters	Hough Method	Sliding Window	Proposed Method
FPS	57	38	45
Accuracy	62.55	76.59	84.46
Safety Area	×	×	○

IV. 시뮬레이션 결과

제안한 알고리즘을 구현한 시스템 사양은 CUP i7-8700K, GPU 1080TI, RAM 32GB이고 Ubuntu 18.04에서 실행된다. 의미론적 영상 분할을 위해서 라벨링(Labeling)된 12,764장의 데이터를 가지고 전

이 학습(Transfer Learning)을 수행한다. 인코더 모델은 VGG-16을 사용한다. 미니 배치(Mini batch) 방식으로 128개씩 적용하고 에포크(epoch)는 500으로 한다. 과대적합(Overfitting)을 피하기 위해서 20% 확률의 드롭아웃(Dropout)을 적용한다. 학습률은 0.001이며 최적화(Optimizer) 함수로는 Adam을 사용해서 손실값을 최소화하는 가중치와 편향값을 구한다. 손실 함수는 크로스 엔트로피(Cross Entropy)를 사용한다. 실차량을 제어하기에는 위험부담이 있어서 CARLA 프로그램의 0.9.7버전을 사용하여 구현한다[7]. 입력되는 이미지는 640×480의 크기로 60 FPS에 VGA급 영상이다. ROI로 상단 30%를 제외시킨다. 그림 6의 (a)는 학습 데이터 중 일부를 나타낸 것으로 라벨링된 Ground Truth는 초록색으로 표현한다. 라벨링된 이미지와 추정된 이미지의 픽셀(Pixel)을 서로 비교해서 정확도를 측정한다. (b)는 입력되는 원본 이미지이다. (c)는 이미지의 왜곡을 보정하고 HDR 알고리즘을 적용한 결과이다. (d)와 (g)는 확률적 허프 변환을 이용한 차선 검출 결과로 직선 형태의 에지를 사용하기 때문에 검출률이 떨어지는 것을 알 수 있다. (e)와 (h)는 25개의 Sliding Window를 적용한 결과로 허프 변환보다는 곡선에 강인하지만 여전히 약천후 등 다양한 도로 환경에 영향을 많이 받는 것을 볼 수가

있다. (f)와 (i)는 제안한 알고리즘의 결과로 딥러닝 기반의 의미론적 영상 분할을 통해서 차량 전방의 차선을 잘 검출하는 것을 알 수가 있다. (j)는 전방에 물체가 있을 때의 결과를 나타낸 것으로 전방 물체와의 안전 범위를 계산해서 충돌 경고를 해준다. 안전 범위는 20m로 설정했다. (k)는 설정한 안전 범위 안으로 물체가 다가올 때 충돌을 회피하기 위해서 Adaptive-MPC로 차량을 제어한 결과를 나타낸다. (l)은 혼잡한 도로 환경과 미학습된 물체의 반사로 인해서 오검출된 결과를 나타낸다. 표1과 같이 기존 방법들과 비교한 결과, 제안한 방법의 정확도가 높은 것을 볼 수가 있다. 그리고 차선 인식뿐만 아니라 공간 정보를 활용해서 안전 범위를 인식할 수가 있다.

VI. 결론

기존의 차선 검출 알고리즘은 곡률 변화가 큰 차선이나 조도가 변하는 상황, 차선의 마모, 악천후 등 다양한 도로 환경에서 차선 검출률이 낮아지는 문제점들이 있다. 이러한 문제점들을 해결하기 위해 본 논문에서는 HDR 전처리와 VGG-16기반에 SegNet을 이용한 차선 검출 알고리즘을 제안한다. 더 나아가 의미론적 영상 분할의 특징을 이용해서 안전 범위를 계산하고 충돌 경보를 예측한다. 충돌이 예상되면 Adaptive-MPC로 차량을 제어해서 충돌을 회피한다. 시뮬레이션 결과 기존 알고리즘보다 곡률이 큰 차선과 다양한 외부 환경에 강인한 것을 알 수가 있다. 동시에 차량 전방의 안전 공간도 판별할 수 있다. 하지만 그림 6의 (l)처럼 학습하지 못한 환경과 물체에 대해서는 오동작하는 것을 볼 수가 있다. 향후 연구에서는 이러한 문제점들을 해결하기 위해서 다양한 외부 환경에서의 차선 이미지들과 물체들을 학습시키고 인코더 모델들을 비교해서 정확도와 알고리즘 연산시간 또한 향상시킬 예정이다.

References

[1] Peijiang Chen, Junhao Jiang, "Algorithm Design of Lane Departure Warning System Based on Image Processing," *2018 2nd IEEE Advanced*

Information Management, Communicates, Electronic and Automation Control Conference, pp.2497-2501, 2018. DOI: 10.1109/IMCEC.2018.8469505

[2] Bubly Barua, Shuva Biswas, Kaushik Deb, "An Efficient Method of Lane Detection and Tracking for Highway Safety," *1st International Conference on Advances in Science, Engineering and Robotics Technology*, 2019.

DOI: 10.1109/ICASERT.2019.8934664

[3] Le-Anh Tran, My-Ha Le, "Robust U-Net-based Road Lane Markings Detection for Autonomous Driving," *2019 International Conference on System Science and Engineering*, pp.62-66, 2019.

DOI: 10.1109/ICSSE.2019.8823532

[4] Junlan Chen, Ke Wang, Huanhuan Bao, Tao Chen, "A Design of Cooperative Overtaking Based on Complex Lane Detection and Collision Risk Estimation," *IEEE Access*, vol.7, pp.87951-87959, 2019. DOI: 10.1109/ACCESS.2019.2922113

[5] Jannik Fritsch, Tobias Kühnl, Franz Kummert, "Monocular Road Terrain Detection by Combining Visual and Spatial Information," *Transactions on Intelligent Transportation Systems*, Vol.15, No.4, pp.1586-1596, 2014. sDOI: 10.1109/TITS.2014.2303899

[6] Liya kolmanovsky, Calos Guardiola, Luigi Glielmo, *Automotive model predictive control models methods and applications*, Springer, 2010. DOI: 10.1007/978-1-84996-071-7

[7] "CARLA Simulator," <http://carla.org/>

BIOGRAPHY

Jung-Hwan Kim (Member)



2013 : BS degree in System Control Engineering, Hoseo Univ.
2015 : MS degree in Electronic Systems Engineering, Hanyang Univ.
2016~current : PhD degree course in Electronic Engineering, Hanyang Univ.

Tae-Min Lee (Member)

2016 : BS degree in Computer Engineering, Mokwon Univ.
2016~current : MS degree course in Electronic Engineering, Hanyang Univ.

Joonhong Lim (Member)

1979 : BS degree in Electronic Engineering, Seoul National Univ.
1981 : MS degree in Electrical Engineering, KAIST.
1986 : PhD degree in Electrical and Computer Engineering, Univ. of Iowa.

1986~1989 : Assistant Professor, Dept. of Electronic Engineering, Korea Aerospace Univ.

1989~current : Professor, Dept. of Electronic Engineering, Hanyang Univ.