

세그멘테이션 기반 차선 인식 네트워크를 위한 적응형 키포인트 추출 알고리즘

이상현 김덕수*

한국기술교육대학교

{master1511, bluekds}@koreatech.ac.kr

Adaptive Key-point Extraction Algorithm for Segmentation-based Lane Detection Network

Sang-Hyeon Lee Duksu Kim*

School of Computer Engineering, Korea University of Technology and Education

요약

딥러닝 기반의 이미지 세그멘테이션은 차선 인식을 위해 널리 사용되는 접근 방식 중 하나로, 차선의 키포인트를 추출하기 위한 후처리 과정이 필요하다. 일반적으로 키포인트는 사용자가 지정한 임계값을 기준으로 추출한다. 하지만 최적의 임계값을 찾는 과정은 큰 노력을 요구하며, 데이터 세트(또는 이미지)마다 최적의 값이 다를 수 있다. 본 연구는 사용자의 직접 임계값 지정 대신, 대상의 이미지에 맞추어 적절한 임계값을 자동으로 설정하는 키포인트 추출 알고리즘을 제안한다. 본 논문의 키포인트 추출 알고리즘은 차선 영역과 배경의 명확한 구분을 위해 줄 단위 정규화를 사용한다. 그리고 커널 밀도 추정을 사용하여, 각 줄에서 각 차선의 키포인트를 추출한다. 제안하는 알고리즘은 TuSimple과 CULane 데이터 세트에 적용되었으며, 고정된 임계값 사용 대비 정확도 및 거리오차 측면에서 1.80%p와 17.27% 향상된 결과를 얻는 것을 확인하였다.

Abstract

Deep-learning-based image segmentation is one of the most widely employed lane detection approaches, and it requires a post-process for extracting the key points on the lanes. A general approach for key-point extraction is using a fixed threshold defined by a user. However, finding the best threshold is a manual process requiring much effort, and the best one can differ depending on the target data set (or an image). We propose a novel key-point extraction algorithm that automatically adapts to the target image without any manual threshold setting. In our adaptive key-point extraction algorithm, we propose a line-level normalization method to distinguish the lane region from the background clearly. Then, we extract a representative key point for each lane at a line (row of an image) using a kernel density estimation. To check the benefits of our approach, we applied our method to two lane-detection data sets, including TuSimple and CULane. As a result, our method achieved up to 1.80%p and 17.27% better results than using a fixed threshold in the perspectives of accuracy and distance error between the ground truth key-point and the predicted point.

키워드: 차선 인식, 커널 밀도 추정, 키포인트, 딥러닝

Keywords: Lane detection, Kernel density estimation, Key-point, Deep learning

*corresponding author: Duksu Kim/ School of Computer Engineering, Korea University of Technology and Education (bluekds@koreatech.ac.kr)

1. 서론

차선 인식(lane detection)은 자율주행 시스템을 위한 핵심 기반 기능 중 하나다[1,2]. 차선 이탈 경고 시스템 (Lane Keeping Assist System, LKAS)과 같은 주행 보조 시스템에서 활용되는 등 자율주행을 넘어 다양한 분야에서 폭넓게 활용되는 기술이기도 하다. 전통적인 차선 인식 기술은 캐니 에지(Canny edge) 검출, 허프 변환(Hough transformation) 등을 통한 차선의 외곽선(edge)을 검출하는 방법[3-5]과 색상 필터로 차선의 색을 추출하는 방법[6] 등을 사용한다. 하지만, 최근 딥러닝(deep learning) 기술의 발전에 따라 딥러닝에 기반을 둔 차선 인식 기술들이 활발히 연구되고 있으며, 전통적인 차선 인식 알고리즘보다 높은 정확도를 보여주고 있다[7-9].

딥러닝 기반 차선 인식 기술은 크게 이미지 세그멘테이션(image segmentation) 기반 모델[7,8], 앵커(anchor) 기반 모델[10,11], 그리고 매개 변수 예측(parameter prediction) 모델 [12,13]로 구분된다. 이미지 세그멘테이션 기반 차선 인식 모델은 기존에 연구되어왔던 이미지 세그멘테이션 네트워크를 그대로 사용할 수 있다는 장점이 있다. 이러한 이유로 최신 연구들에서 가장 많이 채용되며, 전체적으로 좋은 성능을 보여주는 접근법이다. 이미지 세그멘테이션 기반 모델은 이미지를 차선과 배경 영역으로 분할(segmentation)하고, 후처리를 통해 분할된 차선 영역에서 키포인트(key-point)를 추출하는 방법으로 차선을 인식한다. 분할된 영역은 전체 이미지에서 차선이 존재하는 모든 픽셀의 집합으로, 해당 픽셀 중 각 차선을 대표하는 키포인트를 추출해야 한다. 그리고 각 줄에서 추출된 키포인트들이 연결되어 각 차선 객체를 생성한다 (3.1장). 키포인트를 추출하는 대표적인 방법의 하나는, 차선일 확률이 특정 임계값을 넘는 픽셀들을 선택하는 것이다. 하지만 설정한 임계값에 따라 추출되는 키포인트의 수 및 분포가 크게 달라질 수 있으며, 적절한 임계값 설정에 많은 노력이 필요하다. 또한 하나의 차선 영역에 대해서 추출된 여러 개의 키포인트 중 각 줄(이미지의 가로줄)에서 각 차선을 대표하는 키포인트들을 결정하는 후처리 과정도 필요하다는 단점이 있다.

본 논문은 고정된 임계값 설정 없이 각 줄마다, 각 차선에 대한 단일 키포인트를 추출하는 적응형 키포인트 추출 알고리즘을 제안한다 (4장). 동일한 이미지 내에서도 차선의 위치 및 형태의 특성을 반영하고 차선 영역과 배경의 구분을 명확히 하기 위해, 본 논문은 줄 단위 정규화 기법을 제안한다 (4.1장). 그리고, 커널 밀도 추정을 통해 각 줄에서 각 차선의 대표 키포인트를 추출한다 (4.2장). 제안하는 기술은 명시적인 임계값 지정 없이 장면에 동적으로 적응하여 단일 키포인트를 추출한다는 특징을 가진다.

제안하는 적응형 키포인트 추출 알고리즘의 효과를 확인하

기 위해, 본 연구는 ResNet[14]을 이용한 이미지 세그멘테이션 기반 차선 인식 네트워크를 구성하였다. 그리고 수동으로 찾은 최적의 임계값을 사용한 것과 본 연구가 제안하는 적응형 키포인트 추출 알고리즘을 적용한 결과를 비교하였다 (5장).

실험에는 차선 인식 성능 검증을 위해 널리 사용되는 두 개의 차선 인식 데이터 세트(TuSimple[15]과 CULane[24])를 사용하였으며, 키포인트 추출 정확도 및 추출된 키포인트와 정답 키포인트 사이의 거리 오차 측면에서 성능을 비교하였다. 그 결과 적응형 키포인트 추출 알고리즘이 수동으로 최적의 임계값을 찾고 적용하는 방법 대비, 정확도 및 키포인트 거리 오차 측면에서 각각 최대 1.80%p 및 17.27% 높은 성능을 보여주었다. 이러한 결과는 본 연구가 제안하는 적응형 키포인트 추출 알고리즘의 효과 및 유용성을 보여준다.

2. 관련 연구

Huval 등[16]은 합성곱신경망(CNN)을 고속도로 주행을 위한 차선 인식 및 객체 탐지 분야에 사용할 수 있음을 보여주었다. 이후 딥러닝을 이용한 차선 인식 연구들이 활발히 진행되었으며, 대표적인 접근법에는 이미지 세그멘테이션(image segmentation) 기반 차선-배경 분리[7-9,17,18], 앵커 기반 차선 추출[11,19], 그리고 매개 변수 예측(parameter prediction)[12,13] 모델 등이 있다.

앵커 기반 차선 인식 방법은 사전에 정의된 앵커를 통해 고정된 형태의 차선을 추출하는 기법이다. 이 방법은 고정된 앵커를 차선에 근사시켜 문제를 해결하기 때문에, 보이지 않는 환경(no-visual-clue) 문제에 대해 강인하다는 장점이 있다. Li 등[19]은 Faster R-CNN[20]의 영역 제안 네트워크(region proposal network)와 유사한 차선 제안 유닛(LPU, Line Proposal Unit)을 제안하였다. 그 결과, 앵커 기반 차선 인식으로 TuSimple 데이터 세트에서 96.87%의 높은 정확도를 보여주었다. 하지만 미리 정의된 차선의 형태를 기반으로 차선을 분류하는 앵커 기반 차선 인식 방법은, 다양한 차선의 형태에 대한 유연성이 떨어진다는 한계를 가진다.

매개 변수 예측 기반 차선 인식 기술은 차선을 표현하는 고차원 다항함수를 정의하고, 함수의 계수를 추론하는 방법으로 차선을 추출한다. 이 방법은 네트워크에서 차선의 매개 변수만을 출력하기 때문에, 이미지 픽셀 단위로 예측을 수행하는 세그멘테이션 기법 대비 추론 속도가 빠르다는 장점이 있다. Tabelini 등[12]은 매개 변수 예측을 위한 PolyLaneNet을 제안하였으며, 115 FPS (frames per seconds)의 차선 인식 속도를 달성했다. 또한 Liu 등[13]은 자연어 처리에서 주로 사용되는 트랜스포머를 채용하여 다

항식의 계수를 예측하는 방법을 제안하였다. 그 결과, TuSimple 데이터 세트에 대해 96.18%의 높은 정확도와 420 FPS의 빠른 성능을 보여주었다. 하지만 이러한 매개 변수 예측 네트워크들 역시 최대 차선의 수를 미리 정해 놓아야 한다는 한계를 가진다. 또한 차선을 직접 추출하지 않고 고차원 다항함수로 근사시키는 방법으로, 이미지 세그멘테이션 기반 방법보다 비교적 낮은 정확도를 보여준다.

이미지 세그멘테이션 기반 차선 인식 방법은 이미지를 차선과 배경으로 나누어 차선을 추출하는 직관적인 전략을 사용한다. 이 방법은 잘 연구된 이미지 세그멘테이션 네트워크를 쉽게 활용할 수 있는 장점이 있어 활발하게 연구되었다 [7,8,17,18,21]. 이미지 세그멘테이션은 픽셀 단위로 차선을 예측하는 특성이 있다. Pan 등[8]은 이미지의 가로줄(또는 슬라이스, slice) 단위로 합성곱 연산을 수행하는 SCNN을 제안하여, 네트워크가 전역 문맥(global context)을 파악함으로써 가려진 부분에 대해 풍부하게 학습할 수 있게 했다. 그 결과 TuSimple 데이터 세트에서 96.53%의 높은 성능을 달성하였다. Zheng 등[7]은 슬라이스 단위 합성곱을 병렬 처리하는 RESA를 제안하여, SCNN 대비 10배 이상의 슬라이스 단위 합성곱 속도를 보여주었다. Yongqi Dong 등[21]은 SCNN과 RNN을 동시에 사용하여, TuSimple 데이터 세트에 대한 98.19%의 정확도를 보여주었다.

이미지 세그멘테이션 네트워크는 입력 이미지에서 차선 영역을 가려내는 역할을 하며, 차선 영역들로부터 각 차선을 추출하는 과정은 후처리 과정에서 이루어진다. Pan 등 [8]은 이미지를 차선과 배경으로 이진 분류하는 것이 아닌, 각 차선을 구분해서 세그멘테이션하는 SCNN을 제안하였다. 하지만 이러한 다중 클래스 세그멘테이션(multi-class segmentation)을 사용하는 경우, 입력 이미지 내 차선의 개수가 지정되어야 한다는 단점이 있다. Neven 등[9]은 차선 인식을 인스턴스 세그멘테이션(instance segmentation) 문제로 바꾸어, 임의의 차선 개수를 가진 이미지에서도 차선을 각 차선 영역을 잘 분리했다. 또한, 이미지에서 투시 변환(perspective transformation)을 예측하는 H-Net을 제안하여, 차선 영역의 픽셀 군집으로부터 차선을 다항식으로 모델링하였다. Qu 등[25]은 FOLOLane 네트워크를 제안했다. FOLOLane은 차선 영역 세그멘테이션과 함께, 현재 위치에서 같은 차선으로 분류될 다음 차선의 위치를 세그멘테이션하여 키포인트를 개별 차선으로 분리하는 성능을 높였다. Zheng 등[26]은 저수준 특징과 고수준 특징을 상호보완적으로 사용하는 네트워크 구조와 전역 문맥을 반영하는 관심 영역 수집(ROI Gather)을 사용하는 CLRNet을 제안했다. FOLOLane과 CLRNet은 tuSimple 데이터 세트에 대해 각각 96.56과 97.89라는 높은 F1스코어를 달성하였다.

본 연구 또한 이미지 세그멘테이션 기반 차선 인식 네트워크를 사용한다. 하지만 본 연구는 세그멘테이션 결과로부터 키포인트 추출하는 후처리 과정의 정확도를 높이는 방법을

제안한다는 차이점이 있다.

3. 개요

본 장에서는 본 연구가 대상으로 하는 세그멘테이션 기반 차선 인식 방법의 일반적인 절차를 살펴본다 (3.1장). 그리고 본 연구가 제안하는 적응형 키포인트 추출 알고리즘의 개요를 설명한다 (3.2장).

3.1 세그멘테이션 기반 차선 인식 절차

세그멘테이션 기반 차선 인식 기법은 일반적으로, 차선-배경 분리와 차선 모델링, 두 단계로 진행된다. 첫 번째 차선-배경 분리 단계는 입력 이미지를 차선과 배경으로 나누는 과정으로, 이미지 세그멘테이션 기법이 적용된다. 즉, 이미지에서 차선 부분을 전경(foreground)으로, 그리고 기타 부분을 배경(background)으로 분리하는 것과 같다. 이미지 세그멘테이션에는 일반적으로 인코더-디코더(encoder-decoder) 형태의 딥러닝 네트워크가 사용되며, 대표적인 딥러닝 모델로는 U-Net[22]이 있다. 차선-배경 분리 단계의 출력은 이미지 각 픽셀(pixel)에 대한 차선일 확률과 배경일 확률이다.

세그멘테이션 기반 차선 인식 기법의 두 번째 단계는 차선 모델링이다. 차선-배경 분리 결과물은 픽셀별 차선일 확률일 뿐, 어떤 차선에 속하게 되는지에 대한 구분이 없다. 따라서, 차선으로 인식된 픽셀들을 각 차선으로 분류하고 그것들을 연결하여 개별 차선 객체를 생성하는 작업이 필요하다. 차선 모델링의 품질은 키포인트 추출의 정확도와 밀접한 관계가 있다. 키포인트는 일반적으로 주어진 임계값을 사용해서 추출되며, 임계값에 따라 결과가 크게 달라질 수 있다. 그리고 기존 연구들은, 임계값은 경험적 또는 실험적인 방법으로 수동으로 결정하거나, 차선일 확률과 배경일 확률을 단순히 비교하여 이진화하는 방법을 사용한다[7,8]. 본 연구는 이러한 수동적 임계값 설정의 단점을 극복할 수 있는 적응형 키포인트 추출 알고리즘을 제안한다.

3.2 제안하는 알고리즘 개요

Figure 1은 본 연구가 제안하는 차선 인식 알고리즘의 개요이다. 본 연구는 세그멘테이션 기반의 차선 인식 기법을 사용하며, 시스템의 첫 번째 모듈은 차선-배경 분리 딥러닝 네트워크다. 딥러닝 네트워크의 입력은 RGB 이미지이며, 출력은 입력과 같은 크기를 가지는 두 개의 2차원 벡터인 확률맵(probability map)이다. 각각의 확률맵은 입력 이미지의 각 픽셀이 차선일 확률과 배경일 확률을 표현한 값을 가진다. 딥러닝 네트워크는 인코더-디코더(encoder-decoder) 구

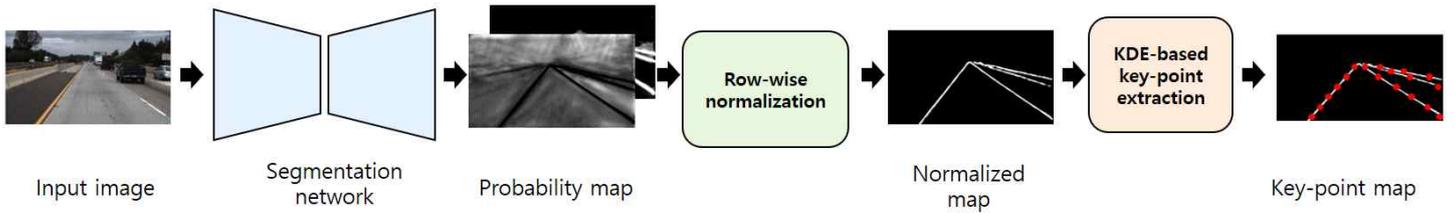


Figure 1: Overview of the proposed lane detection algorithm

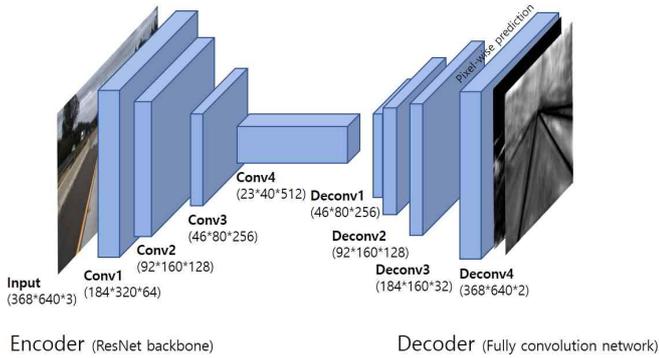


Figure 2: Segmentation network architecture

조로 디자인하였으며, 백본으로는 ResNet[14]을 사용하였다. Figure 2는 본 연구가 사용한 세그멘테이션 네트워크의 구조를 보여준다.

확률맵은 키포인트 추출 모듈로 전달된다. 키포인트 추출 모듈은 줄 단위 정규화(row-wise normalization), 커널 밀도 추정(kernel density estimation), 그리고 키포인트 추출(key-point extraction) 단계로 구성된다. 줄 단위 정규화 과정은 네트워크에서 출력된 확률맵을 이미지의 가로줄(row) 단위로 분리한 후 정규화를 수행하여 차선 영역을 강조한다 (4.1장). 커널 밀도 추정 단계에서는, 각 줄에서 강조된 각 차선 영역(높은 확률값 군집)들의 확률값 분포를 미분이 가능한 연속 함수 형태로 변환한다. 그리고 변환된 함수의 극점 위치의 픽셀을 해당 차선을 대표하는 키포인트로 추출한다 (4.2장). 커널 밀도 추정을 통한 키포인트 추출은 일정 간격(예, 3-5 픽셀) 단위로 수행하고, 키포인트 맵(key-point map)을 생성한다.

차선은 각 높이에 있는 키포인트들을 서로 연결하는 방법으로 모델링 된다. 연결 기준은 키포인트들 사이의 거리로, 가장 가까운 키포인트 쌍을 연결해서 차선을 확장한다. 즉, 시작 줄(예, 가운데줄)의 키포인트들 각각을 개별 차선으로 설정하고, 윗줄(또는 아랫줄)에 있는 키포인트 중 가장 가까운 키포인트와 연결하여 해당 차선을 확장하는 방식이다. 만약 두 키포인트 쌍의 거리가 일정 임계값(예, 50픽셀) 이상이거나 매칭 할 수 있는 키포인트가 없는 경우, 남은 키포인트는 새로운 차선으로 설정한다. 이 과정을 모든 줄에

대해 반복하면 차선 인식이 종료된다.

4. 적응형 키포인트 추출 알고리즘

4.1 줄 단위 정규화 기반 차선 영역 분리

입력 이미지는 세그멘테이션 네트워크를 통해, 각 픽셀에 대한 차선일 확률값을 가지고 있는 확률맵(probability map)으로 변환된다. Figure 3-(b)는 네트워크에서 출력된 확률맵의 예를 보여준다. 차선 영역의 픽셀들이 대체로 높은 값을 가지는 경향을 보인다. 하지만, 차선 영역에 대한 확률값은 입력 이미지 속 차선의 가시성에 큰 영향을 받는다. 다른 물체에 가려져 차선이 보이지 않거나 점선 형태 차선이 가시성이 떨어지는 대표적인 예다. 가시성이 낮은 차선 영역의 경우, 배경과 차선 영역 사이의 확률값의 차이가 작아져서 낮은 차선 추출 정확도를 가지게 된다.

본 연구는 소프트맥스(softmax) 정규화를 사용하여, 차선 - 배경 구분의 명확성 문제를 해결한다. Figure 3-(c)는 확률맵 전체에 대해 소프트맥스 기반 정규화를 수행한 결과다. 정규화를 통해 차선 영역의 구분이 좀 더 명확해진 것을 확인할 수 있다. 하지만, 배경 영역에서 높은 확률값을 가지고 있던 영역이 같이 강조(높은 값을 가지는)되는 현상을 발견하였다. 이는 같은 이미지 내에서도 위치, 차폐, 빛의 세기 등에 따라 서로 다른 확률 분포를 가지기 때문이다. 또한 확률맵 전체에 대한 정규화는, 차선이 없는 영역(예, 지평선 위쪽 영역)까지 정규화에 반영하기 때문에 일부 차선 외 영역이 강조되는 현상을 심화시킨다.

본 연구는 이러한 문제점을 해결하기 위해, 확률맵 전체가 아닌 확률맵의 각 줄(행) 단위로 정규화를 수행한다. 이미지 내 각 영역에 대해 차선의 가시성 및 환경(예, 빛의 세기 등)은 서로 다를 수 있지만, 동일 줄 내에서는 이러한 요소들이 높은 일관성을 가지는 경향이 있다는 특성이 있다. 줄 단위 정규화는 이러한 통찰에 기반을 둔 전략이다. Figure 3-(d)는 줄 단위 정규화의 결과로, 차선 영역이 더 명확히 강조된 모습을 확인할 수 있다. 또한 세그멘테이션(Figure 3-(b)) 및 확률맵 전체 수준 정규화(Figure 3-(c))의 결과에서는 비교적 차선으로 인식되기 쉬웠던 가드레일

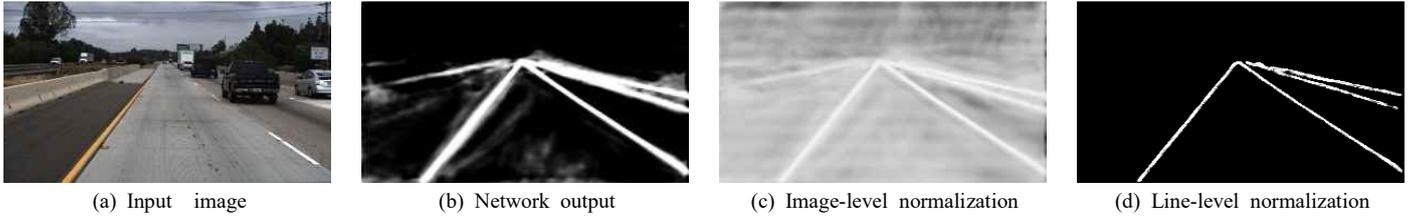
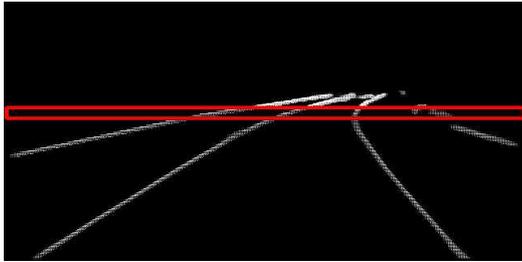


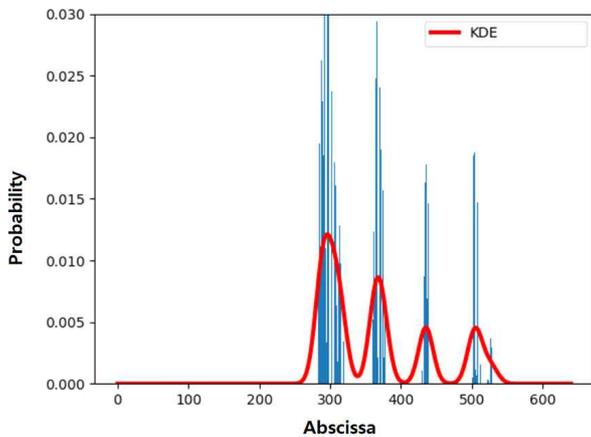
Figure 3: These images show the input image (a), the output of the segmentation network (b), the image-level normalization result (c), and the line-level normalization result (d).



(a) Input image



(b) Normalized image



(c) The result of the kernel density estimation process

Figure 4: The process of key-point extraction by the kernel density estimation

(Figure 3-(a)의 왼쪽 벽)이, 줄 단위 정규화를 거쳤을 때는 다른 차선에 비해 낮은 확률값에 의해 차선이 아닌 것으로 정확히 판단된 것을 볼 수 있다. 줄 단위 정규화 과정을 거친 확률맵은 커널 밀도 기반 키포인트 추출 단계로 전달되어 키포인트 맵으로 변환된다.

지평선 절단: 차선이 없는 영역, 즉 지평선 위 영역에 대

해서는 정규화 등과 같은 연산이 불필요하다. 본 연구의 세그멘테이션 네트워크는 각 픽셀이 차선일 확률과 배경일 확률 두 가지 값을 출력한다. 본 연구는 차선일 확률이 배경일 확률보다 큰 픽셀 중, 가장 위쪽의 픽셀을 기준으로 지평선을 설정하였다.

4.2 커널 밀도 추정 기반 키포인트 추출

정규화된 확률맵에서 차선 영역에 속하는 여러 개의 픽셀들이 배경 영역 대비 높은 확률값들을 가지게 된다. Figure 4는 정규화된 확률맵의 한 줄에 대한 확률값 분포를 보여준다. 각 차선을 대표할 키포인트를 추출하기 위해서 특정 임계값을 기준으로 키포인트 후보들을 추출한 후, 후보들 사이에서 적절한 키포인트를 선택해야 한다. 하지만 이미지에 따라 그리고 각 차선 영역에 따라 확률값 분포가 크게 달라질 수 있어서, 적절한 임계값을 정하기는 쉽지 않다.

본 연구는 특정 임계값 설정 없이, 다양한 형태의 확률 분포에서 각 차선을 대표하는 단일 키포인트를 추출하기 위해, 커널 밀도 추정(kernel density estimation)[23]을 사용한다. 커널 밀도 추정은 관측 데이터로부터 확률의 분포를 표현하는 확률 밀도 함수를 도출하는 기술로, 비모수적(non-parametric)으로 모집단을 추정하는 다양한 분야에서 활용된다[27,28]. 커널 밀도 추정은 확률 밀도 추정 과정에서 커널 함수를 사용하는 것이 특징인데, 커널 함수는 원점을 중심으로 대칭이며 적분값이 1인 비음수 함수를 뜻한다. 이러한 커널 함수는 유니폼, 삼각형, 이중제곱 함수 등 다양한 함수로 구현될 수 있으며 본 연구에서는 일반적으로 사용되는 가우시안 커널(gaussian kernel)을 사용했다.

즉, 커널 밀도 추정을 통해 이산적인 확률값 분포가 연속공간에 대한 함수로 표현된다 (Figure 4-(c)).

최종 키포인트는 도출된 확률밀도함수의 극값(maximum point)을 찾는 것으로 추출된다. 즉, 극값을 가지는 위치(픽셀)가 해당 차선의 키포인트가 된다. 커널 밀도 추정 기반 키포인트 추출은 줄 단위로 진행되며, 일정 간격으로 수행된다. 본 연구는, 줄(픽셀) 간격을 늘려가며 차선 모델링의 성능을 측정한 결과, 3~5 픽셀 간격이 전반적으로 좋은 성능을 보이는 것을 확인하였다 (Table 4와 Table 5). 추출된 키포인트들은 키포인트 맵(key-point map)으로 저장된다. 키

포인트 맵은 차선 추출 모듈로 전달되고, 차선으로 모델링 된다 (3.2장).

5. 결과 및 분석

5.1 데이터 세트

본 연구는 제안하는 알고리즘의 효과를 평가하기 위해 서, 차선 인식 네트워크의 평가에 널리 사용되는 CULane[24] 과 TuSimple[15] 데이터 세트를 사용하였다. Table 1은 두 데이터 세트의 특징과 본 연구의 실험에서 학습 (training) 및 테스트(test)에 사용한 데이터 세트의 크기를 보여준다. CULane은 13만 장 이상의 이미지를 포함하는 대규모 데이터 세트로, 도심지의 다양한 환경에서 수집된 데이터를 포함한다. 차선이 합쳐지거나 차선 일부가 존재하지 않는 등 복잡한 상황을 포함한다. 또한 차량 내부에서 촬영됨에 따라, 전면 유리 반사 등 다양한 잡음(noise)이 존재하는 등 차선 인식 난이도가 높은 데이터도 많이 포함하고 있다는 특징을 가진다. TuSimple은 6,000장 규모의 데이터 세트로, 연속된 차선 인식 수행을 위해서 각 샘플마다 라벨링 되어있지 않은 19장의 연속된 프레임 이미지가 포함되어있다. 고속도로에서 주로 촬영되었기 때문에 차선의 형태 및 표시가 CULane 대비 명확한 편이다. 또한, 카메라를 차량의 상단에 고정한 후 촬영했기 때문에, CULane 대비 잡음이 적은 도로의 이미지로 구성되어 있다는 특징을 가진다.

5.2 성능평가 지표

본 연구는 제안하는 적응형 키포인트 추출 알고리즘의 적용 전후의 성능을 비교하기 위해, 차선 인식 정확도와 키포인트 거리 오차 지표를 사용하였다.

차선 인식 정확도(accuracy) : 차선 인식 정확도는 각 데이터 세트에서 정의하고 있는 측정 방법에 따라 측정하였다. TuSimple 데이터 세트의 경우, 차선의 각도가 일 때 $20/\sin(\theta)$ 픽셀을 정답 인정 거리 임계값으로 사용한다. 즉, 예측한 키포인트와 정답 키포인트 사이의 거리가 임계값 이하라면, 정확히 추론한 것으로 판단하여 정확도를 측정한다. CULane 데이터 세트의 경우, 예측 차선과 정답 차선의 키포인트를 이미지로 만든 후 두 이미지의 IOU(Intersection Over Union)를 측정한다. CULane 공식 벤치마킹 코드에서와 같이 IOU가 0.4 이상일 때 정확히 추론한 것으로 판단하며, F1 score를 기준으로 정확도를 제시한다.

키포인트 거리 오차: 키포인트 거리 오차(key-point distance error)는 정답 키포인트와 추출된 키포인트의 거리 차이의 평균으로, 계산법은 다음 수식 (1)과 같다.

Table 1 : Information of the data sets used in the experiments

Dataset	CULane	TuSimple
Training set size	16.3%	18.2%
Test set size	33.8%	29.8%
Resolution	12.1%	11.1%
Environment	Urban, Highway	Highway
# of lanes	<=4	<=5

$$Error_{dist} = \sum_i^n abs(GT_i - k_i) \quad (1)$$

수식 (1)에서 GT_i 는 i -번째 정답 키포인트의 위치, k_i 는 예측된 키포인트 중 GT_i 와 가장 가까운 키포인트의 위치다. 키포인트 거리 오차는 정답으로 판단된 키포인트에 대해서만 계산된다.

5.3 구현 상세

CULane과 TuSimple는 서로 다른 이미지 크기 및 비율을 가지고 있으며, 세그멘테이션 네트워크에 넣을 때는 이미지를 각각 $800*300$ 과 $640*368$ 로 축소했다. 세그멘테이션 네트워크 학습 시, 손실함수로는 교차엔트로피(cross-entropy)를 사용하였으며 미니 배치 크기를 4(이미지 4장)로 하였다. 도로 이미지에서 차선은 전체 이미지의 2~ 3% 수준의 적은 비율을 차지한다. 교차 엔트로피 사용에 있어 이러한 클래스 불균형 문제를 해결하기 위해, 차선에 속하는 픽셀에 대해 20배의 가중치를 부여하였다. 학습은 CULane은 250 epoch, TuSimple은 200 epoch 동안 진행하였다.

본 연구가 제안하는 적응형 키포인트 추출 알고리즘의 효과를 확인하기 위해, 학습된 세그멘테이션 네트워크를 기반으로 두 가지 버전의 차선 인식 시스템을 구현하였다.

- **임계값 기반 키포인트 추출 (Threshold)**: 고정된 임계값을 기준으로 키포인트를 추출하는 방법이다. 차선마다 여러 개의 키포인트가 추출될 수 있으며, 그 경우 가장 높은 확률값을 가지는 키포인트를 해당 차선의 대표 키포인트로 사용하였다. 임계값의 설정에 따라 키포인트 추출 결과 및 차선 인식 정확도가 크게 달라질 수 있다. 본 실험에서는 다양한 임계값으로 차선 인식을 수행하고, 그 중의 가장 좋은 성능을 비교군의 성능으로 사용하였다.
- **적응형 키포인트 추출 (Ours)**: 본 연구가 제안하는 적응형 키포인트 추출 기법을 적용한 시스템으로, 수동의 임계값 설정 과정 없이 차선별 단일 키포인트가 추출된다.

5.4 결과

Table 2는 두 데이터 세트에 대해, 본 연구가 제안하는 적

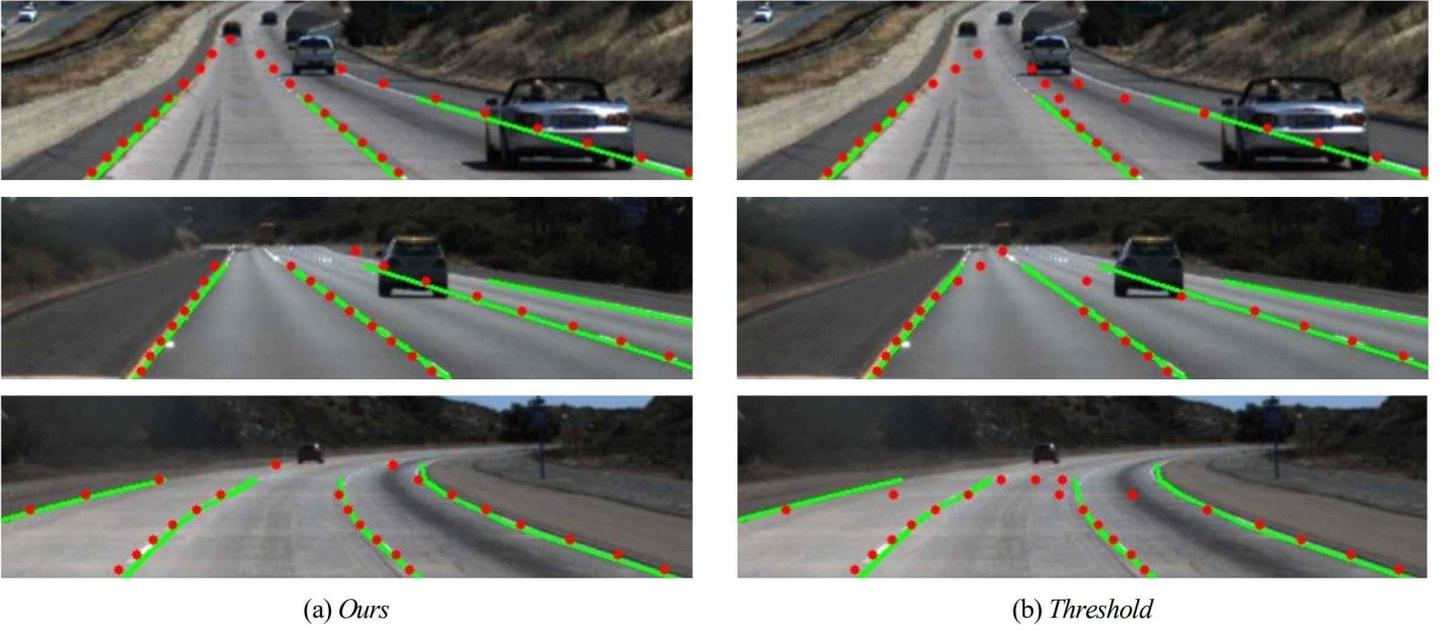


Figure 5 : These images show the key-point extraction results of two algorithms, Threshold and Ours. The green lines show the ground truth lane, and the red points are the key points extracted by the algorithm

Table 2: Performance comparison between two algorithms

	Threshold		Ours	
	Acc.	Error _{dist}	Acc.	Error _{dist}
TuSimple	92.96	330.28	94.76	273.25
CULane	69.98	336.21	70.71	279.39

응형 키포인트 추출 방법의 적용 전후의 성능 변화를 보여준다. 두 데이터 세트 모두에서 *Ours*가 *Threshold* 대비 더 높은 정확도(accuracy)와 낮은 키포인트 거리 오차(*Error_{dist}*)를 가지는 것을 볼 수 있다. TuSimple 데이터 세트의 경우, 정확도는 92.96%에서 94.76%로 1.80%p 향상되었다. 또한, 키포인트 거리 오차는 17.27%만큼 감소하였다. CULane 데이터 세트의 경우, *Ours*가 *Threshold* 대비 정확도와 키포인트 거리 오차를 각각 0.73%p 와 16.90%만큼 개선되었다.

Figure 5는 TuSimple 데이터 세트에 포함된 영상 중 하나에 대한 *Threshold*와 *Ours*의 키포인트 추출 결과다. 빨간 점은 추출된 키포인트들이며, 녹색 선은 라벨링 된 정답 차선이다. *Ours*가 *Threshold* 대비 차선 영역에 대한 키포인트를 더 정확하게 추출하는 모습을 확인할 수 있다. 특히, 차로부터 멀어질수록(지평선에 가까워질수록), *Threshold*의 키포인트 추출 결과에 오류가 큰 것을 확인할 수 있다. 반면, *Ours*는 지평선 영역까지 키포인트를 정확하게 예측하는 모습을 확인할 수 있다.

이러한 정량적 및 정성적 평가 결과는, 본 연구가 제안하는 적응형 키포인트 추출 알고리즘의 유용성을 보여준다.

Table 3 : Effect of the normalization method

		TuSimple	CULane
<i>Ours_{w/oNorm}</i>	Acc.	93.47	69.19
	Error _{dist}	287.62	302.75
<i>Ours_{ImgNorm}</i>	Acc.	93.31	68.72
	Error _{dist}	279.17	291.36
<i>Ours</i>	Acc.	94.76	70.71
	Error _{dist}	273.25	279.39

5.5 줄 단위 정규화의 효과

적응형 키포인트 추출 알고리즘은, 이미지 세그멘테이션 결과에서 차선 영역과 배경의 구분 명확성을 높이기 위해 줄 단위 정규화를 수행한다. 본 연구는, 줄 단위 정규화의 효과를 분석하기 위해 다음 두 가지 변형 알고리즘을 구현하고 그 성능을 비교해 보았다.

- *Ours_{w/oNorm}*: 세그멘테이션 네트워크의 출력에 대한 정규화 과정 없이 사용한 알고리즘으로, 정규화되지 않은 확률맵(Figure 3-(b))에 대해 커널 밀도 추정 기반 키포인트 추출을 수행한다.
- *Ours_{ImgNorm}*: 세그멘테이션 네트워크의 출력에 대해 줄 단위 정규화가 아닌 확률맵 전체에 대한 정규화(Figure 3-(c))를 수행하는 알고리즘이다.

Table 3은 서로 다른 세 가지 정규화 방법을 적용한 결과를 보여준다. 두 데이터 세트 모두에서 본 연구가 제안하는

Table 4: This table shows the performance (F1 score and $Error_{dist}$) of our algorithm with different line spacing for the CULane data set. The bold font denotes the best score, and the italic font means the second one for each column.

Line spacing	Acc. (F1 score)	FP	FN	$Error_{dist}$
1 pixel	0.668	0.572	<i>0.803</i>	280.23
2 pixels	0.633	0.562	0.725	282.71
3 pixels	0.707	0.627	0.811	<i>279.39</i>
5 pixels	<i>0.701</i>	0.692	0.710	265.23
7 pixels	0.667	<i>0.683</i>	0.652	300.27
10 pixels	0.638	0.614	0.664	287.14
15 pixels	0.585	0.592	0.578	292.76

줄 단위 정규화를 적용한 경우가 가장 높은 정확도를 달성하는 모습을 확인할 수 있다. TuSimple 데이터 세트의 경우, *Ours*가 *Ours_{w/oNorm}* 대비 정확도는 1.29%p, 키포인트 거리오차는 5.00% 개선되었다. CULane 데이터 세트에 대해서 *Ours*는 *Ours_{ImgNorm}* 대비 정확도를 1.99%p, 그리고 키포인트 거리오차를 4.10% 향상시켰다. 또한, *Ours*는 *Ours_{w/oNorm}* 대비 정확도를 1.52%p 향상시켰다.

이러한 결과들은 본 연구가 제안하는 줄 단위 정규화의 효과 및 유용성을 보여준다. 또한, 세그멘테이션에서 생성한 확률맵에서 영역의 구분이 불명확할수록(예, CULane 데이터 세트), 줄 단위 정규화가 더 높은 효과를 보여줌을 확인할 수 있다.

5.6 키포인트 추출 간격

본 연구가 제안하는 적응형 키포인트 추출 알고리즘은 일정 간격(줄)으로 키포인트를 추출한다. Table 4와 Table 5는 각각 TuSimple과 CULane 데이터 세트에서 줄 간격에 따른 차선 인식 정확도 및 키포인트 거리 오차의 변화를 보여준다. TuSimple 데이터 세트의 경우 3픽셀 간격에서 가장 좋은 정확도를 보여주며, 1~7 픽셀 간격에서도 준수한 정확도를 달성하는 것을 확인할 수 있다. CULane 데이터 세트에서는 3픽셀 간격에서 가장 높은 정확도를 얻었으며, 5픽셀 간격도 준수한 성능을 보이는 것을 확인할 수 있었다. 두 데이터 세트 모두에서, 가장 높은 정확도를 달성한 줄 간격이 더 낮은 정확도를 보이는 줄 간격보다 높은 키포인트 거리 오차를 보이는 경우들이 있다. 이는 키포인트 거리 오차는 정답으로 판정한 키포인트들에 대해서만 계산함에 따라 발생하는 현상이다. 하지만, 전반적으로 좋은 정확도를 달성하는 3-5 픽셀 간격에서는 키포인트 거리 오차도 준수한 수치를 보이는 것을 확인할 수 있다.

5.7 키포인트 추출 및 차선 인식 속도

Table 5: This table shows the performance (accuracy and $Error_{dist}$) of our algorithm with different line spacing for the TuSimple dataset. FN and FP mean false negative and false positive, respectively. The bold font denotes the best score, and the italic font means the second one for each column.

Line spacing	Acc. (F1 score)	FP	FN	$Error_{dist}$
1 pixel	92.18	0.143	0.093	275.50
2 pixels	<i>94.75</i>	<i>0.093</i>	0.039	263.36
3 pixels	94.76	0.091	<i>0.041</i>	<i>273.25</i>
5 pixels	93.38	0.099	0.050	286.25
7 pixels	92.95	0.117	0.064	297.72
10 pixels	79.92	0.461	0.476	285.50
15 pixels	79.23	0.460	0.475	289.62

Table 6. Processing times of each step in our method

Step		Runtime (ms)	FPS
Segmentation (network)		32.31	30.95
Lane modeling	Row-wise normalization	1.69	591.71
	KDE-based key-point Extraction	11.74	85.17
Total		45.74	21.86

Table 6은 본 연구가 제안하는 차선 인식 알고리즘의 처리 속도를 보여준다. 실험에는 AMD Ryzen 2990WX (4.2Ghz) CPU와 RTX Titan GPU로 구성된 리눅스 시스템을 사용하였으며, 이미지 20장을 처리하는 평균 시간을 측정했다.

차선 인식 과정에서 가장 많은 시간을 차지하는 것은 차선과 배경을 분리하는 세그멘테이션 단계로 평균 31 FPS의 속도를 보여주었다. 차선 모델링 단계는 줄 단위 정규화와 커널 밀도 추정 기반 키포인트 추출 단계로 구분해서 생각할 수 있으며, 각각 평균 592 FPS 및 85 FPS의 속도를 보여주었다. 이는 본 연구가 제안하는 차선 추출 알고리즘의 낮은 연산 복잡도를 보여준다.

6. 결론 및 향후 연구

본 논문은 차선 영역과 배경을 분리하는 세그멘테이션 네트워크의 결과로부터 차선 모델링을 위한 키포인트 추출 알고리즘을 제안하였다. 제안하는 알고리즘은 고정된 임계값을 사용하는 것이 아닌, 주어진 이미지 및 각 영역에 동적 적용하여 키포인트를 추출한다는 특징을 가진다. 본 논문은 세그멘테이션 결과인 확률맵에서 차선 영역과 배경의 구분을 명확히 해주는 줄 단위 정규화 방법을 제안하였다. 그리고 커널 밀도 함수를 이용하여 정규화된 확률맵에서 줄 단

위로 각 차선을 대표하는 키포인트를 추출한다. 제안된 적응형 키포인트 추출 알고리즘은 차선 인식 기술 성능평가에 널리 사용되는 TuSimple 및 CULane 데이터 세트들에 대해, 고정된 임계값 사용 방법 대비 정확도 및 키포인트 거리 오차에서 각각 최대 1.80%p 및 17.27% 높은 성능을 달성하였다. 이러한 결과는 제안하는 알고리즘의 효과 및 유용성을 보여준다.

본 연구에서는 실험적인 방법으로 키포인트 추출에 사용하는 줄 간격을 결정하였다. 향후 연구에서는 키포인트 추출에 사용되는 줄 간격을 자동으로 찾는 방법을 연구하여, 다양한 환경에서 줄 간격까지 동적으로 결정하는 알고리즘을 설계하고자 한다. 또한, TuSimple 대비 상대적으로 낮은 인식 정확도를 보여주는 CULane 데이터 세트에 대해 더 높은 정확도를 달성할 수 있는 딥러닝 기반 차선 인식 알고리즘을 연구하고자 한다.

감사의 글

본 논문은 2022년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지차제-대학 협력기반 지역혁신 사업(2021RIS-004)의 지원을 받아 수행되었음

References

- [1] J. Y. Baek, and M. C. Lee, "Lane Recognition Using Lane Prominence Algorithm for Unmanned Vehicles", *Journal of Institute of Control, Robotics and Systems (ICROS)*, Vol. 16, No. 7, pp. 625-631, 2010
- [2] K.-S. Lee, S.-W. Heo, and T.-H. Park, "A Lane Detection and Tracking Method Using Image Saturation and Road Width Data", *Journal of Institute of Control, Robotics and Systems (ICROS)*, Vol. 25, No. 5, pp. 476-483, 2019
- [3] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on intelligent transportation systems*, vol. 9, no. 1, pp. 16-26, 2008.
- [4] M. Aly, "Real time detection of lane markers in urban streets," *IEEE Intelligent Vehicles Symposium*. pp. 7-12, 2008.
- [5] G. Liu, F. Wörgötter, and I. Markelic, "Combining statistical hough transform and particle filter for robust lane detection and tracking," *IEEE Intelligent Vehicles Symposium*. pp. 993-997, 2010.
- [6] K. B. Kim and D. H. Song, "Real time road lane detection with ransac and hsv color transformation," *Journal of information and communication convergence engineering*, vol. 15, no. 3, pp. 187-192, 2017.
- [7] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "Resa: Recurrent feature-shift aggregator for lane detection," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3547-3554, 2021.
- [8] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [9] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: an instance segmentation approach," *IEEE intelligent vehicles symposium (IV)*. pp. 286-291, 2018.
- [10] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Olivera-Santos, "Keep your eyes on the lane: Attention-guided lane detection," *arXiv e-prints*, pp. arXiv-2010, 2020.
- [11] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending," in *European Conference on Computer Vision*. pp. 689-704, Springer, 2020.
- [12] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Polylanenet: Lane estimation via deep polynomial regression," in *2020 25th International Conference on Pattern Recognition (ICPR)*. pp. 6150-6156, 2021.
- [13] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," *Proc. of the IEEE/CVF winter conference on applications of computer vision*, pp. 3694-3702, 2021.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. of the IEEE conference on computer vision and pattern recognition*, pp.770-778, 2016.
- [15] Tusimple/tusimple-benchmark: [Online]. Available: <https://github.com/TuSimple/tusimple-benchmark/>
- [16] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, Fernando Mujica, Adam Coates, Andrew Y. Ng, "An empirical evaluation of deep learning on highway driving," *arXiv pre-print arXiv:1504.01716*, 2015.
- [17] Z. Qin, H. Wang, and X. Li, "Ultra fast

- structure-aware deep lane detection,” in European Conference on Computer Vision. pp. 276–291, Springer, 2020.
- [18] S. Yoo, H. S. Lee, H. Myeong, S. Yun, H. Park, J. Cho, and D. H. Kim, “End-to-end lane marker detection via row-wise classification,” Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 1006–1007, 2020.
- [19] X. Li, J. Li, X. Hu, and J. Yang, “Line-cnn: End-to-end traffic line detection with line proposal unit,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 1, pp. 248–258, 2019.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, 2015.
- [21] Y. Dong, S. Patil, B. van Arem, and H. Farah, “A hybrid spatial-temporal deep learning architecture for lane detection,” Computer-Aided Civil and Infrastructure Engineering, 2022.
- [22] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” International Conference on Medical image computing and computer-assisted intervention. pp. 234–241, Springer, 2015.
- [23] E. Parzen, “On estimation of a probability density function and mode,” The annals of mathematical statistics, vol. 33, no. 3, pp. 1065–1076, 1962.
- [24] CULane dataset. [Online]. Available: <https://xingangpan.github.io/projects/CULane.html>
- [25] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, Wei Zhang, “Focus on local: Detecting lane marker from bottom up via key point.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
- [26] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, Xiaofei He, “CLRNet: Cross Layer Refinement Network for Lane Detection” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
- [27] Seungki Min, Insung Ihm. Density Estimation Technique for Effective Representation of Light In-scattering. Journal of the Korea Computer Graphics Society, vol. 16, no. 1, 9-20, 2010.
- [28] De Gelder, E., Cator, E., Paardekooper, J. P., Den Camp, O. O., & De Schutter, B. Constrained sampling from a kernel density estimator to generate scenarios for the assessment of automated vehicles. In 2021 IEEE Intelligent Vehicles Symposium Workshops, pp. 203-208, 2021.

〈 저자 소개 〉



이상현

- 2023년 한국기술교육대학교 컴퓨터공학부 (공학사)
- 관심분야: 자율주행, 센서퓨전
- <https://orcid.org/0000-0002-3112-1513>



김덕수

- 2008년 성균관대학교 정보통신공학부 (공학사)
- 2014년 KAIST 전산학과 (공학박사)
- 2014년-2018년 한국과학기술정보연구원 선임연구원
- 2018년-현재 한국기술교육대학교 조교수
- 관심분야: 고성능컴퓨팅, 로보틱스, 홀로그래피 등
- <https://orcid.org/0000-0002-9075-3983>