

# 멀티 태스킹 딥러닝 기반의 도로 환경 인식과 객체 추적 기법 연구

윤도현·장건우·원종진·강연식\*

국민대학교 자동차공학전문대학원

## A Multi-task Deep Learning Method for Road Environment Recognition and Object Tracking

Dohyun Yoon · Keonwoo Jang · Jongjin Won · Yeonsik Kang\*

*Graduate School of Automotive Engineering, Kookmin University, Seoul 02707, Korea*

*(Received 28 August 2023 / Revised 18 October 2023 / Accepted 19 October 2023)*

**Abstract** : In this paper, we will be utilizing a deep learning model with camera sensors to detect objects, lanes, and road boundaries. By applying the ByteTrack technique, we ensure that there is stable tracking even when objects are occluded. Additionally, we will be designing a multi-tasking model that can perform various tasks simultaneously. This enables rapid environmental perception and efficient memory usage in autonomous driving systems. Moreover, this model allows achieving multi-task objectives with the use of a single neural network. This paper validates its viability through the Waymo Open Dataset. The results demonstrate that, in the case of image object detection, our multi-tasking model can outperform both DLT-Net and MultiNet in terms of accuracy and inference speed. Furthermore, relatively superior performance can be observed when detecting road boundaries and lanes as well.

**Key words** : Autonomous vehicle(자율 주행 자동차), Deep learning(딥러닝), Multi tasking(멀티 태스킹), Multi object tracking(다중 객체 추적)

### 1. 서론

자율주행 시스템은 합리적이고 신속한 환경인식을 수행해야 하며 이를 위해 다양한 센서를 활용한다. 이 중 카메라는 다른 센서에 비해 주변 환경을 인식하는 방법이 사람과 유사하고 가격이 저렴하여 대중적인 센서이다. 또한 카메라는 RGB 채널로 구성된 밀도 높은 이미지 데이터를 취득할 수 있는 센서로, 높은 의미론적 추론 성능을 얻을 수 있다.<sup>1)</sup> 최근 이러한 특성과 딥러닝을 활용하여 카메라 센서를 활용한 다양한 객체 검출 모델이 연구되고 있고, Yolo와 같은 객체 인식에 특화된 신경망이 적극적으로 사용되고 있다.<sup>2)</sup> 하지만, 딥러닝 모델을 활용하여 자율주행 시스템을 구성하게 될 경우, 다양한 성능과 연산속도에 대한 요구사항을 모두 만족시키는 신경망을 설계하는 것이 필요하다.

자율주행 시스템의 차량용 컴퓨팅 시스템은 일반적

인 딥러닝 서버와 같이 고성능 스펙을 기대하기 어렵다. 따라서 딥러닝 연산에 필요한 GPU 메모리와 연산부하를 고려한 모델 설계가 필요하다. 본 연구는 객체, 차선, 도로경계를 검출하는 각각의 개별 태스크를 동시에 처리할 수 있는 멀티태스킹 신경망을 구성하였다. 이를 통해 적은 파라미터로 딥러닝 모델을 구성하여 GPU 메모리 사용량과 연산시간을 줄일 수 있었고, 자율주행 시스템에 적합한 신경망을 구성할 수 있었다.

또한, 기존의 멀티태스킹 연구에서 주행 가능한 도로 영역을 인식하여 도로경계를 검출하였지만, 본 연구에서는 연석을 인식하는 태스크를 도로경계 검출 태스크로 정의하였다. 이를 통해 차선 검출 태스크에서 연석을 차선으로 오인식하는 문제를 해결함과 동시에 명확한 기준을 가진 도로경계 검출 결과를 도출해내어 자율주행 데이터로의 활용 가능성을 높일 수 있었다. Waymo

\*Corresponding author, E-mail: [ykang@kookmin.ac.kr](mailto:ykang@kookmin.ac.kr)

<sup>1)</sup>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.

open dataset을 통해 멀티 태스킹 모델에서 새롭게 정의한 개별 태스크들 간의 결합이 가능함을 확인하였고 이에 더해 객체 인식 후처리에 추적을 수행하여 더욱 안정적인 객체 검출을 가능하게 하였다.

섹션 2에서 자율주행 인식 관련 딥러닝과 객체 추적에 대해 살펴보고, 섹션 3에서 본 연구에서 사용된 멀티 태스크 러닝의 구현 방법에 대한 설명을 제공한다. 섹션 4에서는 신경망 후처리에 추적을 수행하는 방법을 설명하고, 섹션 5, 6에서 연구 결과와 향후 전망을 제시하면서 논문을 마무리한다.

## 2. 관련 연구

본 섹션에서는 자율주행 환경인식에 주로 사용되는 Object detection과 Semantic segmentation에 관련된 연구를 살펴본다. 그리고 다양한 작업을 동시에 수행하는 멀티 태스크 러닝 신경망의 사례를 살펴본 뒤, 객체를 추적하는 솔루션을 마지막으로 소개한다.

### 2.1 Object Detection

카메라 데이터를 이용한 객체 인식은 오래된 연구 주제이다. 최근 많이 사용되는 딥러닝 기반의 객체 인식 알고리즘은 2stage 방식과 1stage 방식으로 구분할 수 있다. 2stage 방식은 먼저 객체가 있을 것으로 추정되는 영역들을 검출하고, 이후 해당 영역들의 특징 정보를 추출하여 객체를 판별하고 정확한 위치를 추정한다. 이에 비해 1stage는 전체 이미지를 격자로 나누어 각 격자마다 객체의 존재 여부와 위치를 예측한다. 일반적으로 2stage 방식은 1stage 방식에 비해 정확도가 높지만 연산속도가 느리다는 한계가 존재한다. 따라서 본 연구는 자율주행 시스템에서 사용 가능한 수준의 연산 속도를 확보하기 위해 1stage 방식인 Yolo 시리즈를 토대로 신경망을 설계하고 후처리에 추적 기법을 추가하였다.<sup>3)</sup>

### 2.2 Semantic Segmentation

Semantic segmentation은 초기 모델인 FCN부터 시작하여 경량화된 신경망이 꾸준히 연구되고 있다. U-Net은 대표적인 경량화 모델로, 인코더와 디코더로 이루어져 있으며 Skip connection을 통해 특징맵을 추출한다. 이러한 경량화된 신경망을 이용하여 도로경계 영역을 검출하는 연구가 진행된 바 있고 이를 통해 본 연구에서도 가벼운 모델 설계를 진행하였다.<sup>4)</sup> 단, 목표로 하는 차선과 도로경계가 전체 이미지에서 차지하는 영역이 적기 때문에 해당 특성을 고려하여 손실 함수를 설계하도록 하였다.<sup>5,6)</sup>

### 2.3 Multi-Task Learning

Multi-task learning은 하나의 딥러닝 모델로 다양한 태스크를 수행하는 구조로 실시간 동작에 장점을 가지고 있다. Mask R-CNN은 기존 객체 검출 딥러닝 모델인 Fast R-CNN에 객체 분할 태스크를 추가하여 객체 바운딩 박스의 클래스, 위치뿐만 아니라 Semantic segmentation을 수행하였다.<sup>7)</sup> MultiNet과 DLT-Net은 하나의 인코더와 3개의 디코더로 구성되어, 인코더를 통해 생성된 정보를 각각의 디코더에서 공유하게 하였다. 이를 통해 객체 검출, 객체 분류, 운전 가능 영역 검출 총 세가지 태스크를 동시에 수행하였다.<sup>8,9)</sup>

### 2.4 Object Tracking

객체 추적은 처음 인지한 객체에 대해 고유한 ID를 생성한 이후 다음 이미지 프레임에서 해당 객체에 대해 동일한 ID를 할당해 유지하는 작업이다. 자율 주행에서 객체를 지속적으로 감지하고 추적하는 것은 주변 환경 정보 활용에 도움을 주어 주행에 안정성을 더해준다. 이에 대표적으로 사용되는 방법은 Kalman filter로, 이전 상태를 통해 다음 상태를 예측하고 실제 측정값과의 차이를 통해 재귀적으로 상태를 보정하는 트랙 유지 기법이다. 본 연구에서 활용한 ByteTrack은 객체 검출 결과로 생성된 객체 바운딩 박스에 대해 Kalman filter를 적용하는 기법으로, 일부 가려진 객체에 대해서도 지속적으로 추적이 가능하게 하는 장점을 가진다.<sup>10)</sup>

## 3. 멀티 태스킹 러닝

본 논문에서 멀티태스킹 신경망은 1개의 통합 Backbone과 3개의 Head로 구성되어 있다. 통합 Backbone을 통해 생성된 이미지 특징 정보는 각 Head에서 공유되어 객체, 차선, 도로경계 검출 태스크를 동시에 다룬다. 해당 태스크는 상호 연관 관계가 뚜렷하고 함께 학습하였을 때 전체적인 성능 향상이 가능할 수 있는 조합으로 구성하였다.<sup>11)</sup>

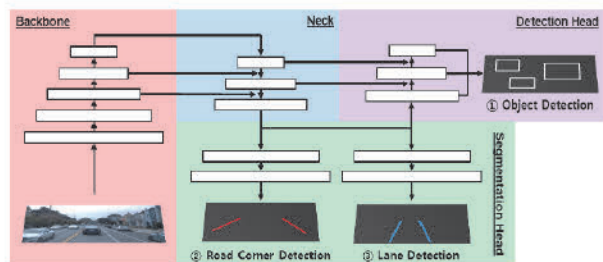


Fig. 1 Multi-tasking neural network configuration diagram

### 3.1 Backbone

딥러닝 신경망에서 Backbone은 인풋 데이터의 특성을 추출하는 과정으로, 복잡한 문제일수록 신경망 레이어를 깊게 쌓은 신경망 구조를 가지게 된다. 각 레이어마다 데이터를 표현하는 정보를 가지고 있으므로 일반적으로 많은 레이어를 가질수록 높은 성능을 나타낸다. 하지만 기존 컨볼루션 신경망은 레이어가 너무 깊어질 경우, 역전파 알고리즘을 통해 학습한 기울기 값이 소실되거나 발산하여 성능 하락이 발생할 수 있다. ResNet은 이러한 기울기 소실 문제를 해결하기 위해 컨볼루션 레이어를 거친 출력값에 입력값을 더하는 Single residual block을 선보였다. 본 연구는 통합 Backbone을 다음과 같은 ResNet으로 구성하여 멀티태스킹에 적합한 복합적인 Feature map을 추출하였다.

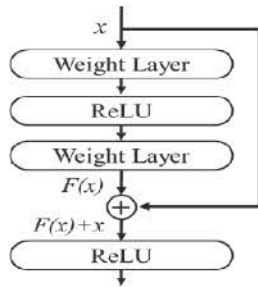


Fig. 2 Single residual block

### 3.2 Neck

FPN(Feature Pyramid Network)은 컨볼루션 레이어를 거쳐 다양한 해상도를 가진 피쳐맵을 추출하는 모델이다.<sup>12)</sup> Fig. 3은 FPN의 데이터 연산 흐름을 나타낸 것이다. 이러한 컨볼루션 레이어 연산과정은 상향 과정과 하향 과정으로 구성되어 있다. 상향 과정에서는 피쳐맵의 크기를 줄여가며 특정 레이어의 피쳐맵을 저장하고, 하향 과정에서는 피쳐맵의 크기를 다시 늘려가며 상향 과정에서 저장해둔 피쳐맵을 합치는 과정을 거친다. 추출한 피쳐맵의 크기가 작을 수록 이미지의 질감과 같은 추상적인 특성 정보를 가지고 있기 때문에, 서로 다른 피쳐맵을 합치는 과정을 통해 객체의 여러 특징들을 탐지할 수 있다.

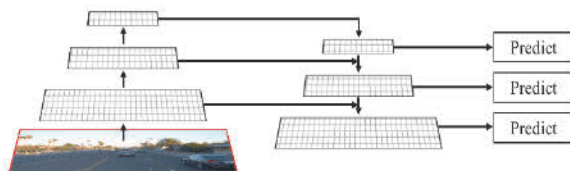


Fig. 3 Feature Pyramid network

### 3.3 Head

#### 3.3.1 차선 및 도로경계 인식 Head

차선 및 도로 경계 인식 Head는 이미지 데이터에 픽셀 라벨링을 수행하는 Semantic segmentation으로 구성하였다. Semantic segmentation은 동일한 객체 클래스를 나타내는 픽셀을 클러스터링하여 나타내는 작업이다. 본 연구에서는 FPN neck을 거쳐 생성된 최종 피쳐맵을 인풋값으로 사용하였고, 각각의 헤드에서 Cross entropy loss를 사용한 이진분류를 수행하여 차선과 도로경계를 나타내었다.

#### 3.3.2 객체 인식 Head

객체 인식 Head는 앵커 박스(Anchor box)기반의 검출을 수행하였다. 앵커 박스 기반 객체 인식은 다양한 크기의 객체가 존재하는 도로 환경에서 장점을 가지고, 1stage 방식을 따르기 때문에 빠른 속도로 추론이 가능하다. 본 연구에서 객체 인식 Head는 Yolo의 객체 탐색 기법을 참고하여 구성하였다. 인풋 데이터로는 차선과 도로경계 Head에서 사용한 Backbone과 FPN 피쳐맵을 공유하였고, 최종적으로 PAN(Path Aggregation Network) 구조를 통해 다양한 스케일의 정보를 융합한 피쳐맵 데이터가 사용된다.<sup>13)</sup> 이러한 과정을 통해 크기가 작은 객체부터 큰 객체까지 다양한 객체를 검출할 수 있다.

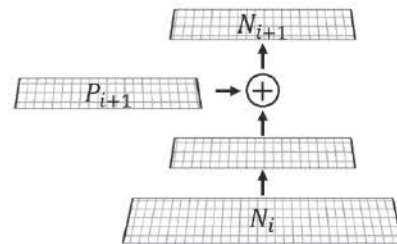


Fig. 4 Path aggregation network

### 3.4 멀티 태스킹 손실 함수

멀티태스킹 신경망은 태스크별로 생성된 손실 함수로 구성되어 있고, 각 태스크별 손실함수에 가중치를 부여하여 계산된 총합이 감소하는 방향으로 구성하였다. 차량 검출 Head의 손실 함수는 식 (1)로 표현할 수 있으며, 클래스 분류 손실,  $L_{cls}$ , 객체 분류 손실,  $L_{obj}$ , 바운딩 박스 손실,  $L_{box}$ 에 개별 가중치를 곱한 총합으로 구성된다.

$$L_{car} = \alpha_{cls}L_{cls} + \alpha_{obj}L_{obj} + \alpha_{box}L_{box} \quad (1)$$

차선과 도로 경계의 손실은 Cross entropy loss로 구성하였다. 이는 예측과 레이블 사이의 픽셀 수준 오차를 계산하여 최소화하는 방법이지만, 차선과 도로경계는 전체 이미지 대비 작은 영역에 존재하므로 부정 편향 손실 문제가 발생한다. 이에 Intersection Over Union(IOU)를 이용한 손실,  $L_{IOU}$ 를 (2)와 같이 구성하여 해당 문제를 해결하였다. 여기서  $L_{CE}$  Cross-entropy 손실 함수를 의미한다. 얇은 차선과 도로경계를 검출용 손실 함수,  $L_{lane,corner}$ 는 (3)과 같이 정의할 수 있다.

$$L_{IOU} = 1 - \frac{TP}{TP + FP + FN} \quad (2)$$

$$L_{lane,corner} = L_{CE} + L_{IOU} \quad (3)$$

최종 손실함수(4)는 다음과 같이 차량, 차선, 도로경계에 대한 손실함수 값에 개별 가중치를 곱한 총합으로 구성된다. 멀티태스킹 신경망의 손실함수는 6개의 파라미터가 존재하며, 하나의 태스크에서 사용한 파라미터가 다른 태스크의 학습에도 영향을 끼치므로 학습 경향을 보며 파라미터를 조절하는 기술이 필요하다.

$$L_{all} = \gamma_{car}L_{car} + \gamma_{lane}L_{lane} + \gamma_{corner}L_{corner} \quad (4)$$

#### 4. 객체 추적

도심지 환경에서 자차 주변 차량은 다양하게 존재하며, 해당 차량 정보를 안정적으로 이용하기 위해서는 끊임없는 인식이 필요하다. 따라서 본 연구에서 객체 인식 결과에 추적을 부여하여, 다양한 상황에서도 차량에 대

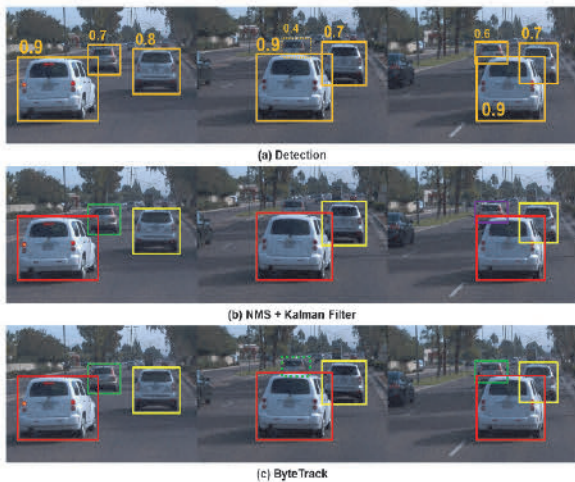


Fig. 5 Comparison of ByteTrack results with NMS tracking

한 인식을 유지할 수 있도록 하였다. 객체 추적은 인식 결과로 생성된 모든 객체 바운딩 박스에 대해 이루어지는 다중 객체 추적으로 이루어지고, 보정된 바운딩 박스와 ID를 추적 결과로 취득하였다.

Fig. 5에서는 인식결과에 NMS(Non-Maximum Suppression)와 Kalman Filter를 부여하는 기존 방식과 본 연구에서 ByteTrack를 적용한 방식을 비교하였다.<sup>14)</sup> 객체 인식의 결과로 각 바운딩 박스에는 Confidence score가 생성되고, 임계값 이상의 Confidence score를 가진 바운딩 박스만을 객체 인식의 결과로 사용하게 된다. 기존 Kalman filter 추적에서는 이렇게 필터링 된 정보로 선별된 객체에 대해서만 추적을 시행하게 된다. 따라서 객체가 가려져 인식단계에서 누락이 발생할 경우 동일 객체에 대한 ID유지에 어려움이 있을 수 있다.

본 연구에서는 Byte track을 이용하여 객체 인식 이후 결과를 활용하는 것이 아닌 NMS 적용단계에서 추적 알고리즘을 적용하였다. NMS 단계에서 알고리즘을 적용할 경우, 필터링 되지 않은 낮은 Confidence score의 바운딩 박스도 추적 후보로 가지게 된다. 따라서 Confidence score가 높은 객체에 대해서 이전 프레임과 비교하여 데이터 결합을 실시하고, 이전 프레임에 존재했지만 현 프레임에서는 Confidence score가 낮아 매칭이 되지 않은 바운딩 박스에 대해 한번 더 데이터 결합을 실행해준다. 그 결과, Fig. 5(a)의 객체 인식 결과를 바탕으로 비교해보면 Fig. 5(b)의 기존의 칼만필터 방식에서는 첫번째 이미지에서 초록색으로 그려졌던 바운딩 박스가 앞차에 의해 가려진 이후 기존에 부여된 아이디를 잃어버려 세번째 이미지에서 새로운 보라색 박스로 인식된다. 하지만 Fig. 5(c)의 ByteTrack의 경우 초록색 바운딩 박스로 표시된 객체가 앞차에 의해 가려지더라도 아이디가 유지되어 더욱 강건한 추적 성능을 보여줄 수 있다. 따라서 많은 차량이 존재하는 도로 환경에서도 더욱 안정적으로 객체 추적이 가능하게 한다.

#### 5. 알고리즘 적용 결과

Fig. 6(a), (b)는 멀티 태스킹 딥러닝 결과를 나타낸 이미지로 차선과 도로경계는 각각 빨강, 초록 픽셀로 가시화하였고 객체는 바운딩 박스로 표현하였다. Fig. 6(a)는 전방 도로에 객체가 다수 존재하는 경우이고 Fig. 6(b)는 차선과 도로 경계가 잘 드러나는 경우이다. 두 경우를 통해 객체와 차선 및 도로경계가 모두 잘 나타나는 것을 확인할 수 있다.

Fig. 7은 멀티태스킹 신경망에서 추론된 인식결과에 ByteTrack 후처리를 통해 추적을 적용한 결과 이미지를 (a), (b), (c) 시간순서대로 나타내었다. 객체 인식 결과에



(a)



(a)



(b)



(b)

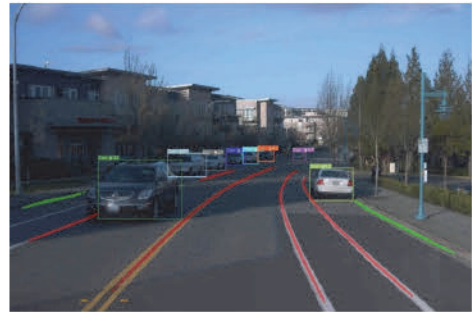
Fig. 6 Multi tasking deeplearning result image

Table 1 Performance metrics for image object recognition

	Recall (%)	AP (%)	FPS
MultiNet	81.3	60	8.6
DLT-Net	88.7	62.7	9.5
<b>Ours</b>	<b>57</b>	<b>71</b>	<b>24</b>

추적을 부여하여, 다양한 상황에서도 차량에 대한 인식을 유지할 수 있도록 하였다. 객체 추적은 인식 결과로 생성된 모든 객체 바운딩 박스에 대해 이루어지는 다중 객체 추적으로 이루어지고, 보정된 바운딩 박스와 ID를 추적 결과로 취득하였다. 추적이 적용된 결과 이미지 상에 나타난 동일 객체에 대한 바운딩 박스의 색과 ID가 동일하게 유지됨을 확인할 수 있다.

Table 1은 다른 멀티태스킹 모델과 함께 나타낸 이미지 객체 인식의 성능지표이다. 본 지표는 GeForce RTX 3080 Ti PC 환경에서 도출해낸 결과이다. 인식 결과, Average Precision(AP)은 71%이었고, 가장 안정적인 성능을 보였던 바운딩 박스 신뢰도 임계값 0.7에서 Recall은 57%, FPS는 24의 높은 성능을 보였다. Recall이 낮음에도 불구하고 AP가 높은 이유는 다소 작은 객체를 인식



(c)

Fig. 7 ByteTrack result image

하는데 어려움이 있지만 인식한 객체에 대해서는 높은 정확도를 나타내기 때문이다.

## 6. 결론

본 연구에서는 멀티 태스킹 딥러닝 모델을 이용하여 차선, 도로경계, 객체와 같은 도로 환경 인식과 객체 추적에 대한 연구를 수행하였다. 멀티 태스킹 모델을 통해 하나의 신경망에서 Backbone을 공유하여 여러 태스크를 동시에 수행하였고, 메모리와 연산속도를 감소시킬 수 있었다. 추가로 인식 객체에 ByteTrack 추적 기법을 적용

하여 동일 객체에 동일한 ID를 부여하였고 자율 주행 환경에서 더욱 강건한 인식 성능을 유지할 수 있었다. 해당 연구는 Waymo open dataset 중 인식대상이 모두 존재하는 10개의 시나리오에서 검증을 진행하였다. 그 결과 GeForce RTX 3080 Ti PC 환경에서 Recall 57%, AP 71%의 객체인식과 24 FPS 추론속도를 확인하였다. 유사한 연구인 MultiNet과 DLT-Net과 비교해보았을 때 높은 AP와 FPS 성능을 확인할 수 있었지만 Recall 값이 낮음을 확인하였다. 이는 Waymo open dataset 차선, 도로경계 추가 레이블링 과정에서 정확한 규격을 적용하여 학습을 진행할 경우 개선될 것으로 생각된다. 이러한 점을 개선하여 향후 인식 대상의 범위를 확장시켜 연구를 진행할 계획이다.

## 후 기

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2021R1A2C2003254). 또한 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임(P0020536, 2023년 산업혁신인재성장 지원사업).

## References

- 1) S. Kim, Y. Kim, H. Jeon, D. Kum and K. Lee, "Autonomous Driving Technology Trend and Future Outlook: Powered by Artificial Intelligence," Transactions of KSAE, Vol.30, No.10, pp.819-830, 2022.
- 2) S. Hwang, J. Choi, H. Park and J. Kim, "A Study on the Implementation of Camera Sensor for Object Detection Based on Deep Learning and Distance Estimation," Transactions of KSAE, Vol.30, No.8, pp.659-665, 2022.
- 3) A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "Yolov4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
- 4) T. Suleymanov, P. Amayo and P. Newman, "Inferring Road Boundaries Through and Despite Traffic," 21st International Conference on Intelligent Transportation Systems(ITSC), pp.409-416, 2018.
- 5) P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou and G. Cottrell, "Understanding Convolution for Semantic Segmentation," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.1451-1460, 2018.
- 6) S. Jadon, "A Survey of Loss Functions for Semantic Segmentation," IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology(CIBCB), pp.1-7, 2020.
- 7) K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-cnn," Proceedings of the IEEE International Conference on Computer Vision, pp.2961-2969, 2017.
- 8) M. Teichmann, M. Weber, M. Zoellner, R. Cipolla and R. Urtasun, "Multinet: Real-time Joint Semantic Reasoning for Autonomous Driving," 2018 IEEE Intelligent Vehicles Symposium(IV), pp.1013-1020, 2018.
- 9) Y. Qian, J. M. Dolan and M. Yang, "DLT-Net: Joint Detection of Drivable Areas, Lane Lines, and Traffic Objects," IEEE Transactions on Intelligent Transportation Systems, Vol.21, No.11, pp.4670-4679, 2020.
- 10) Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu and X. Wang, "Bytetrack: Multi-object Tracking by Associating Every Detection Box," European Conference on Computer Vision, 2022.
- 11) T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik and S. Savarese, "Which Tasks Should be Learned Together in Multi-task Learning?," International Conference on Machine Learning, pp.9120-9132, 2020.
- 12) T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2117-2125, 2017.
- 13) S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.8759-8768, 2018.
- 14) A. Neubeck, and L. Van Gool, "Efficient Non-Maximum Suppression," 18th International Conference on Pattern Recognition (ICPR'06), pp.850-855, 2006.