# **단안 카메라의 객체 검출 및 거리 추정 기반 자율주행 보조 시스템**

San Kim

Aiffel, modulabs

**Abstract**

최근에 자율주행 기술이 급속히 발전함에 따라 다양한 관련 기술들이 발전되며 적용되고 있다. 자율주행 기술로 운전자의 편리성, 안전성을 제고한다. 이를 위해서는 자동차와 다른 객체를 구분하고 그 사이의 거리를 측정하여 주행 여부를 결정해야 한다. 객체 탐지 및 거리 측정을 위해 다양한 데이터를 활용할 수 있다. 최근에는 라이다 센서와 같은 다양한 센서로 데이터를 구성하고 입력으로 받는다. 하지만 이는 큰 비용을 야기한다. 이를 해결하기 위해 이미지 데이터만으로 객체 탐지 모델을 만들고 해당 객체와의 거리를 측정하여 주행 여부를 결정하는 알고리즘을 구상했다. 객체 탐지 모델은 KITTI dataset을 사용하여 학습되었고, 거리 측정에는 단안 카메라로 물체와의 거리를 추정하는 방법을 적용하였다. 이를 통해 객체 탐지를 통한 거리 기반의 자율주행 보조 시스템을 구현하고자 한다.

# **Introduction**

자율주행에 대한 기술이 발전함에 따라 소비자들의 자율주행에 대한 관심 및 수요가 폭발적으로 증가하고 있다. 다양한 글로벌 자동차 제조업 회사가 운전자의 편의를 위한 자율주행 서비스 혹은 보조 시스템을 구축하여 제공한다. 이는 매우 복잡하여 실제 완전한 서비스로 구현되기 위해서는 센서(IoT), 통신(Mobile, Network), 빅데이터, 인공지능 기술이 모두 상호 융합되어야 한다.

최근 자율주행은 첨단 운전자 지원 시스템인 ADAS(advanced driver assistance system)의 형태로 실현되어지고 있다. ADAS는 차량에 장착된 각종 센서와 카메라에서 외부 환경 정보를 감지하고 이를 통해 운전자에게 적절한 조치를 취하도록 알려주거나, 차량 스스로 주행제어를 수행한다. 이 때 외부 환경 정보 및 주행 환경에 대한 데이터를 위해 다양한 장치가 필요하다는 단점이 있다. 이는 비교적 큰 설치 및 유지, 보수 비용을 야기한다. 이를 해결하기 위해 카메라의 이미지 데이터만을 활용한 자율주행 보조 시스템이 제안되었다. 기존의 자율주행 보조 시스템(Going deeper node)은 주행 제어에 대한 조건 상황을 단편적이고 고정적으로 가정했다. 실제 주행 상황에서는 매우 비효율적이며, 운전자와 보행자의 안전에 위험이 될 수 있다는 문제가 있다.

따라서 유연한 주행 제어를 위해 거리 기반의 자율주행 보조 시스템을 구현하고자 한다. 우선, 객체 탐지를 위해 KITTI datasets과 [1]RetinaNet 모델을 활용하여 딥러닝 모델을 구현했다. 이후 탐지된 객체와 사용자 사이의 거리를 단안 카메라를 사용한 이미지와 차량의 속력을 바탕으로 측정한다. 해당 거리를 기준으로 주행 제어를 수행하는 자율주행 보조 시스템을 제안한다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

그림1. 자율주행 시스템과 기반 기술

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

그림2. 자율주행을 위한 다양한 센서 예시

## **2. Related works**

Aiffel going deeper node에서 딥러닝 기반의 객체 탐지 모델을 활용하여 자율주행 보조 시스템을 제안하였다. 객체 탐지를 위해 [1]RetinaNet 기반의 object detection 딥러닝 모델을 구현하였으며, 모델에 의해 탐지된 객체는 두 가지의 상황에 대한 조건을 판단하기 위한 기준이 된다. 해당 자율주행 보조 시스템은 주행 제어에 대하여 다음과 같은 조건을 가정한다. 1. 사람이 한 명 이상 있는 경우, 2. 차량의 픽셀 크기가 300px이상인 경우.

해당 조건은 카메라에 사람이 탐지 되었을 때, 그리고 차가 가까워져서 탐지된 크기가 일정 크기 이상일 때를 판단할 때를 가정한다. 하지만 해당 요구사항은 자율주행 시스템에 부적절한 경우가 발생할 수 있다. 예를 들어, 사람이 이미지에 나타나면 자동차가 주행을 무조건 멈추게 되며, 소형 차량 및 오토바이는 가까이 있더라도 주행 제어가 되지 않는 경우가 생길 수 있다.

# **Method**

단안 카메라의 객체 검출 및 거리 측정 기반 자율주행 보조 시스템에는 다음 두 가지의 과정이 필요하다. 하나는 객체 검출의 과정이며, 그 다음으로 검출된 객체와 사용자 사이의 거리를 구하는 과정이다. 이 과정을 end to end로 구현하기 위해 object detection model을 구현하였고 해당 모델이 반환하는 값을 기준으로 bbox의 pixel좌표를 사용하여 거리 측정 알고리즘을 사용하여 각 class 마다의 고유한 거리 값을 설정하여 해당 값 이하의 거리가 측정되면 차가 멈추고, 이상의 거리가 측정되면 차가 멈추지 않도록 하는 주행보조 시스템을 구현한다.

Object detection 모델을 학습하기 위한 데이터로 KITTI datasets을 사용하였다. 해당 데이터 셋은 자율주행을 위한 데이터 셋으로 2D / 3D object detection 라벨을 제공한다는 점이 특징이다. 데이터셋을 다운 받아 모델의 학습에 사용할 때, tensorflow\_datasets(tfds)에서 제공하는 API를 사용하여 다운로드 하였다. 해당 데이터 셋에서 type, bbox 라벨을 사용하였다. type 라벨에는 다음과 같은 탐지될 객체의 정보를 제공한다. ‘car’, ’van’, ’truck’, ‘pedestrian’, ‘person\_sitting’, ‘cyclist’, tram’, ‘misc’가 type 라벨에 포함된다. 또한 bbox 라벨에는 2D bounding box의 정보를 제공한다. Bbox에는 left, top, right, bottom pixel coordinates의 정보가 들어있다. 해당 데이터를 바탕으로 객체에 bbox를 그려냄으로써 object detection이 수행된다.

Object detecion을 위한 모델은 ‘Focal Loss for Dense Object Detection’에서 제안된 RetinaNet 모델 구조를 사용하였다. 1 -stage detector 모델의 낮은 성능 문제를 해결하기 위해 focal loss와 FPN(feature pyramid network)를 적용한 네트워크이다. 먼저 focal loss는 물체 전경과 배경을 담고 있는 모든 그리드에 대해 한 번에 학습됨으로 인해 생기는 클래스 간의 불균형을 해결하고자 도입되었다.

스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

그림3. Corresponding regions of each pixel in feature map

그림 3을 보면 7 \* 7 feature map에서 각 pixel이 원본 224 \* 224이미지에서의 한 grid의 receptive field를 가지게 된다. Feature map의 각 픽셀은 배경인 부분이 많으므로 imbalanced class data이다. 따라서 모든 grid에 대해 학습될 때, 교차 엔트로피에 modulating factor(1 - pt)를 붙여 imbalance한 배경 gird에 대한 영향을 줄이도록 손실을 조절한다.

텍스트, 라인, 스크린샷, 도표이(가) 표시된 사진

자동 생성된 설명

그림4. Focal loss

γ를 0으로 설정하면 modulating factor가 1이 되어 일반적인 교차 엔트로피가 되며 그림 4에 따라, γ가 커질수록 modulating이 강하게 적용된다.

FPN은 convolution 연산을 통해 추출된 feature을 피라미드처럼 쌓아 사용하는 방식을 사용한다. 앞쪽 layer의 Feature map은 뒤쪽 layer에 비해 좁은 범위의 receptive field를 가지며, 넓은 범위의 feature과 이를 합쳐 보다 다양한 feature를 고려함을 시사했다.

스크린샷, 디자인, 라인이(가) 표시된 사진

자동 생성된 설명스크린샷, 라인, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

그림5. Single feature map vs FPN

그림 5를 보면, single feature map은 기존의 CNN backbone network를 통과한 feature map을 그대로 predict에 사용된 반면, FPN은 backbone의 여러 layer를 통과한 feature map들을 upsampling하여 뒤쪽 feature map과 더하여 사용한다. Backbone network는 ResNet을 사용하였으며, 전체적인 RetinaNet의 FPN 구조는 다음 그림6과 같다.

도표, 라인, 평면도, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그림6. RetinaNet FPN 구조

거리 측정을 위해 카메라의 이미지 센서에서 포착된 객체의 크기를 이용하여 렌즈와 객체사이의 거리를 구하는 식을 적용하였다. 모델이 추론한 객체의 bbox 좌표를 사용한다.

도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그림7. 렌즈와 객체 사이의 거리 측정 단위

그림 7의 단위를 사용하여 거리를 구하기 위한 수식은 다음과 같다.



위 수식 (1)과 (2)를 이용하여 이미지 상의 물체의 크기 비율을 통해 카메라와 물체와의 거리를 구할 수 있다.

따라서 식(4)의 d를 도출하기 위해 두 개의 이미지의 좌표 상의 같은 물체의 특정 길이 a, b와 움직인 거리(카메라 or 객체)를 알아야 한다. 물체의 위치를 object detection 모델을 통해 bounding box를 구하고 tracking하여 이전, 이후 frame 간 박스 크기를 비교하여 거리를 추정하는 방식을 사용한다. 이 때 사용되는 이동 거리는 자동차의 속력을 시간 단위로 곱하여 추정하고 이를 카메라의 이동 거리로 적용한다. 이를 통해 사용자와 객체 거리를 측정하고 각 기준에 맞춰 에 사용한다.

## **4. Conclusion**

기존의 단안 카메라의 자율주행 보조 시스템을 보완하기 위해 거리 측정 알고리즘을 추가했다. 기존의 방식과 동일한 것은 객체 탐지를 위한 딥러닝 모델을 사용했다는 것이다. KITTI datasets으로 RetinaNet 모델을 학습했다. 기존의 자율 주행 보조 시스템을 통해 object detection이 잘 수행되는 것을 확인했다. 이후 사람 class가 등장하는 지, bbox의 pixel의 크기가 300이상인지를 파악하여 암묵적인 거리의 개념을 사용하여 주행 제어를 수행하도록 설계했다.

야외, 도로, 차량, 장면이(가) 표시된 사진

자동 생성된 설명차량, 육상 차량, 야외, 나무이(가) 표시된 사진

자동 생성된 설명

그림8. 기존 자율주행 보조 시스템의 추론 결과

이를 통해 거리 측정에 필요한 정보를 활용하기에 이미 충분한 것으로 파악되었다. 비용이 크지 않은 단안 카메라의 이미지 정보를 사용하여 비교적 세밀하고 효율적인 주행 제어가 필요했다. 따라서 해당 시스템을 바탕으로 거리 추정 개념을 도입하여 자율주행 보조 시스템에 사용하고자 했다.

## **5. Discussion**

제안된 자율주행 보조 시스템에 대한 결과를 파악할 수 없다. 따라서 이를 실제로 구현하고 결과를 확인하는 것이 필요하다. 뿐만 아니라 구현되더라도 정량적 평가에 대한 기준을 마련해야 한다. 기존의 시스템에서 객체 탐지에 대한 metric은 IoU이다. 하지만 자율어떤 기준이 자율주행 보조 시스템이 운전자와 보행자에게 안전성과 편리한 주행 경험을 제공할지에 대해 고민할 필요가 있다.

거리 측정 방식에 대한 class별 기준을 차별화 하여 속력에 대한 입력이 없어도 거리 추정을 할 수 있어야 한다. 따라서 각 class별 기준이 되는 가중치 값을 실험적으로 파악하여 지정하는 과정이 필요하다. 해당 가중치는 거리 측정의 값을 보정하는 역할을 할 것으로 기대된다.

# **References**

1. Lin, Tsung-Yi, et al. "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision*. 2017.