

Object Detection 자율주행 보조 시스템

김태민

요 약

최근에는 자율주행 기술이 급속히 발전하고 있으며, 이에 따라 자동차는 도로를 식별하고 도로 경로를 따라 안전하게 주행할 수 있게 되었다. 그러나 아직 완벽하지 않은 자율주행 시스템은 위험한 상황에서는 운전자의 개입이 필요하며, 판단이 어려운 상황에서는 차량을 멈추어야 한다. 이를 해결하기 위해, 데이터셋을 활용하여 전처리를 수행하고, RetinaNet 모델을 구축하기 위해 ResNet50을 백본으로 사용했다. 또한, RetinaNetBoxLoss를 손실 함수로 도입하였고, IoU를 통해 실제 라벨과 Anchor box 간의 일치 여부를 평가했다. 특정 위치에 도달하면 주행을 멈추도록 하는 정지 조건을 구현함으로써, 자율주행 시스템을 실제로 구현하였다.

1. 서 론

현대 자동차 산업은 자율주행 기술의 급격한 발전으로 새로운 전환점을 맞이하고 있다. 최근에는 자동차가 도로를 식별하고 안전한 경로를 따라 주행하는 능력이 상당히 향상되었다. 그러나 이러한 기술은 아직 완벽하지 않아, 특히 위험한 상황이나 판단이 어려운 상황에서는 운전자의 개입이 필수적이다. 이를 해결하기 위해 KITTI 데이터셋을 활용하여 철저한 전처리를 수행하고, ResNet50을 백본으로 활용하여 RetinaNet[1] 모델을 구축했다. 이 모델에서는 RetinaNetBoxLoss를 손실 함수로 도입하여 object detection 정확도를 높였으며, IoU를 통해 모델이 실제 라벨과 얼마나 일치하는지를 평가하였다. 더불어, 주행 중 특정 위치에 도달하면 주행을 안전하게 멈추도록 하는 정지 조건을 구현함으로써, 현실적이고 안정적인 자율주행 시스템을 실제로 구현하였다.

본 논문은 자율주행 분야에서의 기술적 도약을 제시하며, 안전성과 신뢰성을 고려한 자율주행 시스템의 핵심 구성 요소에 대한 탐구를 목표로 합니다.

2. 관련 연구

Lin et al. [1]은 detection 모델인 RetinaNet 모델을 공개했다. 1-stage detector 모델이 2-stage detector 모델보다 속도는 빠르지만 성능이 낮은 문제를 가지고 있다. 이를 해결하기 위해 focal loss와 FPN(Feature Pyramid Network)를 적용한 모델을 사용한다.

3. 제안 모델

그림 1은 Object detection 모델이 object를 식별하는 본 논문의 전반적인 도식도이다.

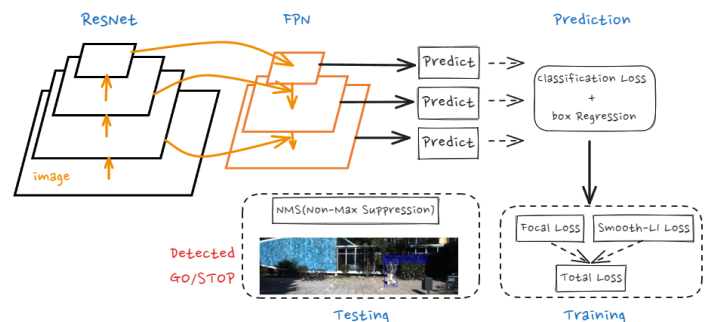


그림 1. 제안 방법의 도식도.

본 논문에서는 RetinaNet 모델의 backbone으로 ResNet 기반의 FPN을 활용한다. 이후, FPN을 통해 생성된 피라미드 레벨별 feature map을 활용하여 object detection을 수행한다. 훈련 단계에서는 Focal Loss와 Smooth-L1 Loss를 결합하여 Total Loss를 계산하고, 이를 통해 모델을 평가한다. 시험 단계에서는 Non-Maximum Suppression(NMS)을 활용하여 중복된 box를 제거하고, 최종 결과를 얻은 후, GO/STOP 여부를 예측한다.

3.1. Focal Loss

$$\begin{aligned} L &= -\log(p) \quad \xrightarrow{\text{반도수의 역수}} \quad L = -\alpha \cdot \log(p) \quad \xrightarrow{\text{Focusing Parameter}} \quad L = -(1-p)^{\gamma} \cdot \log(p) \\ &\text{cross Entropy Loss} \quad \text{Balanced cross Entropy Loss} \quad \text{Focal Loss} \end{aligned}$$

난이도에 비례하는 가중치

그림 2. Focal Loss 수식

그림 2는 기존 1-stage detection 모델에서 발생하는 클래스 간의 불균형 문제를 해결하기 위해 Focal Loss가 도입되었다. 이 모델에서는 물체 전경과 배경을 담고 있는 그리드에 대한 전체 학습이 한 번에 이루어지는데, 이로 인해 원래 물체의 bounding box에

대해서는 큰 손실이 발생하는 경우가 있다. 이러한 상황에서는 손실이 갑자기 작아지면서 밸런스가 무너지게 된다. Focal Loss는 Balanced Cross Entropy와 같이 빈도수의 역수로 가중치를 조절하지만, 학습 진행상황과 관계없이 고정된 빈도수를 사용하는 대신, 모델의 학습 상태나 문제의 난이도에 따라 가중치를 동적으로 조절한다. 즉, 쉬운 문제는 덜 학습하고, 어려운 문제에 높은 주의를 기울이도록 하기 위해 난이도에 비례하는 가중치를 Cross Entropy Loss에 적용한 것이 Focal Loss이다. 이를 통해 모델은 어려운 샘플에 더 집중하면서 불균형 문제를 효과적으로 극복할 수 있게 된다.

3.2. FPN (Feature Pyramid Network)

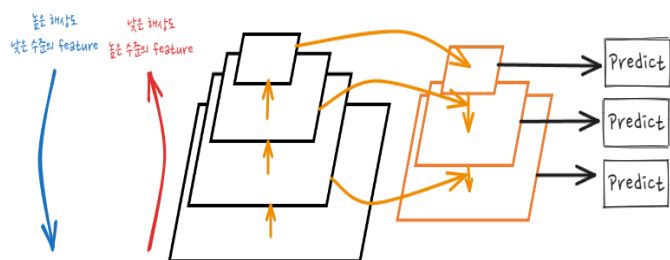


그림 3. FPN의 도식도.

기존의 Object detection 모델은 모델 크기와 무관하게 다양한 크기의 객체를 탐지할 수 있는 방법을 찾으려고 노력했다. 그러나 기존 방식을 사용할 경우 모델의 추론 속도가 느리고, 과도한 메모리 사용량이 발생하는 문제가 있다. 그림 3의 FPN은 이러한 문제를 해결하기 위해 제안된 네트워크로, 임의의 크기의 단일 스케일 이미지를 Conv에 입력하여 다양한 스케일의 특징 맵을 출력한다. FPN의 핵심 아이디어는 Bottom-Up 구조로 구성된 네트워크에서 각 레벨의 특징이 피라미드의 하위 레벨로 전달된다는 것이다. 그 이후 Top-Down방식을 사용하여 각 레벨의 특징 맵과 고해상도 정보를 결합함으로써, 객체의 다양한 크기와 디테일을 처리할 수 있는 피라미드를 형성한다. Lateral Connection은 각 레벨 간의 정보를 연결하는데, 이는 피라미드 내에서 저수준 특징과 고수준 특징 간의 시맨틱 정보를 공유하게 한다. 이를 통해 작은 객체와 큰 객체를 동시에 효과적으로 탐지할 수 있는 네트워크가 형성되며, Faster R-CNN과 같은 object detection 모델에서 주로 백본(backbone)으로 사용된다. FPN은 object detection 모델에서의 효과적인 다양한 크기의 object detection을 위한 핵심 구성 요소로 널리 채택되고 있다.

3.3. 전처리

본 논문에서 제안된 object detection 모델의 학습을 위해 다음과 같은 4단계의 전처리 파이프라인을

구성했다.

1. x와 y좌표 위치 교체:

먼저, 주어진 데이터셋에서 객체의 위치 정보를 효과적으로 활용하기 위해 x와 y 좌표의 위치를 교체한다. 일반적으로 이미지 처리에서는 (y, x) 좌표 체계를 사용하는데, 이를 모델 학습에 맞게 (x, y)로 교체함으로써 일관된 입력 형식을 제공한다.

2. Flip 하기:

데이터의 다양성을 확보하고 모델의 일반화 성능을 향상시키기 위해 이미지에 무작위로 수평 뒤집기를 수행한다. 이렇게 함으로써 모델은 뒤집힌 상황에서도 객체를 올바르게 인식할 수 있는 능력을 향상시킨다.

3. 이미지 크기 조정 및 패딩 추가:

모델의 입력 크기를 통일시키기 위해 이미지 크기를 조정하고, 필요에 따라 패딩을 추가한다. 이는 모델이 서로 다른 크기의 입력 이미지에서도 일관된 방식으로 객체를 감지하고 학습할 수 있도록 도와준다.

4. 좌표계 수정:

객체의 위치 정보를 표현하는 좌표계를 수정한다. 주로 사용되는 $[x_{min}, y_{min}, x_{max}, y_{max}]$ 형식의 좌표를 $[x_{min}, y_{min}, width, height]$ 형태로 변환하여 모델이 더 직관적으로 객체의 위치를 학습할 수 있도록 한다.

이러한 전처리 파이프라인을 통해 모델은 다양한 상황에서 일관된 입력 형식으로 학습되며, 데이터셋의 다양성과 일관성을 확보하여 object detection 성능을 향상시킬 수 있다.

3.4. RetinaNet 모델

본 논문에서 제안된 object detection을 위한 RetinaNet 모델은 강력한 컨볼루션 네트워크인 ResNet을 backbone으로 채택하고, 이를 기반으로 FPN을 통합한다. 이를 통해 다양한 크기와 비율의 객체에 대해 강력한 탐지 능력을 제공한다. 이후, 객체의 위치를 예측하기 위해 각 픽셀 위치에 다양한 크기와 비율의 Anchor를 생성한다. Classification Head는 객체의 클래스를 분류하며, Box Regression Head는 객체의 bounding box를 조정하는 역할을 담당한다. 그 후에는 Focal Loss를 도입하여 객체와 배경 클래스 간의 불균형을 효과적으로 해결한다. 이 모델은 다양한 객체에 대한 정확한 detection ability를 보여준다.

5. 실험 결과

본 논문에서 사용한 하이퍼파라미터는 다음과 같다. ‘learning_rates’는 learning_rate.schedules를 사용하여 설정하였으며, ‘batchsize = 2’, ‘epoch = 7’로 설정하였다.

5.1. 데이터 세트

본 논문에서 사용한 데이터 세트는 표 1와 같다.

표 1. 사용한 데이터 세트

	Train	Validation	Test
#Of Data	6,347	423	711
Classes	‘Car’, ‘Van’, ‘Truck’, ‘Pedestrian’, ‘Person_sitting’, ‘Cyclist’, ‘Tram’, ‘Misc’		

KITTI 데이터셋의 Train 데이터셋은 6,347개, Validation 데이터셋은 423개, Test 데이터셋은 711개이다. Object class 개수는 8개이다.

5.2. 평가 지표

본 논문에서는 중복 버그 예측 성능을 평가하기 위해 다음의 성능 평가 지표 [31, 32]를 사용한다.

$$IoU(A, B) = \frac{\text{면적}(A \cap B)}{\text{면적}(A \cup B)} \quad (1)$$

$$\text{RetinaNetLoss} = \text{Classification Loss} + \text{Box Regression Loss} \quad (2)$$

$IoU(A, B)$ (1)은 두 bounding box의 겹친 영역의 크기를 두 bounding box 전체의 영역으로 나눈 비율을 나타내며, 일정 임계값 이상일 때 두 bounding box를 ‘일치’했다고 판단한다.

RetinaNetLoss (2)은 먼저 Classification Loss는 객체의 클래스를 분류하는 작업에서 사용되는 loss function로서, Cross-Entropy Loss를 변형하여 구현되었다. 그리고 Box Regression Loss는 객체의 bounding box를 예측하는 작업에서 사용되는 loss function로서, Smooth L1 Loss를 변형하여 구현되었다. RetinaNetLoss은 객체의 클래스 분류와 bounding box 예측을 모두 고려하여 모델을 훈련하며, 최적화 과정에서 이 손실을 최소화하는 방향으로 진행된다.

5.3. 실험 결과

그림 5은 본 논문의 제안한 모델의 Loss의 성능을 나타낸다.

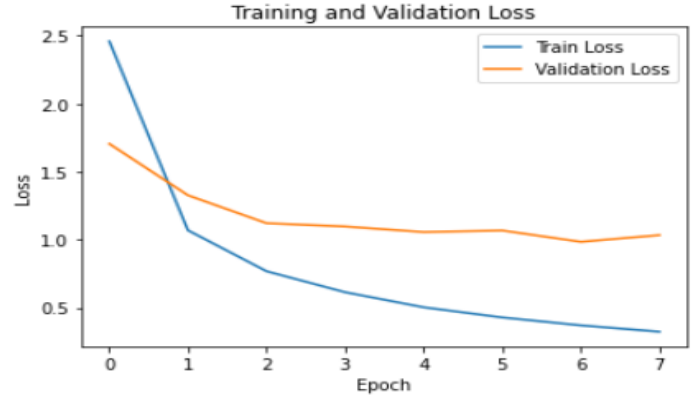


그림 5. 제안한 모델의 성능 결과.

X축은 Epoch 수를, Y축은 Loss로 했을 때, Train Loss는 빠른 속도로 수렴하는 것을 관찰할 수 있다. 이는 모델이 훈련 데이터에 잘 적응하고 있는 것을 나타낸다. 반면에, Validation Loss는 수렴 속도가 상대적으로 느리지만, 여전히 감소하고 있는 것으로 보인다. 이는 모델이 새로운 데이터에 대해서도 일반화되고 있으며 훈련 데이터에 지나치게 적응되지 않고 있는 것을 확인할 수 있다.

5.4. 추론 결과

그림 6번은 학습된 모델을 사용하여 추론한 결과를 나타낸다.



그림 6. 추론 결과.

Test 데이터셋에서 100개의 후보만 골라서 NMS를 활용하여 중복된 box를 제거하고 학습된 결과를 확인한 결과를 보이고 있다.

다음 그림 7은 정지 조건을 넣어 조건에 맞으면 “Stop” 아닌 경우 “Go”를 반환하는 자율주행 시스템이다.



그림 7. 자율주행 시스템 결과.

정지 조건은 사람이 한 명 이상 있는 경우, 차량의 크기(width or height)가 300px이상인 경우 ‘Stop’이 결과로 도출된다.

6. 토의

6.1. 위험요소

본 논문에서는 7,481개의 KITTI 데이터셋을 활용하여 실험을 진행했다. 그러나 해당 모델이 다른 데이터셋에 일반화되어 적용 가능한지 여부는 추가적인 검증이 필요하다. 본 논문에서 사용된 하이퍼파라미터는 기본 설정으로 수행되었으며, 항상 좋은 성능을 보장하는 것은 아니다. 따라서 향후에는 하이퍼파라미터 최적화 기법을 활용하여 모델을 최적화할 계획이다. 또한, 모델의 성능 평가 방법은 추가적인 모델 평가 척도를 도입하여 성능을 더욱 확실하게 검증할 계획이다.

7. 결론

최근 자율주행 기술의 급속한 발전 속에서, 운전자 개입이 필요한 위험 상황과 판단이 어려운 상황에 대한

대응이 중요한 과제로 부상하고 있다. 본 논문에서는 KITTI 데이터셋을 활용하여 RetinaNet 모델을 통한 안정적인 자율주행 시스템을 제안했다. 전처리 단계에서는 데이터셋의 신뢰성을 높이기 위한 과정을 수행했고, ResNet50을 backbone으로 채택하여 RetinaNet을 구축했다. 또한 RetinaNetBoxLoss를 통해 object detection의 정확도를 높였으며, IoU를 활용하여 모델의 일치 여부를 정량적으로 평가했다. 특히, 정지 조건을 도입하여 특정 조건이 발생되면 주행을 멈춰야한다는 신호를 보내는 시스템을 실현했다. 실험 결과에서는 Train Loss가 빠른 수렴을 보이고, Validation Loss는 상대적으로 느린 수렴 속도를 나타내면서도 계속해서 감소함을 확인할 수 있다. 이러한 결과는 모델이 훈련 데이터에 잘 적응하면서도 새로운 데이터에 대해서도 일반화되고 있다는 것을 알 수 있다. 물론, 본 논문에서 적용한 모델과 방법론이 다른 데이터셋에 대해서도 효과적으로 작동하는지 추가 검증이 필요하며, 하이퍼파라미터 최적화 등을 통한 모델 성능 향상에 대한 노력이 필요하다. 향후에는 더 다양한 모델 평가 척도를 적용하여 모델의 성능을 보다 심층적으로 평가할 계획이다.

참고 문헌

- [1] T. Y. Lin, P. Goyal, R. Girshick, K. He, & P. Dollár, "Focal loss for dense object detection.", In Proc. of the IEEE international conference on computer vision, pp. 2980-2988, 2017.