

2D 이미지에서 사람의 이륜차 탑승 여부 판정

김도형, 배재용, 이도윤, 함지웅
Seoul National University

shapespeare@snu.ac.kr, qowodyd0116@snu.ac.kr, Owoolee@snu.ac.kr, amoretspero@snu.ac.kr

Abstract

물체 인식 단계에서, 복합적 요소의 인식은 인간의 자연스러운 인지 작용 중 하나이다. 많은 시각 인식 네트워크는 복합적 요소를 분류된 개별적 인식의 단순 합으로 인식하고, 이는 특정 분야에서 치명적인 정보의 손실을 야기하곤 한다.

이번 프로젝트에서는 사람-이륜차 복합구조의 인식을 주요 목표로 한다. 시각 인식의 결과 *feature*로 물체의 3D *Orientation*를 추가하고 이 특성에 주목하여 복합적 요소의 공간적 관계를 확인하는 처리를 더해봄으로써 향상된 결과를 확인하는 것을 목표로 한다.

1. 프로젝트 소개

Object Detection 기술이 발달함에 따라 이를 실생활에 적용한 예시도 다양하게 증가하였다. 특히 차량의 경우 후방 카메라를 통해 후진이나 주정차 시 사람이나 이륜차 등의 장애물을 미리 구분하여 탐지할 수 있다. 다만, 기존의 Object Detection 기술은 화면에서 사람과 이륜차를 탐지 할 수는 있지만, 구체적으로 그것이 이륜차에 탑승한 사람인지, 아니면 원근감에 의해 실제로는 멀리 떨어져 있는 사람이 겹쳐져서 보이는 것인지 잘 구분하지 못한다. 화상과 실제 물체 사이의 거리를 정확하게 확인하기 어려운 상황에서, 이를 해결하기 위해 새로운 접근법을 제안하고자 한다.

1.1. 진행할 작업

사람이 이륜차에 탑승했는지 여부를 판단하기 위해 ‘이륜차에 탑승한 사람’이라는 레이블링된 이미지를 따로 학습시키는 방법이 아닌, 기존의 ‘사람’과 ‘이륜차’를 인식하는 방법에 대해 그 결과를 바탕으로 탑승 여부를 판단하는 방법을 사용할 것이다. Object Detection을 위해 CenterNet[8]을 사용할 것이며 프로젝트 상황에 맞게 모델에 변형을 가하고 따로 선정한 데이터셋으로 학습시킨 후, postprocessing 과정을 추가할 것이다.

이번 프로젝트에서 인식할 대상은 사람과 이륜차로 한정되기 때문에 통상적인 Object Detection에 비해 인식할 카테고리 수가 적어 모델을 변형할 필요가 있다. 여기서 이륜차는 오토바이, 자전거 등의 세부 카테고리로 나눌 수 있으며, 얻을 수 있는 dataset에 추가적인 카테고리들이 있다

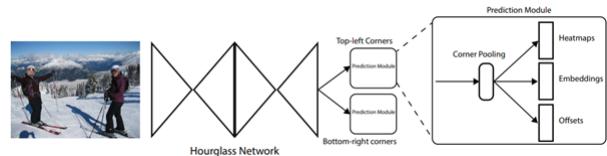


Figure 1. CornerNet의 구조

면 더 추가할 수도 있다. 또한 단순 boundary box detection이 아닌 3d orientation을 얻을 수 있도록 backbone 네트워크를 선정하고 CenterNet[8]에서도 이를 처리할 수 있게 해주어야 하며, 이후 작업을 위해 모델을 학습시키고 평가하는 모듈도 작성해야 한다. 학습용 데이터셋을 구하고 정제하는 과정도 필요하다. 이후의 절에 제시된 dataset들에서 필요한 이미지, 즉 사람과 이륜차가 포함된 이미지만을 추출하고 이들의 annotation에서도 불필요한 부분을 제거하고 형식을 통일하는 작업이 필요하다.

이렇게 얻은 정보들을 바탕으로 실제 탑승여부를 판단할 모듈의 작성이 필요하다. boundary box interpolation을 통해서 상하 관계에 있는 (사람, 이륜차) 쌍을 추려내고, IoU와 3d orientation 정보를 바탕으로 실제 탑승여부를 판단할 것이다. 최종적으로 신경망 모델을 내재해서 이미지를 받아서 이를 평가하는 모듈과 실제 탑승 여부를 평가하는 모듈을 연결하여 프로젝트가 완성된다.

2. 선행연구 조사

2.1. CornerNet

CornerNet[3]은 사진으로부터 객체의 bounding box를 결정하기 위해 box의 왼쪽 위, 오른쪽 아래, 두 keypoint를 예측하는 모델이다. Anchor box를 사용하는 기존의 모델과 달리 학습할 때 positive 값과 비교해 맞춰야 할 negative 결과의 가능성이 대폭 감소하고, hyperparameter의 수가 줄어들어 훨씬 빠른 학습을 진행할 수 있다.

CornerNet[3]은 Hourglass[6]을 backbone으로 삼은 신경망 모듈을 사용해 최종 결과를 좌상단/우하단 keypoint를 결정하는 각각의 모듈에 전달하고 각 모듈은 max-pooling을 사용해 keypoint를 확인할 수 있도록 한다. 이후 Ground-truth keypoint와의 비교를 통해 heatmap을 구하고 최종적으로 특징 점들을 한 쌍씩 묶기 위해 Associative Embedding[5]을 사용해 결과 쌍을 도출한다.

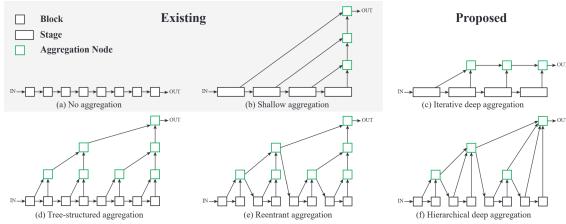


Figure 2. 기존에 존재하는 Aggregation model (b) 와 이를 발전시킨 (c)~(f) 구조. 해당 논문에서는 (c) 와 (f) 구조의 사용을 통해 성능 증대를 확인하였다. Block은 layer의 단순 그룹 규모의 구조이며, 동일한 feature resolution의 block을 묶어 stage라는 큰 규모의 구조를 형성한다.

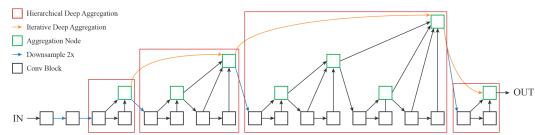


Figure 3. 분류를 위한 DLA[7] 구조.

그러나 CornerNet[3]은 두 쌍의 keypoint만을 활용하여 근접한 물체들을 탐지하는 데에 정확도가 감소하고 bounding box의 크기만을 결정하다 보니 부정확한 bounding box 결과를 다수 출력하는 문제점을 보일 때가 있다.

2.2. DLA-34

DLA[7]는 이전에 존재하던 Aggregation method를 발전시킨 구조이다. 기존에 존재하는 feature aggregation을 공간적 및 의미론적으로 다시 고찰하여, Skip connection architecture 관점에서 새로운 aggregation method인 IDA, HDA를 제안하였다.

IDA는 Feature pyramid networks (FPNs)를 발전시킨 공간적 파악을 위한 aggregation 구조이다. feature resolution의 규모인 stage에 적용한다. 선형이며 level에 따라 급진적인 과정인 FPNs를 개선시켜, 비 선형적이고 점진적인 과정을 도입하였다.(Fig 2-(c)) HDA는 Densely connected networks (DenseNets)를 발전시킨 의미론적 파악을 위한 aggregation 구조이다. 네트워크의 많은 block 규모의 feature를 병합하여 지엽적인 물체의 특성을 확인한다. Skip 연결에 대한 DenseNets의 관점은 따르면서, 깊이와 효율성을 개선시킨 tree 구조를 도입하였다.

이러한 DLA 구조를 이용하여, 각 HDA 상에서는 같은 stage 내부의 feature map들의 connection을 만든다. IDA는 다른 resolution, 즉 다른 stage와의 지속적인 connection을 만든다. 이러한 새로운 feature aggregation method를 통해, Semantic Segmentation, Boundary Detection 등에서의 모델 성능을 향상시킬 수 있다.

2.3. Objects as Points (CenterNet)

CenterNet[8]은 이미지 상의 물체 인식을 1-stage detector 방식으로 수행한다. 조밀한 AnchorBox들 간의 겹친 정

도를 분석했던 기존 신경망들과 달리 CenterNet[8]은 물체의 중심점과 Box size를 동시에 찾아내며, 여러 Anchor들을 묶을 필요가 없어 빠른 속도를 보인다. 네트워크에서는 먼저 backbone network를 통해 이미지의 heatmap과 box size, center offset으로 원래에서 4차원 늘어난 결과를 얻는다. 이후 heatmap은 3×3 maxpooling을 이용해 local maxima들을 뽑아내며, 이들에 논리값 연산을 적용해 최종 중심점을 결정하고, 나머지 결과값들을 박스 크기를 결정하고 미세 조정을 하는데 사용한다. Heatmap을 구할 때에는 Focal Loss가 사용되었는데, 실제 경계의 중심점 (가우시안 커널을 거친 Ground truth center point 값이 1)에서와 다른 점에서의 식의 형태가 다르며, 각 식에는 학습이 덜 됐을 때에는 Loss를 크게 만들지만 학습이 진행되며 무의미해지는 항이 포함되어 있다.

이 논문에서는 불필요한 과정들을 생략하고 Single Keypoint Estimation으로부터 다른 신경망들과 크게 다르지 않은 성능을 냈다는 것을 강조하고 있다. 또한 쉽게 3D object detection, human pose recognition으로 확장될 수 있다고 주장하며, 이 프로젝트에서는 이러한 점을 활용해 볼 것이다.

3. 데이터셋

이번 프로젝트의 목적을 달성하기 위해서는 사람과 이륜차를 학습해서 이의 3d bounding box를 얻어낼 수 있는 모델의 학습이 필요하다. 이번 절에서는 기존의 CenterNet이 이 task를 달성하기 위해 선택한 dataset과 이를 사용한 방법, 이를 변형해서 우리가 이 프로젝트에서 사용한 dataset의 선정과 data preparation 과정을 살펴본다.

3.1. CenterNet의 3D bounding box task

CenterNet[8]은 물체의 3D bounding box를 감지하는 task를 위해 KITTI dataset[2]을 사용했다. KITTI[2] 2D object dataset은 자율주행 상황에서 여러 object를 감지해서 자율주행차가 적절한 행동을 취하기 위한 dataset으로, 7481장의 학습용 이미지를 제공하며 이는 모두 9개 카테고리에 속하는 80256개의 annotation을 갖고 있다.

각 이미지에 대해서는 해당 이미지를 활용한 카메라의 calibration 정보를 제공한다. 각 annotation에 대해서는 object의 image boundary 밖 존재 여부(truncated), object가 다른 object에 의해 가려졌는지의 정도(occluded), 카메라 좌표계에서의 object의 관찰각(alpha), 이미지에서의 2D bounding box(bbox), 물체의 크기 정보(dimensions), 카메라 좌표계에서의 object의 위치 정보(location), 카메라 좌표계에서의 object의 Y축 방향 회전각의 정보(rotation_y)를 제공한다.

CenterNet[8]은 이 정보를 갖는 KITTI[2] 2D object detection dataset을 COCO format으로 변환한 후에 DLA-34[7]와 같은 적절한 Backbone Network를 이용해 구성한 신경망에 학습시키는 방식으로 3D bounding box task를 수행한다.

이때 KITTI[2] 2D object detection dataset은 test set에 대해서도 위에서 언급한 것과 동일한 형태의 데이터를 얻어내게 하고 추가로 detection 결과에 대한 score를 추가로

Category	보행자, 이륜차, 사람이 탑승한 이륜차
Image	Camera calibration 정보
Annotation	KITTI dataset에서 제공하는 모든 정보

Table 1. 학습에 사용할 Dataset이 제공해야 하는 필수 데이터

제출하게 하므로 CenterNet[8]도 이에 맞게 위에서 언급한 train set과 동일한 형식에 score만 추가된 결과를 추론 결과로 반환한다.

3.2. 프로젝트를 위해 필요한 데이터의 종류

우선, CenterNet[8]이 KITTI dataset[2]에서 주어진 모든 데이터를 활용하기 때문에 위에서 언급한 이미지, annotation에 주어진 정보들이 모두 필요하다. 한편 object의 카테고리에 대해서는 KITTI dataset[2]이 제공하는 9개의 카테고리 중 보행자와 자전거에 탑승한 사람 2개의 카테고리와 더불어 자전거 또는 오토바이를 포함하는 이륜차 자체에 대한 object annotation이 필요하다. 즉, 이를 표로 요약하면 Table 1과 같다.

그러나 KITTI dataset[2]은 본 프로젝트를 진행하는데 필수적인 카테고리 정보를 제공하고 있지 않기 때문에 다른 dataset을 찾아야 했으며, 그 결과로 위의 조건을 모두 충족시키면서 훨씬 더 많은 이미지와 annotation을 제공해 줄 수 있다고 판단한 NuScenes dataset을 선정하게 되었다.

3.3. NuScenes 데이터셋

NuScenes dataset[1]은 Motional 사에서 공개한 자율주행 상황에 대한 매우 큰 크기의 dataset이다. 6개의 카메라와 1개의 LIDAR, 5개의 RADAR, 1개의 GPS, 1개의 IMU를 통해 얻은 데이터를 제공하며 전체 dataset은 모두 1000 개의 scenes, 약 140만장의 카메라 이미지, 약 39만개의 LIDAR Sweeps, 약 140만개의 RADAR sweeps, 약 140만개의 object bounding box를 제공하고 있다.

또한, 카테고리 역시 위 절에서 설명한 필요조건을 모두 만족하는데, 모두 23개의 카테고리 정보를 제공하며 보행자, 자전거, 오토바이, 이륜차를 탑승한 사람의 카테고리를 제공해 이륜차의 종류까지 구분한 조금 더 세밀한 학습이 가능하다.

추가로, 공식적으로 제공하는 NuScenes to KITTI format converter가 있기 때문에, 이를 CenterNet[8]에서 제공하는 KITTI to COCO format converter와 결합하면 매우 쉽게 dataset을 학습이 가능한 형태로 변환할 수 있다는 장점도 있는 dataset이다.

3.4. 데이터셋 정제 및 준비

NuScenes dataset[1]은 필요로 하는 조건들을 모두 만족하지만 KITTI dataset[2]에 비해서 데이터의 양이 매우 많아 전체 데이터를 모두 사용할 경우 모델의 정확도 면에서는 분명한 도움이 되겠지만 학습에 필요한 computational resource가 너무 많다는 단점이 있다. 따라서 전체 데이터 중 일부를 사용하되, KITTI dataset[2]이 train set으로 제공한 데이터 중 본 프로젝트에서 필요한 object annotation과

Categories	Pedestrian, Bicycle, Motorcycle, Rider
# of images	6539 images Train/Validation split: Random, Train: 4692 images, Validation: 1173 images, Test: 674 images
Annotations (train)	Pedestrian: 19635 Bicycle: 646 Motorcycle: 575
Annotations (validation)	Pedestrian: 4835 Bicycle: 156 Motorcycle: 146
Annotations (test)	Pedestrian: 1963 Bicycle: 81 Motorcycle: 22 Rider: 694

Table 2. 실제 학습에 사용한 데이터의 통계

그것들이 포함된 이미지의 수보다는 많은 데이터를 부분적으로 변환하여 사용했다.

NuScenes dataset[1]의 전체 데이터는 10개의 blob으로 나뉘어서 제공되는데, 본 프로젝트에서는 이들 중 blob 01부터 03까지에서 Front camera와 LIDAR의 조합으로 얻어진 10120장의 KITTI format 이미지를 사용했다. 이들 중 프로젝트에서 필요한 카테고리 중 적어도 하나 이상을 포함하는 6539장의 이미지를 다시 선정했고 이들 이미지와 annotation에 대해서 필요한 카테고리 정보만 남긴 KITTI format의 데이터를 얻을 수 있었다.

얻은 데이터의 train, validation, test split은 우선 프로젝트의 목표가 '이륜차에 탑승한 사람'에 대한 정보 없이 '이륜차'와 '사람'에 대한 학습만으로 탑승 여부를 가려내는 것에 있으므로 'Rider'가 포함된 이미지는 모두 test split에 속하도록 구성했다. Train, validation split의 경우 test split에 쓰인 이미지를 제외한 5865장의 이미지 중 임의로 80%인 4692장을 train set으로, 20%인 1173장을 validation split으로 사용했다. 모델의 학습에 사용한 데이터의 통계는 Table 2와 같다.

4. 모델 설정 및 학습

선정한 모델은 이후 사람의 탑승 여부를 확인할 수 있도록 방향 데이터를 추정할 수 있어야 한다. 기존에 선정한 모델들은 2D 상의 정보만을 추출할 수 있어, 이로부터 물체가 바라보는 방향을 구체적으로 추정할 수 없었다. 이를 위하여 선정한 모델은 DLA[7]-34를 backbone으로 하고 있으며, 입력 데이터를 normalize한 후 촬영 시의 시야각 데이터(alpha)에 따라 회전 변환하여 각 물체의 깊이(depth)와 3차원 좌표(dim)을 통해 각 물체를 포함하는 3D bounding box와 그 물체가 바라보는 방향을 추정한다.

모델 선정 이후 이를 직접 활용할 수 있도록 약간의 변화를 주었다. 입력 데이터를 축소 변환하고 normalize 함에

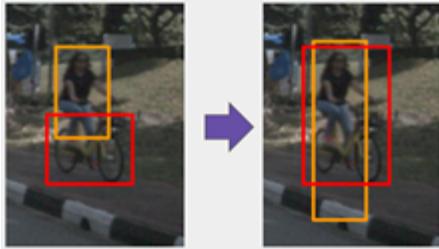


Figure 4. 사람과 이륜차의 2D bounding box 확장 예시

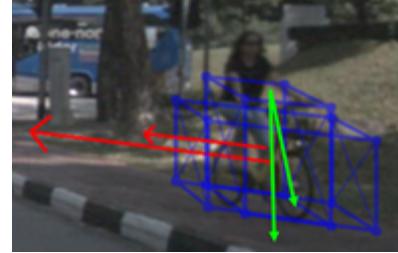


Figure 5. 사람과 이륜차에 대해 얻은 3D 정보 예시

있어서 사용하고자 하는 데이터셋에 맞게 수치를 조정하고, 인식하는 물체의 종류를 pedestrian, bicycle, motorcycle의 세 종류로 국한하였다.

이후 이를 batch size 8로 총 70 epoch만큼 학습시켰다. 이 때, 5번의 training epoch마다 한 번씩의 validation 과정을 수행하였다. learning rate는 1.25×10^{-4} 이 되 step에 따른 learning rate decay를 수행하여, 45 epoch와 60 epoch에서 $\frac{1}{10}$ 으로 감소시켜 최종적으로 1.25×10^{-6} 까지 내려간다. 학습 환경은 Google Colab에서 nVIDIA Tesla K80 한 대를 사용해 한 epoch 당 약 30분의 시간이 경과하였고, 총 학습 시간은 (validation 과정을 포함해) 약 40시간이 소요되었다.

5. 이륜차 탑승 여부 판정

앞에서 훈련용 이미지들을 선별하고 모델을 학습시킨 것은 CenterNet[8]을 이용해서 이미지 내에서 사람이나 이륜차로 추정되는 부분을 찾아내며 그와 동시에 3D 정보를 추정하기 위해서였다. 이번 절에서는 찾아진 사람, 이륜차들에 대해서 실제로 탑승자를 이루는 쌍이 무엇인지 찾아내기 위해서 이 프로젝트에서 사용한 기법에 대해서 설명한다.

학습시킨 모델을 backbone으로 CenterNet[8] Evaluation을 하는 과정만을 모듈로 추려내면 heatmap estimation, classification, non-maximum suppression 등의 과정은 모듈에 내재된다. 결과적으로 입력은 이미지이고 출력은 클래스 별로 나누어진 annotation이 된다. 이 annotation은 COCO format으로 크게 BBox정보, 3D BBox 정보, 신뢰도로 나눌 수 있다. 이 중 3D정보는 우리는 이륜차가 일반적으로 매우 얇은 두께를 가짐을 알고 있기 때문에 이 사실을 제외하고 방향 벡터, 중심 벡터, y축 방향 높이로 정보들을 압축하여 사용할 수 있다. 이런 정보들을 이용해서 사람과 이륜차 중 후보가 되는 쌍을 선별하고, 선별된 쌍들에 대해서 실제 탑승자인지 여부를 판단하게 된다.

먼저 탑승자인지 판단할 사람, 이륜차 쌍을 선별해 내야 한다. 탑승의 의미 그대로 받아들이면 이륜차의 바로 위에 사람이 있는지 판단하면 되지만, 사람과 이륜차가 어느 정도는 겹쳐 있는 구간이 존재하기 때문에 기준이 애매하고 여러 사람과 여러 이륜차가 공존하는 경우에는 여러 쌍이 되는 경우를 비교해줘야 한다. 그래서 정량적 기준으로 nms에서 널리 쓰이는 IoU 지표, 즉 2D BBox의 $(PiX't)/(iiX't)$ 를 사용했으며, 사람이 이륜차 위에 있

다는 사실은 Figure 4의 왼쪽과 같이 사람은 밑으로 2배 늘린 BBox, 이륜차는 위로 2배 늘린 BBox를 사용하는 것으로 반영했다. 이러면 똑바로 놓인 카메라로 찍힌 이미지에서만 작동하는 지표가 되나 물체 인식, 특히 자율주행 분야에서 쓰이는 이미지는 대부분이 이 조건을 만족하기 때문에 그대로 채용했다. 세부적으로는 각 이륜차가 IoU 지표가 가장 높게 나타난 사람을 택하게 되고, 이 때 IoU 지표가 특정 threshold($=0.2$)를 넘지 못하는 경우에는 어떤 사람과도 매칭되지 못 할 수 있다. 예외적인 경우로 어떤 이륜차가 자전거로도, 오토바이로도 인식되기도 하고, 여러 이륜차가 겹쳐 있는 경우도 있는데, 이런 경우 서로 다른 이륜차가 같은 사람을 택하는 경우가 있다. 이때는 한 사람이 여러 이륜차를 동시에 탈 수는 없다는 지식에 따라서 이륜차 중 신뢰도 값이 가장 높은 이륜차가 그 사람을 택하게 된다.

사람과 이륜차의 탑승자 후보 쌍이 선별되었으면 각 탑승자 후보가 실제로 탑승자인지 여부를 판단할 방법이 필요하다. 이를 위해서 탑승자에 대한 일반적인 지식인 ‘사람과 탈 것은 같은 방향을 향한다’라는 점으로부터 얻는 성질인 방향의 유사성과 ‘사람은 안장 위에 있다’라는 점으로부터 얻는 성질인 공간적 유사성을 사용할 것이다. Figure 5처럼 얻은 3D 정보를 바탕으로 방향벡터를 얻어낼 수 있기 때문에 이들의 cosine similarity로 방향의 유사성을 표현한다. 또한 나머지 3D 정보인 중심벡터와 y축 방향 높이를 이용해 사람의 top point에서 Cycle의 bottom point를 향하는 벡터를 구성할 수 있고 이 벡터와 y축 방향의 단위 벡터의 cosine similarity로 공간적 유사성을 표현한다. 최종 기준이 되는 점수는 방향적 유사성과 공간적 유사성을 곱한 것의 절댓값이다. 두 유사성이 코사인 값이기 때문에 최종 점수는 0과 1사이의 값이 되고, 이 점수가 특정 Threshold($=0.5$)를 넘으면 탑승자로 판단한다.

6. 결과 분석

프로젝트에서 만들어낸 모델은 이미지 내에서 탑승자를 사람과 이륜차의 쌍으로 판단하여 찾아낸다. 모델의 정확도를 분석하기 위해서는 Ground Truth에서 탑승자가 그 위치에 있는지 여부로 판단해야 한다. 여기서 결과는 두 BBox의 쌍이고, GT에서는 Rider의 BBox 하나이기 때문에 먼저 결과 BBox를 left top corner에서 right bottom corner로의 확장된 BBox로 변환하고 IoU를 계산하여 판단하였다. Threshold는 0.125로 일반적으로 cycle은 너비가

Image	Number	Proportion(%)
Total Test Images	674	—
Has Rider	60	8.90
Has Rider (GT matched)	56	8.30

Table 3. 최종 인식 결과

넓게 인식되는 것에 비해 Rider들은 세로로 길게 BBox가 쳐진 점을 반영한 값이다.

최종 결과는 Table 3과 같다. 탑승자를 사람과 이륜차로 분리해서 인식해내는 능력(8.90%)은 떨어졌으며, 사람과 이륜차 쌍이 탑승자임을 판단해내는 능력(93.3%)은 높게 나타났다. 첫 번째 과정에서 나타난 오류로는 탑승자를 사람이나 이륜차 중 한 종류로 합쳐서 인식해버리는 경우가 있었으며, 사람이 너무 많이 인식된 때 이륜차가 잘 인식되지 않는 경우도 나타났다. 원인은 NuScenes[1] 데이터셋에서 지나치게 다른 물체에 가려진 물체도 레이블에 포함되어 있었던 점, 학습 데이터에 탑승자에서의 사람과 이륜차는 없고 독립된 사람과 이륜차만 있었던 점, 학습 데이터량의 부족 등으로 판단된다. 두 번째 과정에서 탑승자 여부를 잘못 판단한 경우는 주로 쌍이 잘못 지어진 경우로 이륜차 앞을 지나가는 다른 사람이 있었던 경우, 두 이륜차가 겹쳐 있었던 경우 등이 있었다. 이런 경우 주로 공간적 유사성이 낮아서 탑승자가 아닌 것으로 판단되었으나, 0.51등 아슬아슬한 수치로 기준을 통과한 경우에는 결국 GT와 매칭되지 않았다.

7. 결론 및 개선점

프로젝트를 진행하며 가능한 개선점들이 고려되었다. 데이터셋과 관련되어 많은 부분이 개선되었지만, 시간적인 한계로 미처 개선하지 못한 부분에 대해서도 추가적인 논의가 이루어졌다.

논의된 개선점은 크게 두 가지 범주로 볼 수 있다. 학습 관점과 모델 관점으로 나눌 수 있다. 학습 관점에서는, 준비한 데이터 세트 크기의 부족으로 인해 과소적합의 가능성을 완전히 배제할 수 없었던 점이 있다. 이번 프로젝트에서, NuScenes[1] 데이터 세트의 변환을 통해 우리가 원하는 공간적 정보를 가지는 데이터 세트를 준비하였는데, 시간적 제약과 기존의 라벨링된 데이터 자체 양의 한계로 인해 과적합을 완전히 배제할 수 있을 만한 데이터의 양을 확보하지는 못했다. 또한, 이미지 처리과정에서 표준화과정에서의 평균과 분산 변수, 혹은 손실함수 관련 변수에 관한 심화적 논의가 온전히 이루어지지 못했다.

이번 프로젝트에서는 일반적인 분류가 아닌 특정 라벨에 대한 분류에 관심을 두고 있으므로, Ray-Tune[4] 과 같은 프레임 워크를 이용한 Hyperparameter search를 통해, 성능향상을 기대할 수 있을 거라 기대된다.

모델 관점에서는, 학습시킨 다중 레이블 분류의 고질적인 문제에서 기인된 개선점이 고려되었다. 다중 레이블 분류에서, 비슷한 공간상에 둘 이상의 물체가 있는 경우, 점수가 높은 물체가 존재하는 경우 하나로만 라벨링이 이루어 진다. Rider 판정을 위해서는 사람과 이륜차 각각에

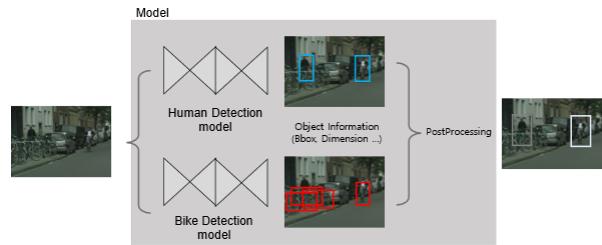


Figure 6. 사람과 이륜차를 별도로 학습하는 분류기

대해 독립적인 라벨링이 올바르게 이루어지는 것이 바람직하다. 이런 개선점 하에, Figure 6 두개의 분류기를 이용한 Grouping 방법을 제안할 수 있다.

사람을 분류하는 분류기와, 그 이외의 탈것에 대한 정보를 분류하는 분류기의 두 모델을 쓴다면, 우리가 원하는 방향의 라벨링이 잘 이루어질 것이며, 이번 프로젝트에서 설계된 후처리 기법을 통해서 Rider 검출이 더욱 잘 이루어 질 수 있을 것이라 기대된다. 이와 같은 분리된 분류기 방법론은 이번 프로젝트뿐 아니라, 실제 교통상황에서 각 개체에 대한 주의도가 다른 현실상황에서도 고려해야 할 방법론이라 생각된다.

마지막으로 프로젝트를 요약하면 다음과 같다. NuScenes[1] 데이터 세트를 통해, 기존의 CenterNet[8]에 적용시킬 수 있는 데이터셋을 수집하였고, Google Colab 환경에서 CenterNet[8] 구조를 이용하여 사람, 자전거, 오토바이의 3-label classifier를 학습시켰다. 분류된 개체는 Backbone Network를 통해 각각의 공간적 정보를 출력한다. 이러한 공간적 정보를 활용하여 방향적, 공간적 유사성을 후처리를 통해 수치화 하였고, 이 수치를 활용하여 Rider 판정을 이루어 냈다. 결론적으로, Person, Cycle을 Grouping 하여 Rider를 판정하는 부분에서는 유의미한 성능을 확인할 수 있었으며, 나아가 딥러닝 학습 결과를 통해 얻어진 라벨링된 물체에 대한 공간적 정보들을 통하여 추가적인 공간적 해석의 가능성을 확인할 수 있었다.

References

- [1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Lioung, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 3, 5
- [2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 3
- [3] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2
- [4] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018. 5

- [5] Alejandro Newell, Zhao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 2274–2284, Red Hook, NY, USA, 2017. Curran Associates Inc. [1](#)
- [6] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hour-glass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. [1](#)
- [7] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. [2](#), [3](#)
- [8] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019. [1](#), [2](#), [3](#), [4](#), [5](#)