## [자연어처리 - 머신러닝기초 팀팀 클래스 과제] 인공지능을 활용해서 사회문제 해결하기

**Object Detection을 활용하여 오토바이 운전자 및 승객의 헬멧 착용 여부를 판별하여 미착용 문제를 해결하기**

**자동차IT융합학과 20203330 김명준**

**1. 서론**

**1)인공지능의 활용한 문제 해결**

지금의 사회는 끊임 없는 산업 혁명과 발전을 통해서 지금의 인공지능 사회까지 발전하였습니다. 전에는 인공지능이 다소 두렵지 않던 시대에서 지금의 인공지능의 비약적인 발전은 모든 산업내에서 핵심적입니다. 특히 인공지능은 인간 대비 시간, 돈, 장소등의 자원의 모든 측면에서 유리합니다. 옛날에는 인공지능의 성능이 인간보다 낮아서 일부분만을 인공지능으로 대체하거나 인간의 보조 도구로서 사용되었다면, 이제는 결과를 인간과 비슷하게 혹은 동일 자원으로는 더욱 높은 결과를 뽑아내어 이제는 전면 모든 부분을 인공지능으로 대체하는 부분도 생겨나고 있습니다.

특히 인공지능은 값(data)를 기반으로 결과를 예측하는 일을 잘합니다, 이를 다시 말하면 인공지능은 훈련만 잘된다면 모르는 상황에서도 인간보다 더 결과를 이상적으로 뽑아낼 수도 있다는 뜻입니다. 인공지능이 사회에서 이제는 핵심적이라는 것은 이정도로 이야기하고, 이제는 이런 인공지능을 토대로 어떤 사회문제를 해결할지 이야기 해보겠습니다.

**2)문제 정의**

저는 자율주행, 컴퓨터 비전 등과 같은 분야에 관심이 있어서 이미지 인식이나, 2D→3D 복원 등에서 인공지능을 활용하는 것에 관심이 있습니다. 이런 관심을 기반으로 현재 사회의 문제인식을 하였을 때 “오토바이 헬멧 미착용”에 대한 문제를 중심으로 인공지능을 활용하여 이를 해결하려고 생각하였습니다.

이는 오토바이 운전자의 생명에 위험을 가하고 이는 전반적으로 운전자의 운전 안전 의식을 망가트리며 타인에게도 피해를 끼치는 상황이 올 수도 있습니다. 우리나라에서도 많이 일어나는 문제이지만 오토바이 유동 인구가 많은 인도에서는 오토바이 헬멧 미착용 문제가 너무나 빈번하게 일어납니다. 인도는 세계에서 가장 큰 오토바이 시장 중 하나입니다, 이러한 이유는 인도는 국토는 넓고 인구수도 엄청 많아서 유동 인구가 엄청난데 전철, 버스등의 대중교통이 잘 발전되어있지 않고 교통체계가 잘 잡혀있지 않아서 개인 이동 수단으로 오토바이는 인도에서 가장 핵심적인 이동수단이 되었습니다. 가장 핵심적인 이동수단으로 자리를 잡고 오토바이는 교통 사고 발생시 더욱 생명에 위험이 큰것에 불구하고 오토바이운전자나 승객중 헬멧 착용자는 극소수에 달합니다. 인도 내에서 헬멧을 착용하지 않아 발생하는 오토바이 교통사고 사상자는 한 도시 기준 90명정도 라고 합니다. 또한 인도는 마치 자동차처럼 승객을 포함해서 일가족등 다수가 오토바이에 탑승하고 있는 경우가 많습니다, 이는 더욱 교통사고 발생률을 높이는데 이러한 상황에도 불구하고 헬멧을 착용하지 않는 자들은 너무나 많습니다. 인도에는 약 1억 3,000만 대의 오토바이가 돌아다닌다고 합니다. 그래서 아무리 이런 상황을 제제하고 법률적으로 규제하려고 해도 인력은 부족하고 모든 부분을 규제하기는 거의 불가능 한 수준입니다. 그래서 이러한 부분에서 엄청나게 많은 헬멧 미착용 오토바이 운전자들을 단속하는데 인공지능을 활용한다면 기존에 비해서 인력도 크게 필요하지않을 것이고 처벌이나 제제를 훨씬 수월하게 하여 문제를 해결하는데 도움이 될 것이라고 생각합니다. 이 문제를 해결하는 것을 확장하면 헬멧을 착용하지않고 킥보드를 타는 인원, 운전 불가능한 인원등을 식별하는 부분까지의 문제도 해결가능하다고 생각합니다.

이렇게 오토바이 헬멧 미착용 문제를 해결하면 기대되는 효과로는 여러가지가 있는데 우선 교통사고 사망률을 감소시킬 수 있습니다. 헬멧 착용은 오토바이 사고시 운전자의 머리를 보호하는 역할인데 이를 제제함으로써 헬멧의 착용율을 높이면 중상 및 사망률을 낮출 수 있습니다. 또한 이를 제제함으로써 운전자들의 교통 법규 준수를 강요하며 이는 전체적인 교통 환경의 안전성을 높일 수 있을 것입니다. 또한 사고가 줄어 의료보험의 측면에서도 보험사가 운전자에게 보험료를 낮추게 하는 효과도 누릴 수 있고 전체적으로 사고 발생시 사고의 정도(중상 및 사망)를 낮추어 국가와 개인 전체의 의료비를 낮추어 경제적 이점을 누릴 수 있을 것이라고 생각합니다. 현재 사회에서 큰 문제임과 동시에 이 문제를 해결했을 때 오는 이득이 큰 점을 고려하면 이 문제를 해결하는 것이 좋은 접근이라고 생각합니다.





그림1. 인도에서의 오토바이 실태 (가족이 탑승하고 있으며 모두가 헬멧을 착용하지않은 상태)



그림2. 인도의 오토바이 탑승량 실태

**2. 적용 및 운용 방법**

**1)idea**

cctv나 경찰차량의 카메라에 이미지 인식(object detection)인공 지능을 통해서 이부분을 해결하려고 하였습니다. cctv나 경찰차량의 카메라를 통해서 실시간으로 헬멧 착용여부를 판별하거나 혹은 실시간으로 하지않고 헬멧착용여부를 판별하여 이후에 제제를 가하는 식으로 운용하여 해결할 수 있습니다. 이는 인간으로는 관리할 수 있는 영역의 한계를 해결해주고, 속도, 자원의 측면에서 유리합니다. 실시간으로 처리할때는 YOLO와 같이 실시간에 빠르게 inference를 할 수 있는 모델, 실시간이 안닐때는 정확하게 결과를 내는 transformer계열 기반의 DETR모델등을 활용하여 사용한다면 더욱 성능을 극대화 할 수 있을 것이라고 생각합니다.

**2)flow 정의**

구체적으로 data수집 방법등과 같은 부분에 대해서도 간략하게 계획을 세워보겠습니다. Cctv, drone, dash cam,경찰차에 카메라를 부착하여 data를 수집하고 이에 대해서 labeling하여 data를 얻습니다. 이렇게 얻은 data를 기반으로 이미지 인식에 강력하게 성능을 드러내는 cnn(convolutional neural network)구조 기반의 모델을 사용합니다, 요즘에는 vision분야에서도 transformer구조를 활용하므로 이는 크게 상관은 없습니다. 하지만 transformer기반의 모델도 결국은 image feature를 뽑아내는데 cnn기반의 모델을 사용한다는 점에서 cnn을 적극활용한다고 작성하겠습니다. 모델을 만들어낸 다음 label을 토대로 헷멧 미착용을 찾습니다, label의 구성은 motorbike, helmet, nohelmet으로 이루어져서 motorbike 탑승자에 한해서 helmet착용자와 미착용자를 식별합니다. 이러한 과정에서 여러 loss function을 활용하고, data augmentation(밤이나 먼지가 끼는 상황 ,악천후의 상황등 의 불안정한 환경에서도 잘 작동하게 강건성을 키우기 위해서 해상도나 밝기 blur등의 augmentation), hyperparameter tuning등을 활용합니다. 이렇게 잘 학습한 모델을 기반으로 교콩 카메라나 드론 등의 실시간 영상을 처리하는 모델, 실시간으로 처리하지않는 모델을 통합하는 시스템을 구축합니다. 그래서 헬멧 미착용자가 식별되면 이를 object tracking과 같은 추가 기술을 활용하여 제제를 하거나 알림을 보내 경찰관에게 제제를 가하게 지원합니다.

**3)구현**

실제 label 구성하기→ 2024\_aicity\_challenge에서 진행하였던 challenge의 dataset을 활용하여 적용하여 보았습니다. 인도에서의 오토바이 헬멧 미착용에 대한 challenge가 있어서 적용해서 해보면 좋을 것 같아서 활용해보았습니다.

구현은 detr이라는 model을 활용해서 helmet detection을 수행하였고 mmdetection이라는 framework를 활용하여 수행하였습니다. Train은 총 20,000장의 dataset으로 진행하였고 cocodataset으로 pretrain하여 진행하였으며, epoch은 60으로 진행하였습니다. Test data는 20,000장으로 train과 data양이 동일합니다.

**Ai city challenge에서 제공하는 인도의 helmet dataset규정 및 설명  
Challenge Track 5: Detecting Violation of Helmet Rule for Motorcyclists**

Motorcycles are one of the most popular modes of transportation, particularly in developing countries such as India. Due to lesser protection compared to cars and other standard vehicles, motorcycle riders are exposed to a greater risk of crashes. Therefore, wearing helmets for motorcycle riders is mandatory as per traffic rules and automatic detection of motorcyclists without helmets is one of the critical tasks to enforce strict regulatory traffic safety measures.

*Track 5: Detecting Violation of Helmet Rule for Motorcyclists*

Motorcycles are one of the most popular modes of transportation, particularly in developing countries such as India. Due to lesser protection compared to cars and other standard vehicles, motorcycle riders are exposed to a greater risk of crashes. Therefore, wearing helmets for motorcycle riders is mandatory as per traffic rules and automatic detection of motorcyclists without helmets is one of the critical tasks to enforce strict regulatory traffic safety measures.

* + Data

The training dataset contains 100 videos and groundtruth bounding boxes of motorcycle and motorcycle rider(s) with or without helmets. Each video is 20 seconds duration, recorded at 10 fps. The video resolution is 1920×1080.

Each motorcycle in the annotated frame has bounding box annotation of each rider with or without helmet information, for upto a maximum of 4 riders in a motorcycle. The class id (labels) of the object classes in this dataset is as follows:

* 1, motorbike: bounding box of motorcycle
* 2, DHelmet: bounding box of the motorcycle driver, if he/she is wearing a helmet
* 3, DNoHelmet: bounding box of the motorcycle driver, if he/she is not wearing a helmet
* 4, P1Helmet: bounding box of the passenger 1 of the motorcycle, if he/she is wearing a helmet
* 5, P1NoHelmet: bounding box of the passenger 1 of the motorcycle, if he/she is not wearing a helmet
* 6, P2Helmet: bounding box of the passenger 2 of the motorcycle, if he/she is wearing a helmet
* 7, P2NoHelmet: bounding box of the passenger 2 of the motorcycle, if he/she is not wearing a helmet
* 8, P0Helmet: bounding box of the child sitting in front of the Driver of the motorcycle, if he/she is wearing a helmet
* 9, P0NoHelmet: bounding box of the child sitting in front of the Driver of the motorcycle, if he/she is wearing not a helmet

The groundtruth file contain bounding box information (one object instance per line) for each video. The schema is as follows (values are comma-separated):

〈video\_id〉, 〈frame〉, 〈bb\_left〉, 〈bb\_top〉, 〈bb\_width〉, 〈bb\_height〉, 〈class〉

* + - 〈video\_id〉 is the video numeric identifier, starting with 1. It represents the position of the video in the list of all videos, sorted in alphanumeric order.
    - 〈frame〉 represents the frame count for the current frame in the current video, starting with 1.
    - 〈bb\_left〉 is the x-coordinate of the top left point of the bounding box.
    - 〈bb\_top〉 is the y-coordinate of the top left point of the bounding box.
    - 〈bb\_width〉 is the width of the bounding box.
    - 〈bb\_height〉 is the height of the bounding box.
    - 〈class〉 is the class id of the object as given in the labels information above.

The test dataset will contain 100 videos of 20 seconds each, recorded at 10 fps, similar to the training dataset videos. The test dataset will be released later.

* + Task

Teams should identify motorcycle and motorcycle rider(s) with or without helmet. Similar to the training dataset, each rider in a motorcycle (i.e., driver, passenger 1, passenger 2) is to be separately identified if they have a helmet or not.

* + Submission Format

One text file should be submitted, containing on each line, details of a detected object and the corresponding class id (as per the labels information). The submission format schema to be followed is as follows (values are comma separated).

〈video\_id〉, 〈frame〉, 〈bb\_left〉, 〈bb\_top〉, 〈bb\_width〉, 〈bb\_height〉, 〈class〉, 〈confidence〉

* + - 〈video\_id〉 is the video numeric identifier, starting with 1. It represents the position of the video in the list of all videos, sorted in alphanumeric order.
    - 〈frame〉 represents the frame count for the current frame in the current video, starting with 1.
    - 〈bb\_left〉 is the x-coordinate of the top left point of the bounding box.
    - 〈bb\_top〉 is the y-coordinate of the top left point of the bounding box.
    - 〈bb\_width〉 is the width of the bounding box.
    - 〈bb\_height〉 is the height of the bounding box.
    - 〈class〉 is the class id of the object as given in the labels information above.
    - 〈confidence〉 is the confidence score of the bounding box (values between 0 to 1).
  + Evaluation

The metric used to rank the performance of each team will be the mean Average Precision (mAP) across all frames in the test videos. mAP measures the mean of average precision (the area under the Precision-Recall curve) over all the object classes, as defined in PASCAL VOC 2012 competition.

Additional Datasets

External dataset or pre-trained models are allowed only if they are public. Teams that wish to be listed in the public leader board and win the challenge awards are NOT allowed to use any private data or private pre-trained model for either training or validation. The winning teams and runners-up are required to submit their training and testing codes for verification after the challenge submission deadline in order to ensure that no private data or private pre-trained model was used for training and the tasks were performed by algorithms and not humans.

위의 내용을 기반으로 ground truth를 만들고 이를 시각화 해보았습니다.

**Input image**: video100개에 대해서 하나의 video당 20sec, 10fps -> 200개의 image

1920 x 1080의 resolution (model운용에 따라서 resize필요)

**Label 구성** : 0:motorbike, 1:helmet, 2:nohelmet : 운전자 승객의 위치를 묶어서 그냥 헬멧을 쓴사람과 헬멧을 쓰지않은 사람을 잡게 label 재구성(기존에는 오토바이, 운전자(헬멧 착용 미착용), 승객1(헬멧 착용 미착용), 승객2(헬멧 착용 미착용), 승객0(헬멧 착용 미착용)-> 각각 운전자 탑승 뒤의 순서이고 승객0은 운전자 앞에 앉는 작은 아이를 의미), 오토바이에 탑승한 인원들만을 고려하게끔 오토바이는 그대로 식별 (coco dataset의 annotation형식을 따름으로 json구성은 image, annotation, categories로 구성 이는 각각 image의 정보, annotation 정보, class를 의미하여 json에 접근하여 불러올 image의 경로를 받아오고 이를 토대로 해당하는 annotation을 받는형식)

test시에도 json을 만들어야하는데 이부분에서는 annotation이 존재하지 않기 때문에 이미지의 경로 및 categories만 주면 된다. 이를 토대로 DETR이라는 model을 활용하여 helmet의 착용여부를 식별하게끔 모델을 훈련시켜보았다.

**output구성**: output은 image에 객체에 해당하는 class와 bbox를 출력한다. 시각화를 용이하게 하기 위해서 test시에 test를 할 이미지와 비교하여 bbox를 시각화하여 비교한다.

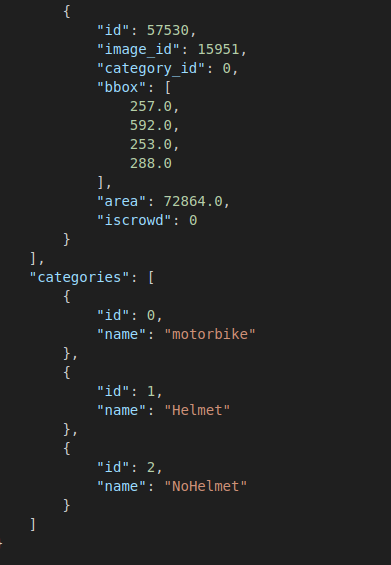


그림3, label.json 형식

**3. DETR**

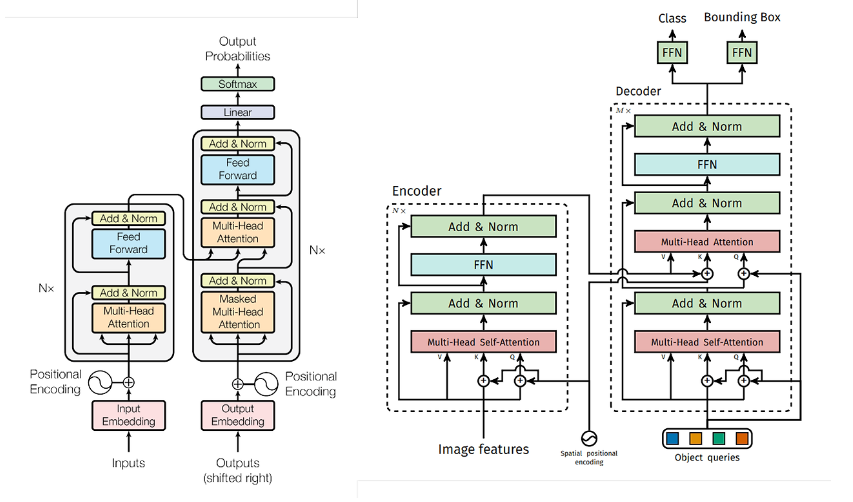
detr 모델 구조를 알고 쓰는 것이 좋아보이고, 이미지분야에 transformer를 적용한 model이라 자연어처리 교과목 때 배운 transformer를 떠올리며 모델에 대해서 알아보겠습니다. 우선 DETR은 End-to-End Object Detection with Transfomers라는 논문에서 제시한 모델 구조입니다. 위 논문은 기존에 이미지 인식분야에서 활발히 쓰이는 2step 방식의 object detection 방식에서 벗어나서 end to end형식으로 object detection을 수행하는 것을 제시하였습니다. 또한 기존의 모델 구조에서는 non maximum suppression같은 post process를 필요로 하였는데, object query로부터 나오는 결과물이 일대일로 대응하게 하여 이분 매칭방법으로 object detection을 수행하기 때문에 후처리를 따로 필요로 하지 않습니다. (기존의 transformer구조인 encoder, decoder구조를 기반으로 모델을 구성) 정리를 하자면 기존에는 anchor방식으로 간접적으로 object를 detection했다면, directly하게 object를 query를 가지고 직접 bounding box와 class를 예측하겠다라는 내용입니다.

그림4. Machine translation transformer model architecture & DETR model architecture

transformer에서 제안한 model구조인 encoder, decoder부분을 착안하였고 attention기법을 적용합니다, 차이점이라면 decoder부분에서 masked multi-head self attention을 수행하지 않는데요, 이는 task의 차이 때문에 query를 전체적으로 attention하는 부분에서 발생합니다. Machine translation부분에서는 decoder부분에 query들은 target sentence의 word로 구성됩니다, 그래서 다음에 나올 단어를 미리 attention하는 것은 cheating을 하는 것으로 inference시에는 활용하지 못하니 masking을 하는 것이였으나, object detection을 수행하는 task에서 query는 object query로서 각각이 물체를 나타내는 query로 작용하기에 한 이미지 내에서 object들 간의 context, 배경을 이해하는 과정이 필요하므로 이를 masking할 필요가 없고 이미 이미지가 주어지면 객체들이 동시에 존재하므로 이들을 attention하는 것이 cheating도 아니므로 masking을 수행하지 않습니다.

간단히 순서를 파악해보겠습니다.   
1번째로는 image feature map을 형성합니다, 이는 cnn backbone(resnet)을 거쳐서 나온 image feature를 1x1 conv layer 를 거치고 spatial dimension flatten 과정을 거쳐서 C x H x W에서 C보다는 작은 d라는 차원의 d x hw로 변경합니다. 이렇게 변경하는 이유는 2차원적인 feature를 attention할 수 없기 때문에 flatten하여 transformer구조에 태우기 위해서 입니다. 그렇다면 flatten을 하면 이들은 위치정보를 잃어버리기 되는데 이를 방지 하기 위해서 positional encdoing을 수행합니다. 이는 2D image를 flatten하였기에, spatial 정보를 따로 positional encoding을 하는 것이고, input차원이 d였을때 x,y 차원에 대해서 row, column- wise하게 2/d 크기의 sin, cos 함수를 적용한 이후 channel 별로 concat하여 수행합니다. 이렇게 만들어진 image feature를 가지고 encoder부분에서 multi head self attention을 수행하고 ffn과정을 거쳐서 최종적으로 decoder부분에서 쓰일 key value를 형성합니다.

decoder부분에서는 object query가 존재하는데요, 이는 learnable한 embedding query라고 보시면 됩니다, 또한 이를 각 객체 하나에 대응되는 bbox, class를 예측 하는 query로 보면 되겠습니다. Object query는 d x N이고 여기서 N은 한 이미지내에서 detection가능한 객체의 수 입니다. 이렇게 만들어진 query를 가지고 self attention밑 encoder의 출력과 cross attention을 한 결과물을 ffn을 2번을 진행하는데 이부분은 detr에서의 head부분입니다. 이 ffn은 각각이 bbox(localization)과 class를 예측하는 부분으로 이루어져 있습니다.

종합하여 정리해보면 여기서는 N개라는 정해진 객체를 이분매칭하여 detection하기에 NMS같은 후처리를 요하지않고, (N개 보다 작은 객체의 수가 이미지에 존재시 객체와 매칭되는 것이 아니라 객체가 없다는 no object라는 class와 매칭되게 한)한번의 forward로 object사이의 relation을 예측할 수 있다는 장점이 있습니다.

이렇게 해서 transformer기반의 구조를 object detection에 적용한 detr모델의 기법 및 architecture를 알아보았습니다.



그림5. 안개 낀 상황에서의 detection



그림6. helmet을쓴경우만있는사진 

그림7. helmet을쓴경우와 helmet을 안 쓴 경우의 사진

****

그림8. Detection을 잘 못한 경우: 다중 detection및 class틀림(helmet 미착용자를 검출 못함)

****

그림9. 안개상황속 helmet, no helmet을 잠 검출하였지만, 탑승객에 대해서 bbox소멸 문제의 사진

**4. 추후 개선 점 및 결론**

결과를 보면 예측했던 class가 바뀌어 버려서 일관성을 잃는 결과가 보이기도 하였다. 이러한 부분을 개선하기 위해서 confidence를 기준으로 confidence를 높게 예측 한 class로 일관성있게 유지하는 부분을 고려할 수 있을 것 같고, 아직은 운전자만 예측을 잘하고, 만약에 운전자와 승객모두 helmet을 다 쓰면 iou가 많이 겹쳐서 하나의 bbox로 결과가 나오는 경우가 있다. (동일 class에 관해서 동일 객체라고 판단한 후처리가 들어간 것인지 의문 )이를 해결하기 위해서는 2step으로 생각을해서 사람과 오토바이를 먼저 detection하고 난후에 rule을 걸어서 오토바이를 타고 있는 사람을 대상으로 helmet 착용여부를 모두 판별하는 식으로 개선을 할 수 있을 것 같다. 결과물이 생각했던 것보다는 좋지 않았지만 추후에 개선여지가 높고, 앞서 말한 부분을 개선한다면 더 좋은 결과물을 낼 수 있을 것이라고 생각합니다. 추가적으로 이 dataset을 가지고 chaellenge를 진행하였던 수상작들의 생각을 보면 여러 재밌는 생각들이 있었는데, 오토바이와 사람을 하나의 큰 bbox로 만들어서 crop하고 region of interest처럼 활용해서 그부분에 대해서 헬멧 여부를 판별하는 식으로 한 팀도 있었고, 대회는 class가 9개인데 class long tail 문제를 해결하기 위해서 앞서 말한 것처럼 사람으로 먼저 detection을 하고 helmet을 썼는지 안 썼는지 확인한 다음 오토바이의 이동 벡터를 기준으로 오토바이의 이동벡터 부분에 각 사람의 bbox중심 혹은 좌상단 좌표 가 가장 가까운 사람을 운전자, 승객1, 승객2 등등으로 rule형식을 적용하여 한 재밌는 곳도 있었다. 또한, 실전에 운용될 것을 확장하여 sort와 같은 tracking 알고리즘을 활용하여 detection한후 객체를 tracking하여 기록하는 방식 혹은 앞서 말했던 예측한 class가 바뀌는 문제를 sort와 같은 알고리즘을 통해서 동일 객체에 대해서는 confidence가 가장 높게 예측한 class로 일관지어서 유지하는 방식도 있었다. 이러한 부분들을 보완하고 조금 거 실시간운용적인 측면으로 개선한다면 충분히 현업에서 사용할 수 있을 것이라고 생각합니다.

결론적으로 오토바이 헬멧 미착용에 대한 문제 정의를 하고 이를 해결하기 위해서 인공지능을 활용하여 이를 해결하려고 했습니다. 실시간 운용이 궁극적으로는 나아가야하는 방향이기에 얼마나 모델을 경량화하여 운용할지 또는 더 간단한 방식으로 알고리즘을 적용하여 좋은 결과를 낼지 등등을 더 고려해야합니다. 저는 시간이나 실시간 운용측면을 고려하지는 않았기에 이부분을 고려하여 개선해 나가는 것이 중요한 것 같습니다. 실제 사회에서 문제가 되어서 인간으로는 해결하기 어렵고 인공지능을 활용한다면 이득을 많이 볼 것 같은 문제를 직접 정의해보는 부분도 매우 좋은 경험이였고, 다소 보기 어려운 상황이고 이런 특별한 상황을 다루는 dataset을 다뤄보고 인공지능을 적용해봤다는 측면도 좋은 경험이 되었고, 자연어 처리때 배운 transformer구조 기반의 모델을 object detection분야에 적용한 모델을 공부하고, 적용해본것이 매우 뜻 깊었던 것 같습니다. 다음에 기회가 난다면 위 보완할 부분들을 보완하고, 독자적인 모델 구조나 알고리즘을 개발해보고 싶습니다.

5. pretrain ablation

coco dataset으로 pretrain을 한 model의 parameter를 가지고 있는 것과 없는 것의 차이는 매우 큽니다. 이번을 계기로 그러한 부분을 더욱 크게 느꼈고, 수업시간에 배운 pretrain의 중요성에 대해서 한번 더 상기할 수 있었습니다. 이번에 detr모델을 훈련시킬 때 있어서 coco dataset으로 pretrain시킨 model로 돌려보는 과정과 그 parameter를 가지고 훈련을 진행하는 것과 최종적으로 60epoch훈련시킨 parameter를 가지고 훈련시키는 것의 차이를 시각화 해보았습니다. Downstream task도 detection, pretrain도 detection이라 head를 수정할 일은 없었고 pretrain모델 구조를 수정하지는 않았기에 parameter initialization부분에서의 차이만 존재했다고 생각합니다. 더군다나 detr과 같은 transformer기반의 모델은 수렴하려면 많은 양의 data를 필요로 하는데 이부분에서 pretrain은 필수적 이였다고 생각합니다. (오토바이 helmet train dataset이 20,000장밖에 되지 않은 것을 감안하면 pretrain을 하지 않았을 때는 detr과 같은 transformer구조 기반 모델은 매우 성능이 좋지 않을 것이라고 추측합니다)

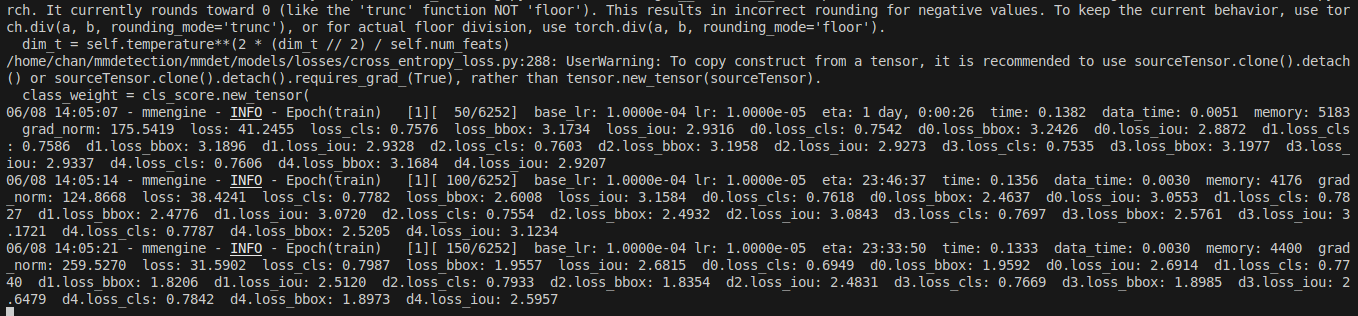


그림 10. coco dataset으로 pretrain을 하지 않고 돌렸을 때 초기 loss가 41.2455에서 시작

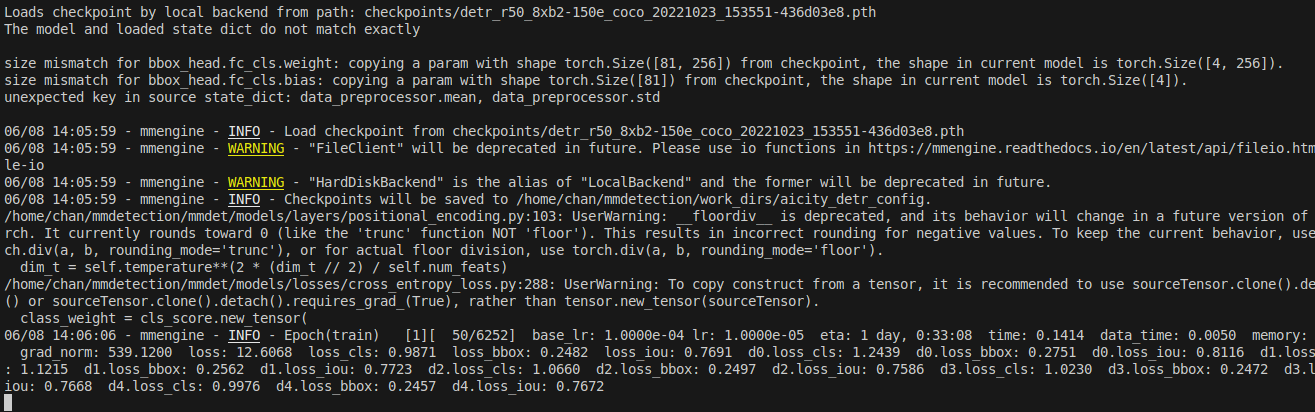


그림 11. coco dataset으로 pretrain을 하고 돌렸을 때 초기 loss가 12.6068에서 시작

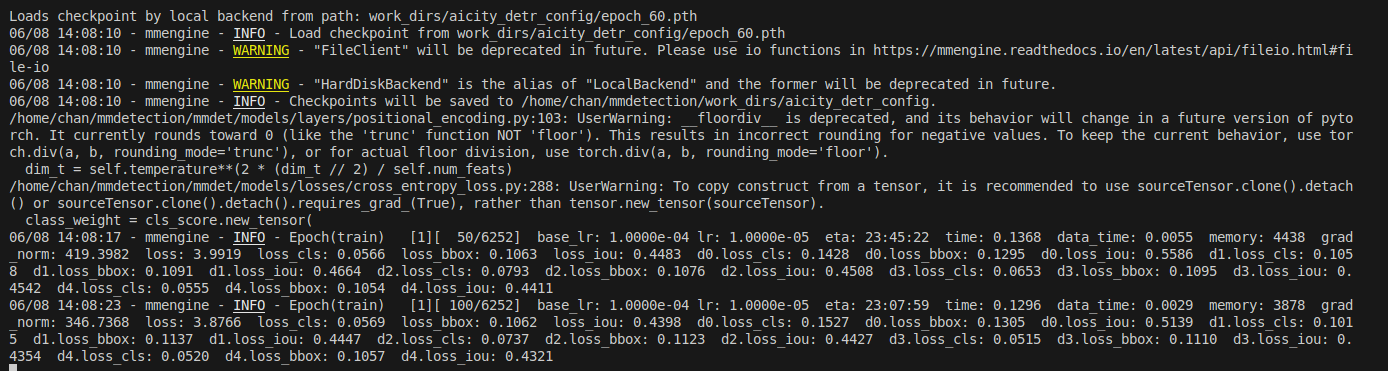


그림 12. Coco dataset으로 pretrain하고 helmet dataset에서 60epoch훈련한 parameter로 pretrain시 초기 loss가 3.9919에서 시작

제출 관련하여 data는 첨부하지 않았습니다. Data는 용량이 커서 따로 보관하고 있습니다(코드만 업로드하나 data file도 제출하여야 한다면 말씀해주시면 ssd로 직접 가져가겠습니다!) 제출 파일은 mmdetection에서 필요한 부분만 직접 수정하였고, 전 처리하는 코드나 시각화 혹은 json만들기 config, datset load, detr \_config등 일부분을 처리해야하는 dataset에 맞게, task에 맞게 맞춰서 수정하였습니다. 주요 코드는 **configs/detr/aicity\_detr\_config.py, configs/detr/aicity\_detr.py**, **configs/\_base\_/dataset/aicity\_coco\_detection.py, mmdet/ dataset/ custom\_coco.py**등을 기존에 참고하는 code들을 기반으로 만들었고 전처리는 대부분 train\_pre\_process에 있는 기존에 존재하는 전 처리 코드들을 참고하여 만든python file들로 전 처리하였습니다.