일단 수요자를 ''단독주택'으로 한정

고민한 방법 - model 측면

- 사람이 들고 있는 물체를 object detection 모델로 탐지하여 손에 그 물체가 있다가 사라지는 순간을 쓰레기를 버린다고 판단하기로 함
    - 물체에 대한 bounding box는 object detection 모델의 결과로 뽑고
    - 손에 물체가 있다고 판단하기 위해 손의 위치가 필요하므로 openpose 모델을 통해 손의 좌표를 뽑음
    - 이 두 모델을 합쳐서 구현할 수 있을 것 같음
    - openpose를 통해 '손' 이라는 좌표를 뽑고 그 좌표를 기준으로 손에 대한 bounding box를 침
    - 그 bounding box에 object detection model이 뱉은 '물체'에 대한 bounding box가 어느정도 겹칠 경우 손에 물체가 있다고 판단
- 사람이 들고 있는 물체에 대한 variation이 매우 크다
    - 데이터를 어떻게 만드느냐가 중요할 듯함
- 물체의 class가 무엇인지는 중요하지 않고 손에 물체가 있냐 없냐만 중요

데이터 수집 및 처리

- 높은 위치에서 영상을 각각 다른 각도에서 찍음
- 사진 한 장당 1초에 5장 정도의 frame을 뽑으려 했지만, 어짜피 같은 시간대에 같은 장소에서는 데이터가 큰 차이가 없고 1초에 1~2장만 frame을 뽑아도 augmentation으로 충분히 커버가 가능할 것이라 판단하여 1초에 2장만 뽑았음
- 데이터를 찍는 장소를 여러군데로 해야 함
- 시간대(조명)에 따라 다르게 찍어야 함
- 다양한 물체로 데이터를 만들어야 함
- class를 다 다르게 각자 부여하기엔 데이터가 한 클래스마다 너무 많이 필요할 듯 하고, 클래스를 하나로 하기에는 variation이 너무 클 듯 함
    - 우선은 하나의 클래스 'trash'만 부여서해서 학습을 시켜보고 결과를 보자
- '손'에 들어있는 물체를 혹시 손과 같이 bounding box를 쳐서 annotation을 하면 그 특징을 model이 학습하며 조금이나마 배울 수 있지 않을까 하여 물체와 그 물체를 쥐고 있는 손을 같이 동시에 bounding box를 치고 annotation을 함

학습 시 문제 및 모델 선택 과정

- realtime 모델에 성능도 어느정도 잘 나와야 함 - 작은 물체에 대한 탐지가 가능해야 함
    - 1 stage 모델에 대해 논문 search
- m2det라는 모델로 성능을 체크해보고자 함
    - 논문상에서 Yolo v3보다 속도 및 map가 좋다고 나오기 때문
- trash라는 class하나만 주고 학습을 한 결과 -> 아예 학습이 안됨
    - 아무래도 데이터에 대한 variation으로 전혀 데이터의 특징을 배우지 못한 것으로 보임

    - class를 늘려서 학습을 해볼까 하다가, 모든 데이터에 다시 각각 class를 부여하는 수고를 해보기 전에 pretrained model로 우선 해보고 가능성을 보기로 함
- COCO, VOC 데이터로 학습이 된 모델에 추가적으로 우리가 만든 데이터로 학습을 하기 위해 기본적으로 pretrained model을 제공하는 object detection model을 찾아봄
    - m2det의 경우 pretrained model을 제공하지 않기에 고려하지 않음
    - YOLO v3 (2018)
        - 가장 보편적으로 많이 쓰이는 realtime object detection모델
    - Training-Time-Friendly Network for Real-Time Object Detection(ttfnet) (2020, AAAI)
        - 이번 2020 AAAI에 accept된 논문, training 시간이 매우 짧으며 pertained model도 제공하고 실시간 및 성능도 좋음
    - 우선 우리가 만든 데이터로 학습하지 않고 돌려본 결과 yolo v3보다 ttfnet이 더 realtime에 성능도 좋았기 때문에 ttfnet을 사용하기로 결정

ttfnet

- 학습 시 loss가 에폭을 늘려도 내려가지 않고 들쭉날쭉 함
- 몇 가지 가정을 세워봄
    1. 비닐봉지, 과자 박스, 음료수 병 등 서로 다른 크기와 모양의 물체를 모두 'trash'라 설정한 것에 대해 variation이 너무 커서 아예 학습을 하지 못함
    2. 데이터를 만들 때의 규칙이 잘못되었을 수도 있음
    3. 논문 상의 컴퓨터 사양 세팅과 내 컴퓨터(1080ti)의 사양이 정확히 동일하진 않아 hyper parameter tuning이 필요할 수도 있음 (논문 상에선 multi gpu를 사용)
- 우선 가장 바로 해볼 수 있는 방법인 hyper parameter를 바꿔봄
    - 논문상에서 learning rate를 auto scale로 적용했는데 gpu 갯수만큼 나눠보니 loss가 잘 떨어짐
- 문제를 해결 후 pretrain model에 추가로 우리 데이터로 학습을 시킨 결과 손에 있는 물체의 class는 틀려도 bounding box는 잘 쳐줌
- label을 'person', 'trash'의 class label만으로 학습을 시켜본 결과 역시 결과가 잘 나오지 않았음.
    - COCO dataset의 원래 class label에 trash를 추가해서 학습을 시킨 것이 label은 잘 안맞더라도 bbox는 제대로 잘 쳐주는 경향을 볼 수 있었음

Main

- openpose를 통해 '손' 이라는 좌표를 뽑고 그 좌표를 기준으로 손에 대한 bounding box를 침
- 그 bounding box에 object detection model이 뱉은 '물체'에 대한 bounding box가 어느정도 겹칠 경우 손에 물체가 있다고 판단
- 손에 대한 bounding box를 치기 위해 사람에 대한 bounding box를 활용해야 함
    - 손에 대한 bounding box를 어떤 기준으로 칠 것인가(크기는 어떻게 할 것인가)
    - 사람의 bbox의 일정 비율로 정함
- object detection model의 경우 pretrain한 것에 추가로 손에 있는 물체를 학습시켰으므로 overfit 가능성이 큼 - 사람에 대한 bbox를 잘 찍어주지 못함
    - 추가로 데이터를 만들 때 사람에 대한 bounding box와 손에 들고있는 물체에 대한 bounding box

를 모두 쳐서 만듦
    - 학습 결과 결국 '손'과 '사람'에 대해 bounding box를 잘 쳐줌
- 그러나 이 때 꼭 '손'에 있는 물체만 bbox를 치는 것은 아니고 다른 물체도 당연히 쳐질 수도 있기 때문에 그 물체가 '손'에 있는 물체라는 것을 알기 위해서는 openpose를 통해 손의 위치를 파악할 필요가 있음
- 각 좌표마다 score값이 나오는데, 그 score가 0.7 이하인 것은 안나오게 함(threshold) - 손이 가려졌을 때도 낮은 score값으로 어떻게든 찍어내려는 것을 방지하기 위해

Final - 통합

- 어짜피 모델이 영상을 입력으로 받아도 한 frame별로 처리하므로 두 모델(ttfnet, openpose)을 각 frame마다 처리 하고 다음 frame으로 넘어가도록 할 수 있음
    - ttfnet이 Realtime이어도 openpose가 속도가 제대로 안나오면 결국 real time으로 할 수 없음
- Ttfnet 모델에 손에 있는 object가 잡히고 그것이 openpose를 통해 설정한 손의 경계 내에 들어간다하면 전역변수 num_object+1
    - num_object : 손에 들고있는 물체의 총 갯수
    - 이 num_object가 원래 갯수보다 줄었을 때 쓰레기를 버렸다고 판단
- 그러나 num_object가 준 이유는 반드시 쓰레기를 버린것만은 아닐 수도 있음
    - 사람 수가 a,b 총 2명인데 물체가 3개일 경우
    - case 1 : 사람 a가 물체 한 개, b가 물체 2개를 들고 있었음. 그런데 사람 b가 영상 내에서 사라져서 물체가 2개가 사라지고 한 개만 남음 -> 남은 사람1, 물체 1
    - case 2 : 만약 사람 a가 사라지고 b가 물체 하나 버림 -> 남은 사람 1, 물체 1
- 위 2가지 case를 어떻게 구분을 할까?
    - 사람을 tracking
    - 이전 frame과 현재 frame의 openpose 좌표간의 유사도를 통해 frame간 사람 tracking 가능
    - 사람을 tracking하면서 사람마다 dictionary 형태로 {사람 id : 물체 num}와 같이 정의해둔다면 사람이 사라지는지, 물체를 버리는 건지를 check할 수 있음
- 사람 수가 동일한데, num_object가 실제 fps의 반 정도의 frame(30fps라면 대략 15frame)이 지날때까지 원래 상태보다 작은 상태라면, 누군가가 쓰레기를 버렸다고 판단
- 성능이 100%가 아니고 데이터를 만들 때 손과 같이 보이는 것을 box를 쳐서 만들었으므로 아주 가끔(드문 확률로)
    - bbox를 치는 threshold를 조금 더 높이거나, 연속으로 최소 2~3frame정도 동안은 찍히는 것에 한해서만 num_object를 count해줌