



AI 경진대회

- 객체 검출 기반 범용적 쓰레기
무단 투기 검출 네트워크

MVP동찬
2023.10.24

Index

1. 프로젝트 개요
2. 활용 데이터
3. 기반 기술 사전 조사
4. 모델 개발 방법
5. 실험 및 평가
6. 활용 및 기대효과



프로젝트 개요

- ❖ 쓰레기 무단 투기로 발생하는 과도한 예산 지출, 환경오염, 미관 손상 등을 방지하기 위하여 이미지 데이터 분석을 통한 쓰레기 무단 투기 해결
- ❖ Object Detection 모델을 통한 무단 투기 쓰레기의 존재 여부와, 위치를 파악하며, 영상 내에 무단투기 쓰레기의 존재여부를 확인하는 모델을 제안하고, 추가적인 데이터셋을 제안함.
- ❖ 다양한 상황에서의 무단투기 쓰레기의 검출에도 강인한 네트워크를 제안함.

일반적 상황에
맞는 추가적인
데이터셋

모델 변경
augmentation
데이터 후처리

다양한 상황에
강인하고 쉽게
사용할만한 모
델의 제안

활용 데이터

기존 제공 데이터셋 -> AI HUB 포털 내 등재된 쓰레기 봉투 영상-이미지 데이터

- 학습 데이터셋 2885장, 테스트 데이터셋 371장으로 구성



추가적으로 필요한 부분

1. 모두 같은 해상도의 이미지
2. 동일한 type의 object (하얀색, 검은색 봉투)

-> 추가적인 데이터셋을 사용 및 일반적 상황을 고려하기 위한 augmentation 환경 고려

활용 데이터

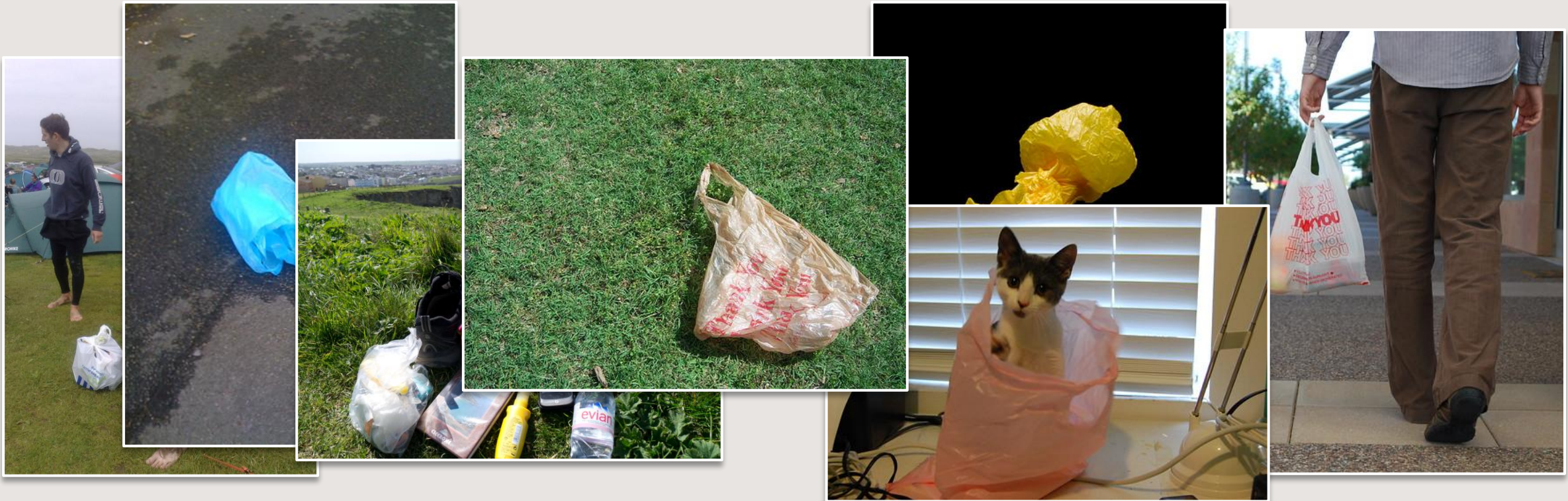
좀 더 일반적인 쓰레기 봉투의 특징을 추가해주기 위한 데이터셋 survey

-> PlasticVNOI dataset 사용 (일반적인 비닐봉지를 detection하기 위한 데이터셋)

-> 여러 resolution+ 여러 type의 데이터가 존재

-> 일반적인 비닐봉지를 모두 포함하므로 쓰레기 봉지라고 보기 어려운 outlier 데이터가 존재.

-> 이러한 데이터들을 정제하고 쓰레기 봉투라고 판단되는 약 200개 정도의 데이터만을 추가적으로 이용



활용 데이터

좀 더 일반적인 쓰레기 봉투의 특징을 추가해주기 위한 데이터셋 survey

-> PlasticVNOI dataset 사용 (일반적인 비닐봉지를 detection하기 위한 데이터셋)

-> 여러 resolution+ 여러 type의 데이터가 존재

-> 일반적인 비닐봉지를 모두 포함하므로 쓰레기 봉지라고 보기 어려운 outlier 데이터가 존재.

-> 이러한 데이터들을 정제하고 쓰레기 봉투라고 판단되는 약 200장 정도의 데이터만을 추가적으로 이용



기반 기술 사전 조사

객체 검출이란...

- 여러 물체에 대해 어떤 물체인지 분류하는 classification task
- 그 물체가 어디에 있는지 bounding box를 통해 위치 정보를 나타내는 localization task

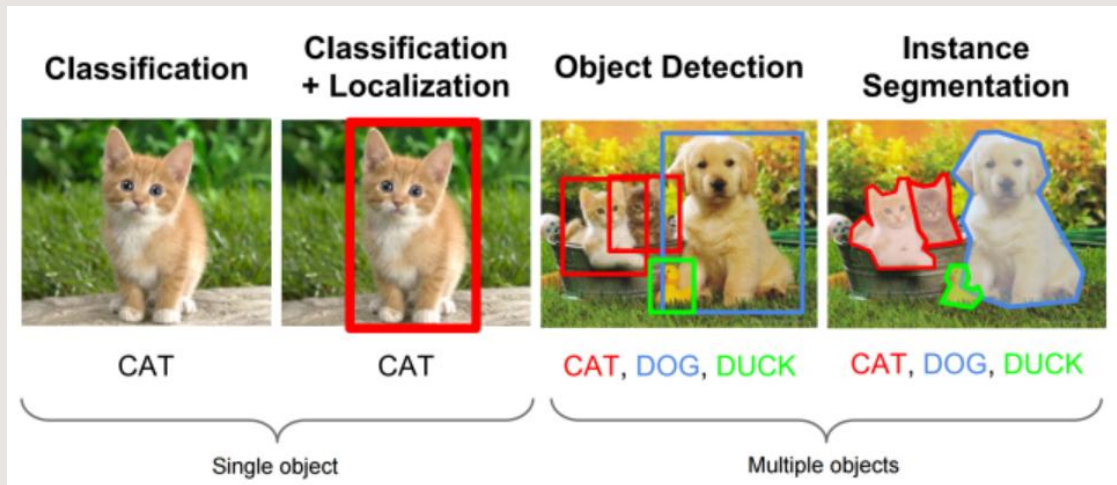


그림 1. 객체 검출 연구

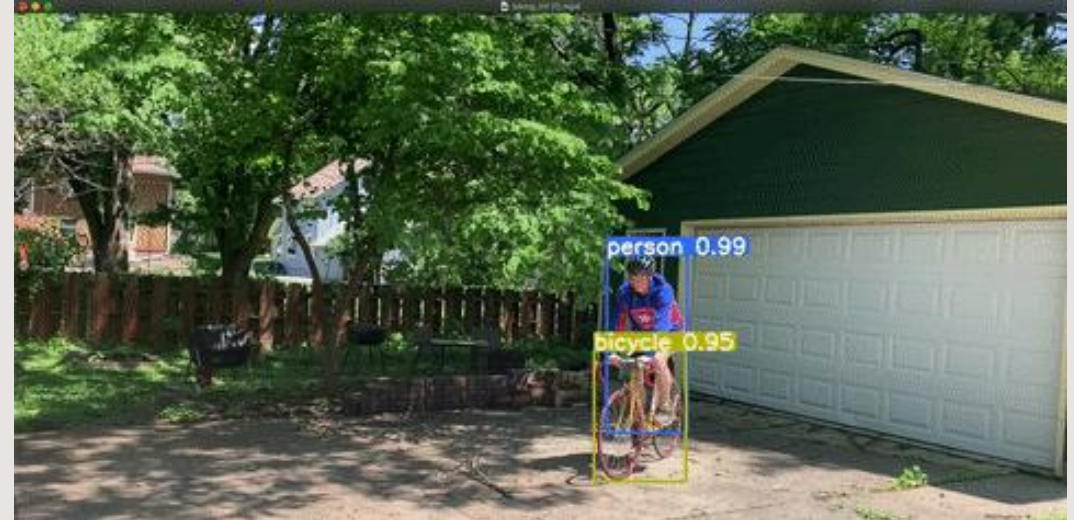


그림 2. 객체 검출 결과 예시

모델개발방법

• Baseline Model - YOLOv5

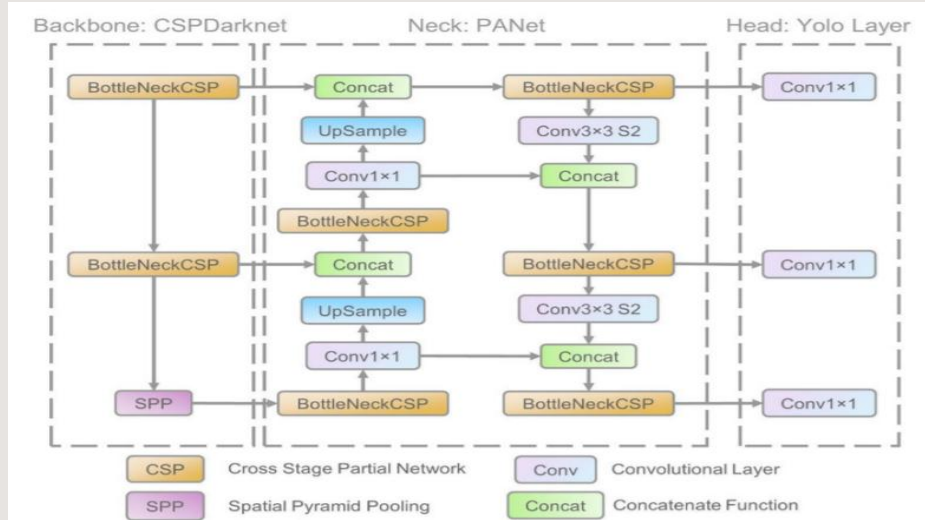


그림 9. YOLOv5 프레임워크

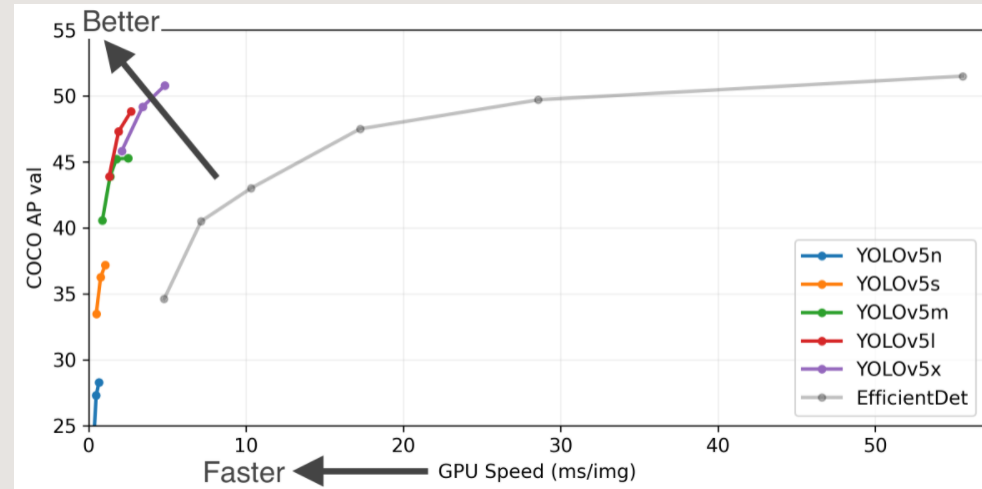


그림 10. YOLOv5 속도 대비 성능

- Backbone: CSP-darknet
- Feature map을 분리하여 연산하는 CSP 기법을 사용하여 성능 및 속도의 효율을 보임
- 최근에 나온 yolo 모델들 중 가장 변경 및 수정이 간편하며, 추후 개발을 통해 보드에 올리는 것도 제일 편한 것으로 알려져 있음

모델개발방법

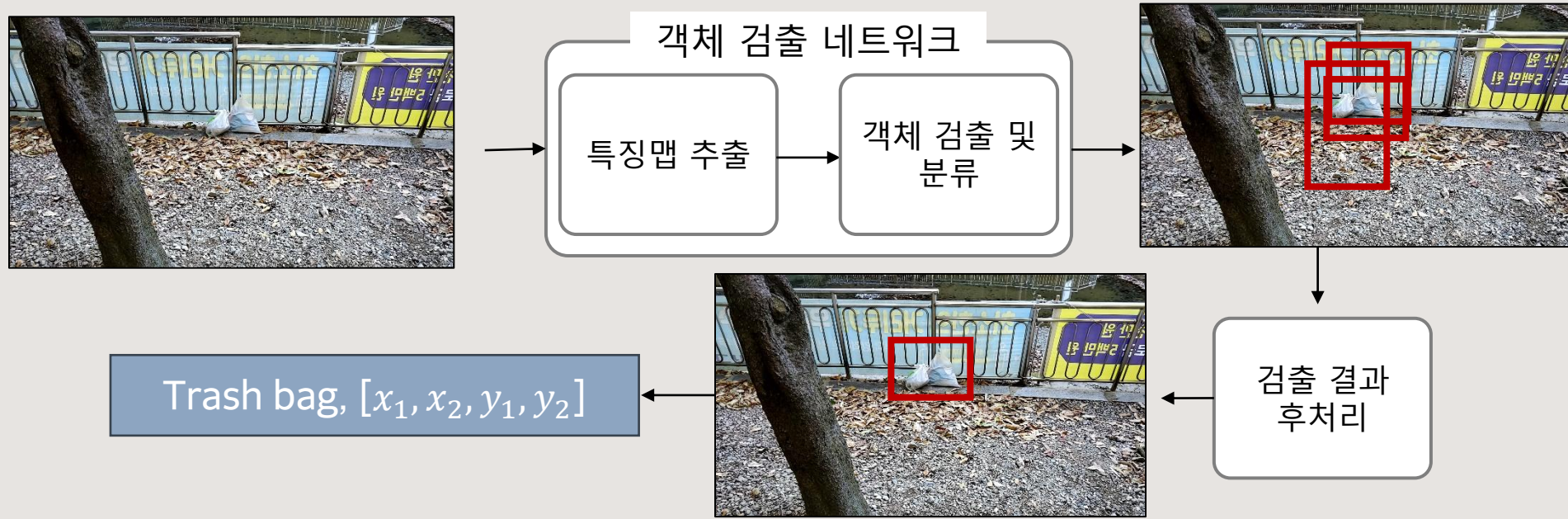


그림 11. 알고리즘의 전체 순서도

- 1) 네트워크의 input은 RGB 영상 프레임
- 2) CSP DarkNet을 사용하여 영상의 특징맵을 추출함
- 3) 객체 검출을 통해 Trash 에 해당하는 검출 박스만 추출하고, 동시에 confidence score를 측정함.
- 4) 검출 박스는 4개의 좌표값 (x, y, w, h)로 표현됨
- 5) 후처리 과정을 통해 상황에 맞는 detect를 하게 됨.

모델개발방법

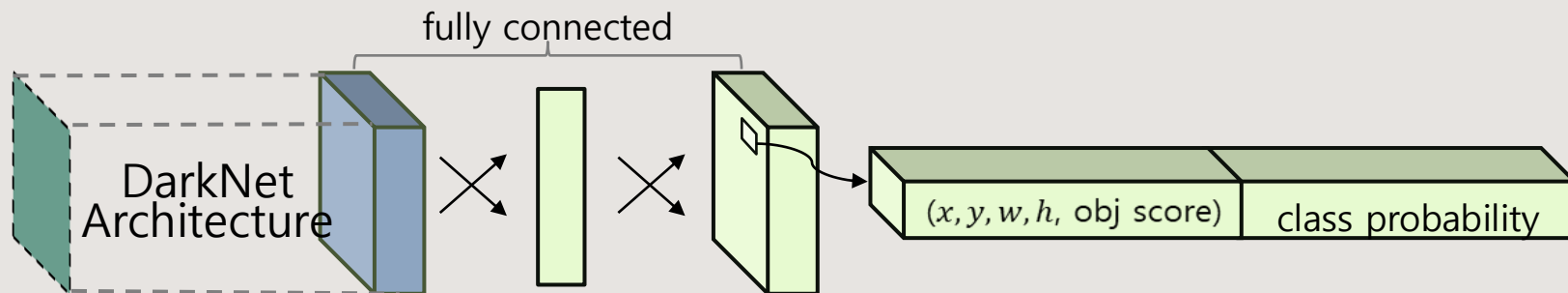


그림 13. 모델 구조

- DarkNet은 23개의 layer를 가짐
- 데이터 전처리
 - Random crop을 통한 scale 변환
 - Mosaic 변환을 통한 데이터 증강
 - HSV 값 변환



그림 14. Random crop 예시

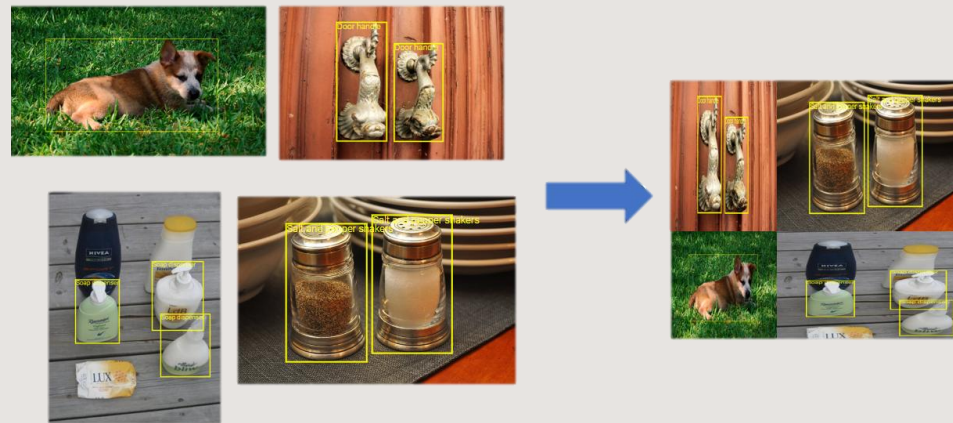


그림 15. Mosaic 예시

모델개발방법

HSV augmentation

비닐봉투와 배경의 색깔이 비슷해 잘 구분되지 않는 샘플, 저조도 환경에서 촬영된 샘플 등의 어려운 케이스들을 잘 학습할 수 있도록 hue (색상), saturation(채도), value(명도) 값을 random하게 변경함

✓ 추가적인 Data를 이용한 augmentation을 수행하여 일반적인 쓰레기 봉투도 검출할 수 있도록 함.



Various Models

실제 적용에서는 기기에 따라 inference 속도의 차이 및 메모리의 한계가 존재할 수 있으므로 여러 type의 모델을 학습하여 small, medium, large 모델의 학습을 진행함.

실험 및 평가

- 평가 지표

[1] Precision: 검출 결과들 중 옳게 검출한 비율

[2] Recall: 실제 옳게 검출된 결과물 중에서 옳다고 예측한 것의 비율

[3] mAP50: Precision-recall 그래프의 아래 면적

[4] FPS: 초당 처리 가능한 프레임 수

*위 지표들의 값이 높을수록 정확함

- 실험환경

- NVIDIA RTX 3090ti

- Pytorch 1.10.1+cu111

- Python 3.9.0

- Class: Trash

- Conf Threshold: 0.3 / IOU Threshold: 0.6

- 여러 모델의 실험

[1] Small, medium, large 모델의 학습 및 실험을 통해 상황과 환경에 맞는 모델을 선정할 수 있도록 함.

[2] Augmentation + 추가 데이터 + Large모델을 대표 best 모델로 선정하여 정성평가를 보여주었으나, 정량평가에서는 다른 모델들도 실험을 진행하여 결과를 확인함.

실험 및 평가-정량평가

모델 사이즈, 모듈, 데이터셋에 따른 경향성 비교

Model size	실험조건		Precision	Recall	mAP50	Params	GFlops
	Augmentation	Additional DB					
Small model	high	X	0.947	0.901	0.942	7,020,913	15.8
Medium model	high	X	0.948	0.938	0.946	20,852,934	47.9
Large model	high	X	0.949	0.926	0.947	46,108,278	107.6
Large model	Medium	X	0.958	0.931	0.948	46,108,278	107.6
Large model	Low	X	0.957	0.929	0.955	46,108,278	107.6
Large model	X	X	0.911	0.915	0.915	46,108,278	107.6
Large model	Medium	O	0.947	0.943	0.948	46,108,278	107.6
Small model	Medium	X	0.95	0.916	0.934	7,020,913	15.8
Small model	Medium	O	0.955	0.901	0.942	7,020,913	15.8

- 경향성을 확인하기 위해 끝까지 실험을 돌리지 않고 300epoch정도의 적은 학습을 진행한 결과이며, HSV augmentation을 주었을 때 small model임에도 불구하고 Large model-no augmentation 보다 좋은 성능을 보임.
- 단, augmentation은 큰 확률 값과 강도를 사용했을 때보다 중간정도의 값들을 사용했을 때의 성능이 더 좋음. 데이터 변화를 강하게 주었을 때 오히려 모델의 학습에 방해하는 것으로 추정.
- 추가적인 db를 사용하였을 때 precision과 mAP가 소폭 상승하는 것을 확인할 수 있었음.

실험 및 평가-정량평가

모델 사이즈, 모듈, 데이터셋에 따른 경향성 비교

Model size	실험조건		Precision	Recall	mAP50	Params	GFlops
	Augmentation	Additional DB					
Small model	high	X	0.947	0.901	0.942	7,020,913	15.8
Medium model	high	X	0.948	0.938	0.946	20,852,934	47.9
Large model	high	X	0.949	0.926	0.947	46,108,278	107.6
Large model	Medium	X	0.958	0.931	0.948	46,108,278	107.6
Large model	Low	X	0.957	0.929	0.955	46,108,278	107.6
Large model	X	X	0.911	0.915	0.915	46,108,278	107.6
Large model	Medium	O	0.947	0.943	0.948	46,108,278	107.6
Small model	Medium	X	0.95	0.916	0.934	7,020,913	15.8
Small model	Medium	O	0.955	0.901	0.942	7,020,913	15.8

- 경향성을 확인하기 위해 끝까지 실험을 돌리지 않고 300epoch정도의 적은 학습을 진행한 결과이며, HSV augmentation을 주었을 때 small model임에도 불구하고 Large model-no augmentation 보다 좋은 성능을 보임.
- 단, augmentation은 큰 확률 값과 강도를 사용했을 때보다 중간정도의 값들을 사용했을 때의 성능이 더 좋음. 데이터 변화를 강하게 주었을 때 오히려 모델의 학습에 방해하는 것으로 추정.
- 추가적인 db를 사용하였을 때 precision과 mAP가 소폭 상승하는 것을 확인할 수 있었음.

실험 및 평가-정량평가

모델 사이즈, 모듈, 데이터셋에 따른 경향성 비교

Model size	실험조건		Precision	Recall	mAP50	Params	GFlops
	Augmentation	Additional DB					
Small model	high	X	0.947	0.901	0.942	7,020,913	15.8
Medium model	high	X	0.948	0.938	0.946	20,852,934	47.9
Large model	high	X	0.949	0.926	0.947	46,108,278	107.6
Large model	Medium	X	0.958	0.931	0.948	46,108,278	107.6
Large model	Low	X	0.957	0.929	0.955	46,108,278	107.6
Large model	X	X	0.911	0.915	0.915	46,108,278	107.6
Large model	Medium	O	0.947	0.943	0.948	46,108,278	107.6
Small model	Medium	X	0.95	0.916	0.934	7,020,913	15.8
Small model	Medium	O	0.955	0.901	0.942	7,020,913	15.8

- 경향성을 확인하기 위해 끝까지 실험을 돌리지 않고 300epoch정도의 적은 학습을 진행한 결과이며, HSV augmentation을 주었을 때 small model임에도 불구하고 Large model-no augmentation 보다 좋은 성능을 보임.
- 단, augmentation은 큰 확률 값과 강도를 사용했을 때보다 중간정도의 값들을 사용했을 때의 성능이 더 좋음. 데이터 변화를 강하게 주었을 때 오히려 모델의 학습에 방해하는 것으로 추정.
- 추가적인 db를 사용하였을 때 precision과 mAP가 소폭 상승하는 것을 확인할 수 있었음.

실험 및 평가-정량평가

모델 사이즈, 모듈, 데이터셋에 따른 경향성 비교

Model size	실험조건		Precision	Recall	mAP50	Params	GFlops
	Augmentation	Additional DB					
Small model	high	X	0.947	0.901	0.942	7,020,913	15.8
Medium model	high	X	0.948	0.938	0.946	20,852,934	47.9
Large model	high	X	0.949	0.926	0.947	46,108,278	107.6
Large model	Medium	X	0.958	0.931	0.948	46,108,278	107.6
Large model	Low	X	0.957	0.929	0.955	46,108,278	107.6
Large model	X	X	0.911	0.915	0.915	46,108,278	107.6
Large model	Medium	O	0.947	0.943	0.948	46,108,278	107.6
Small model	Medium	X	0.95	0.916	0.934	7,020,913	15.8
Small model	Medium	O	0.955	0.901	0.942	7,020,913	15.8

- 경향성을 확인하기 위해 끝까지 실험을 돌리지 않고 300epoch정도의 적은 학습을 진행한 결과이며, HSV augmentation을 주었을 때 small model임에도 불구하고 Large model-no augmentation 보다 좋은 성능을 보임.
- 단, augmentation은 큰 확률 값과 강도를 사용했을 때보다 중간정도의 값들을 사용했을 때의 성능이 더 좋음. 데이터 변화를 강하게 주었을 때 오히려 모델의 학습에 방해하는 것으로 추정.
- 추가적인 db를 사용하였을 때 precision과 mAP가 소폭 상승하는 것을 확인할 수 있었음.

실험 및 평가-최종모델 정량평가

최종 모델 (1000 epoch full training)

Model size	실험조건		Precision	Recall	mAP50	Params	GFlops	FPS
	Augmentation	Additional DB						
Large model	Medium	O	0.987	0.998	0.989	46,108,278	107.6	25.3

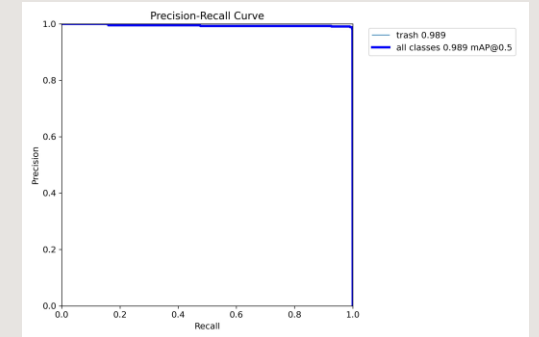


그림 16. Precision-recall curve

```

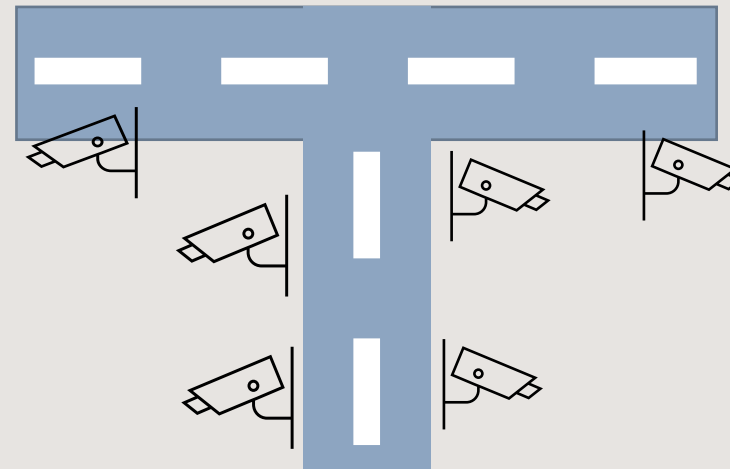
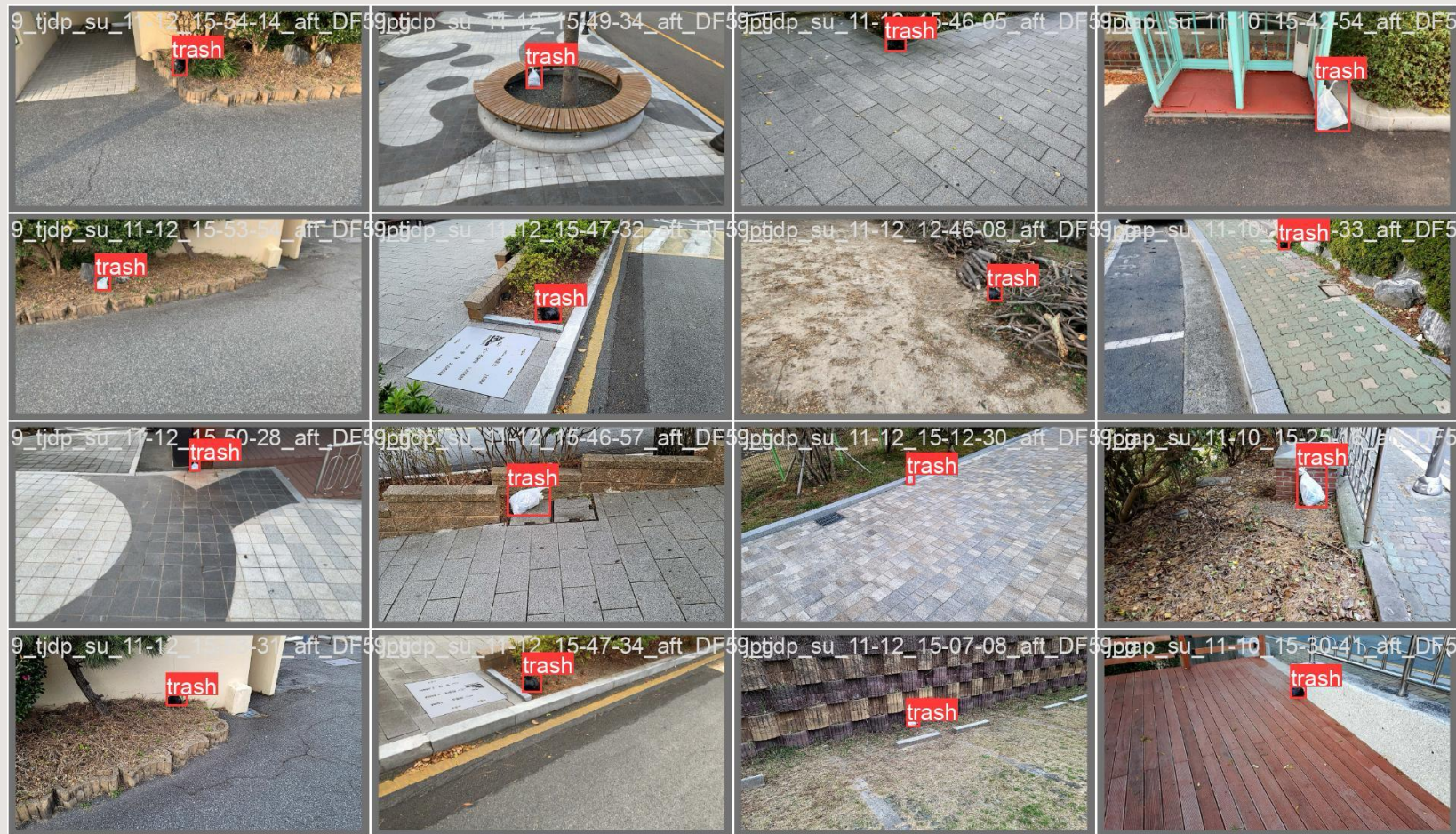
work@main1[30DuHWE3-session]:~/data/Code$ python val.py
/usr/local/lib/python3.10/dist-packages/pandas/core/computation/expressions.py:20: UserWarning: Pandas requires version '2.7.3' or newer of 'numexpr' (version '2.7.0' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
from pandas.core.computation.check import NUMEXPR_INSTALLED
val: data=./data/aicontest.yaml, weights=/home/work/.data/Code/runs/train/weights/best.pt, batch_size=32, imgsiz=640, conf_thres=0.3, iou_thres=0.6, max_det=300, task=val, device=, workers=8, single_cls=False, augment=False, verbose=False, save_txt=False, save_hybrid=False, save_conf=False, save_json=False, project=runs/val, name=exp, exist_ok=False, half=False, dnn=False, valpath=/home/work/.data/dataset/bbox_list/bbox_list_Valid.txt
WARNING ⚠ confidence threshold 0.3 > 0.001 produces invalid results
YOLOv5 🚀 2023-10-6 Python-3.10.6 torch-2.1.0a0+b5021ba CUDA:0 (NVIDIA RTX A6000, 48677MiB)

Fusing layers...
Model summary: 267 layers, 46108278 parameters, 0 gradients, 107.6 GFLOPs
TQ /home/work/.data/dataset/bbox_list/bbox_list_Valid.txt
self.path : /home/work/.data/dataset/bbox_list/bbox_list_Valid.txt
val: Scanning /home/work/.data/dataset/labels/Valid/0_ydsp_cy_11_11_10_44_04_for_DFS_cache' images and labels... 371 found, 0 missing, 3 empty, 0 corrupted: 100% ██████████ | 371/371 [00:00<?, ?it/s]
Class  Images  Instances    P      R    mAP50  mAP50-95  100% ██████████ | 12/12 [00:09<00:00, 1.32it/s]
all      371      433    0.987  0.998    0.989    0.958
Speed: 0.1ms pre-process, 4.3ms inference, 3.4ms NMS per image at shape (32, 3, 640, 640)
    
```

그림 17. 주어진 서버에서 최종모델을 돌린 결과

- 최종 모델인 large-medium augmentation-new dataset은 1000에폭으로 학습하여 모든 평가지표에 대해 최고의 성능을 달성함
- 최종 모델은 1초당 25프레임을 처리할 수 있어 실시간 사용이 가능함. (Nvidia GeForce 3090ti 한대, 배치사이즈 1)

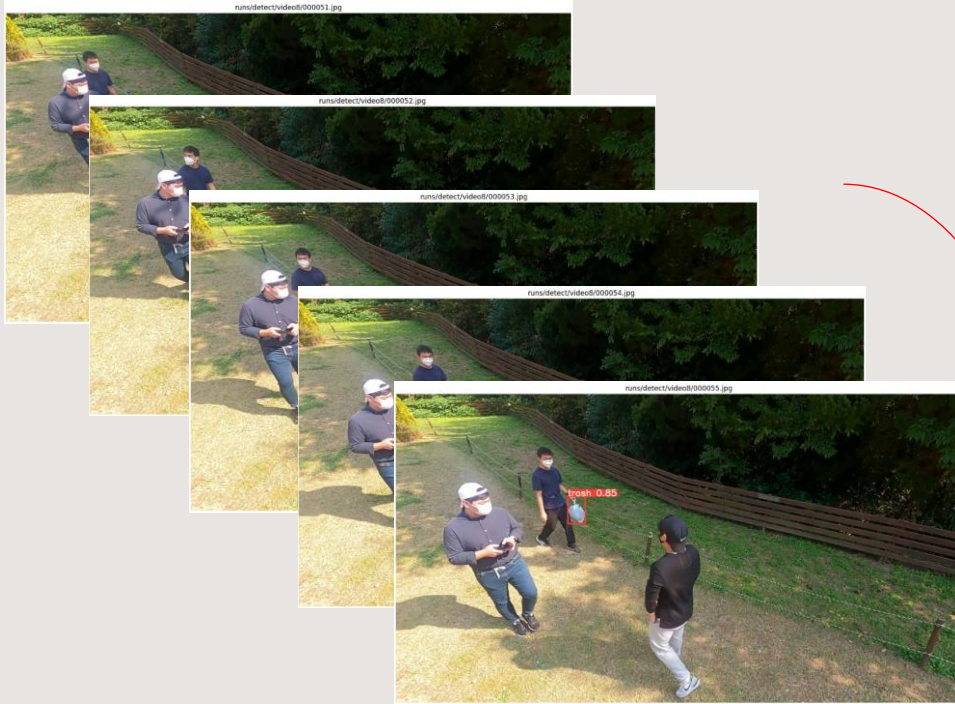
실험 및 평가 - 최종모델 정성평가 (AIHub DB) 다양한 CCTV



- 다양한 형태와 색상의 쓰레기 봉투를 잘 검출함.
- 멀티 스케일 학습 데이터셋을 사용하여 다양한 사이즈의 타겟 물체에 모두 강인하게 작동함. 특히 매우 작은 크기의 쓰레기봉투를 잘 검출함.
- 보도블럭과 같이 패턴을 갖는 배경 때문에 구별이 어려운 케이스도 잘 검출함

실험 및 평가 - 최종모델 정성평가 (Video AIHub DB)

AIHub 공원 주요시설 및 불법행위 감시 CCTV 영상 데이터



연속적인 n개의
Frame에서 동일한 위치에
쓰레기가 검출되는 경우만
무단투기로 간주

쓰레기 무단
투기 집중 지역



- 쓰레기 무단 투기가 자주 일어나는 곳에서의 영상 처리도 상세히 알 수 있음
- Object detection모델의 특성상 쓰레기 봉투가 검출여부만 판단이 가능하기 때문에 연속적인 n개의 frame에서 동일한 위치의 쓰레기가 발견되었을 때만 쓰레기 무단투기라고 판단하여 관리자에게 정보 전달

```
5]: !python detect.py --source="/home/work/.data/dataset/video_example" --name="video" --video=True --frame_num_threshold=20
```

```
/usr/local/lib/python3.10/dist-packages/pandas/core/computation/expressions.py:20: UserWarning: Pandas requires version '2.7.3' or newer of 'numexpr' (version '2.7.0' currently installed).
from pandas.core.computation.check import NUMEXPR_INSTALLED
detect: weights=runs/train/weights/best.pt, source=/home/work/.data/dataset/video_example, data=data/aicontest.yaml, imgsz=[640, 640], conf_thres=0.35, iou_thres=0.7, max_det=1000, device=, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=runs/detect, name=video, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False, vid_stride=1, shown_img_num=4, detect_result=False, video=True, frame_num_threshold=20
requirements: /home/work/.local/lib/python3.10/site-packages/requirements.txt not found, check failed.
YOLOv5 🚀 2023-10-6 Python-3.10.6 torch-2.1.0a0+b5021ba CUDA:0 (NVIDIA RTX A6000, 48677MiB)
```

Fusing layers...

Model summary: 267 layers, 46108278 parameters, 0 gradients, 107.6 GFLOPs

000092.jpg 를 포함한 이전의 연속된 20장의 frame에서 쓰레기 봉투가 검출되었습니다. 쓰레기 무단투기가 의심됩니다.

실험 및 평가 - 최종모델 정성평가 (plastic VNOI DB)



- 새로운 DB에서도 쓰레기 봉투를 잘 찾는 모습을 확인할 수 있음.
- 이에 다른 환경의 이미지에서도 모델이 성공적으로 trash detection이 진행되는 것을 알 수 있음.
- 다양한 해상도 및 저조도 환경에서도 쓰레기 봉투가 detection되는 것을 확인

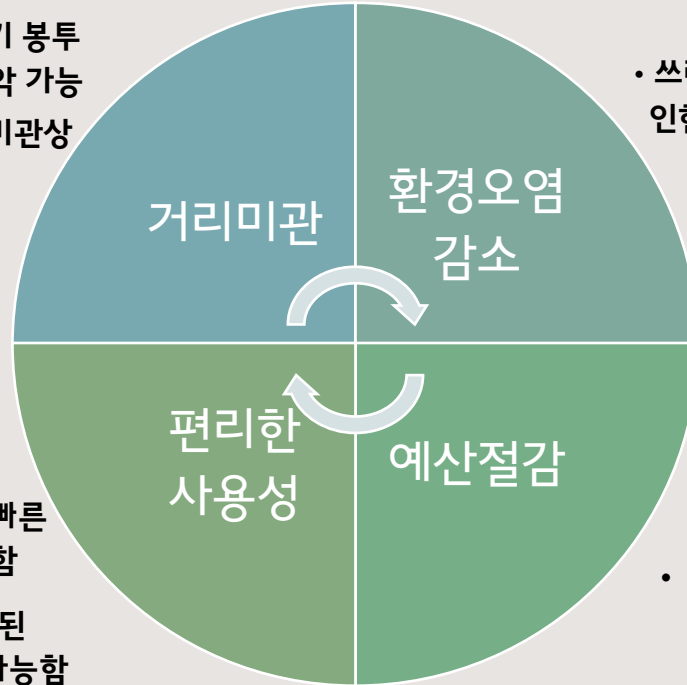
요약

- ❖ 객체 검출 네트워크에 기반한 쓰레기 무단 투기 검출기를 제안함.
- ❖ 데이터셋이 객체 검출기에 적합하게끔 라벨의 형식을 바꿔 사용함.
- ❖ 기존의 네트워크에 HSV augmentation과 같은 데이터 증강 방법을 도입하여 검출이 어려운 환경의 데이터에도 강인하게 작동할 수 있도록 함.
- ❖ 제공된 데이터셋은 실세계에서 무단 투기되는 쓰레기 봉투의 형태의 다양성을 반영하지 않음. 따라서 이 한계를 보완하기 위해 추가 데이터를 수집하여 학습에 추가로 사용함.
- ❖ 추가로 사용한 데이터는 제공된 데이터보다 다양한 해상도의 이미지로 구성되어 있으며, 타겟 물체의 크기와 그 색상과 형태가 다양함.
- ❖ 사용성을 고려하여 가장 성능이 낮으면서 속도가 빠르며 가벼운 small 모델부터 좋은 성능과 속도를 보이는 medium 모델, 그리고 가장 성능이 좋으며 속도가 느린 large 모델을 비교 실험함.
- ❖ 정량적 결과에는 검출 정확도 뿐만 아니라 모델 속도와 파라미터 개수도 포함함. 추후 실제 CCTV와 같은 디바이스에 탑재할 경우 해당 기기의 스펙에 맞는 모델을 선택하여 사용할 수 있음.

활용 및 기대효과



- 제안한 네트워크를 통해 쓰레기 봉투의 무단투기의 유무와 위치 파악 가능
- 쓰레기 무단투기로 발생하는 미관상의 문제 해결



- 쓰레기 무단 투기 및 미분리 배출로 인한 폐기물 감축



- 여러 상황에 맞는 모델을 제안하여 가볍고 빠른 모델부터 큰 모델까지 빠른 속도를 보이며, 실시간 검출이 가능함
- 새로운 데이터 추가 수집 시 기 학습된 모델을 추가 학습하여 업데이트가 가능함
- 동영상 데이터가 아닌 사진 data만으로도 확인이 가능함.

- 무단으로 투기되는 쓰레기 처리로 인하여 매년 소요되는 쓰레기 처리 비용 약 3400억 등의 예산 절감 효과



감사합니다

Q&A

