

R-CNN 기반 Segmentation 치료 협동로봇(Collaborative Robot)

- 팀장 : 박주환
- 팀원 : 장성혁, 오세현

RECS Lab.
Robotics and Embedded Control Systems

1. 프로젝트 개요

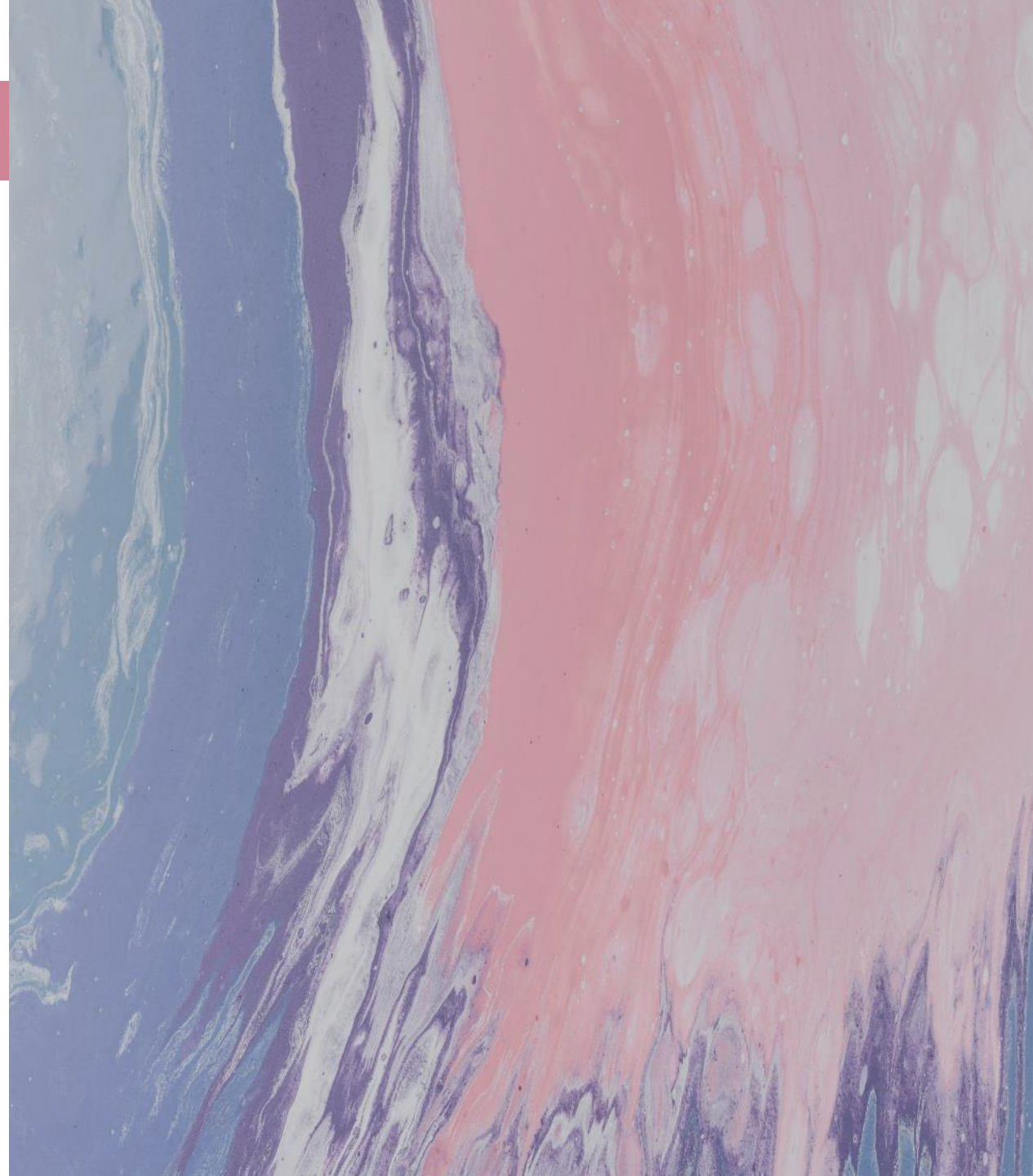
- 프로젝트 선정배경
- 프로젝트 목표

2. 프로젝트 소개 및 단계별 진행과정

- 협동로봇 소개
- 사용 알고리즘
- 모델 데이터
- 프로젝트 흐름도
- 데이터 전처리 과정
- 데이터 학습
- 학습된 데이터로 새로운 이미지 예측

3. 기대효과 및 느낀 점

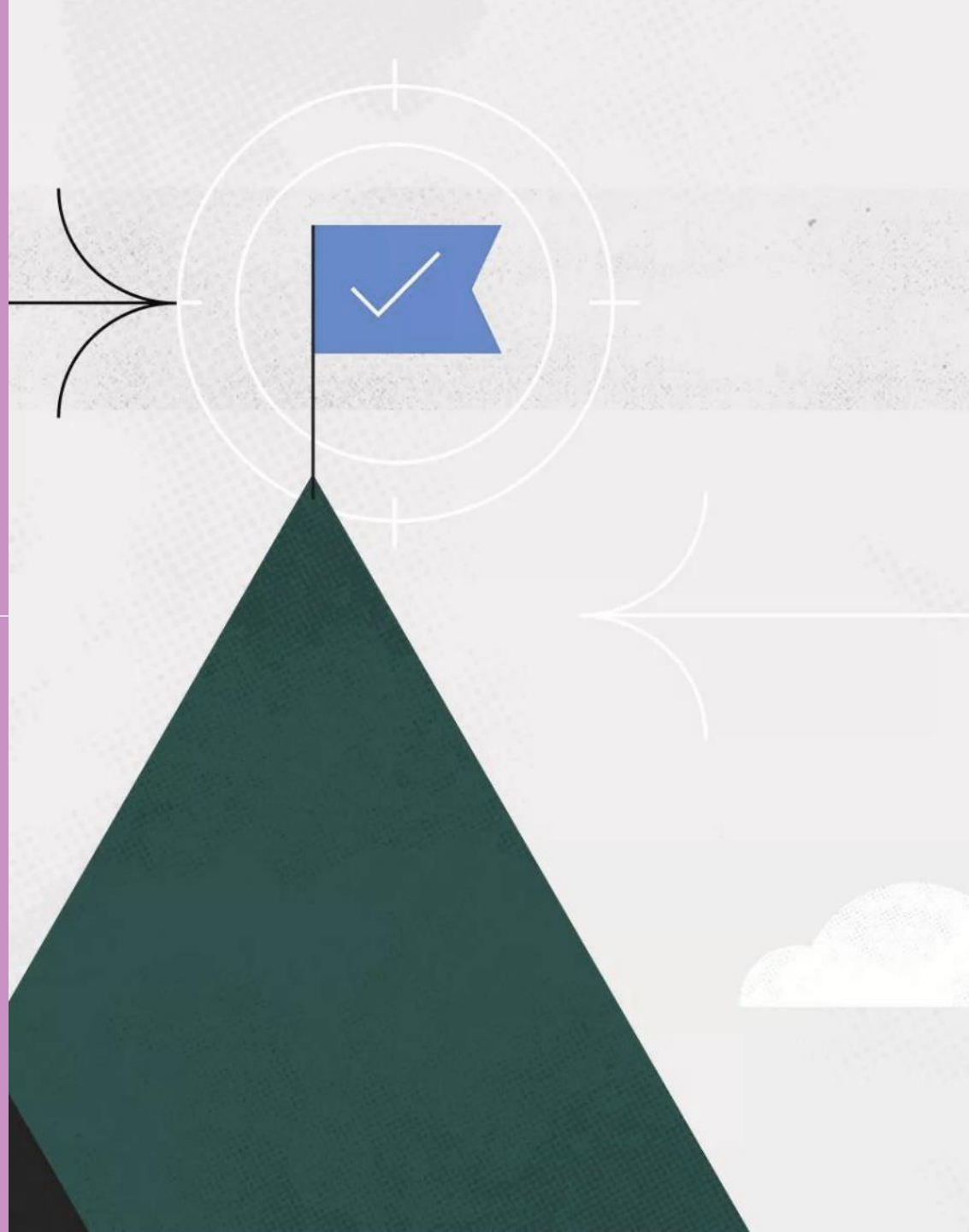
- 기대효과
- 프로젝트를 진행하며 느낀 점



Part 1,

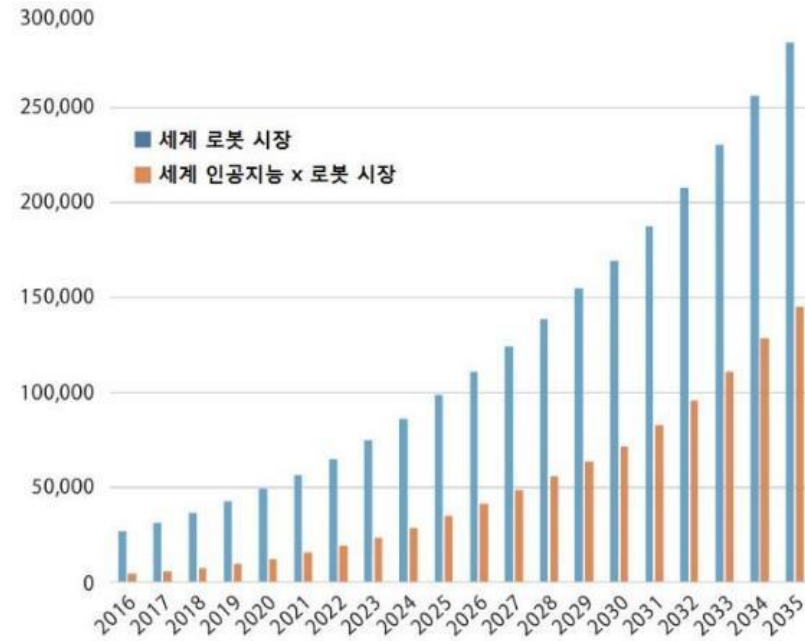
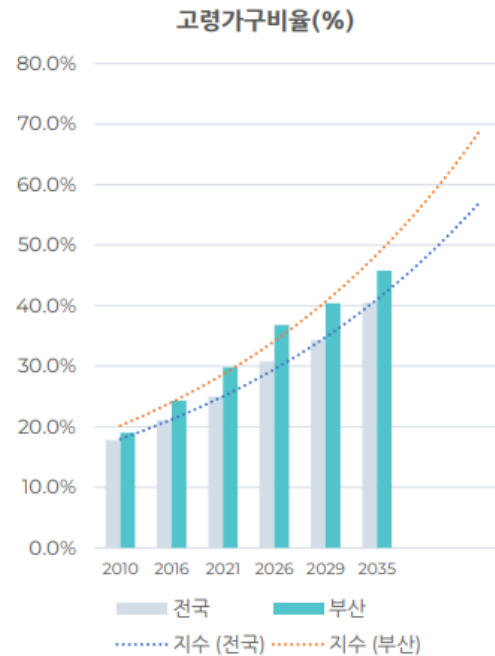
프로젝트 개요

- 프로젝트 선정배경
- 프로젝트 목표

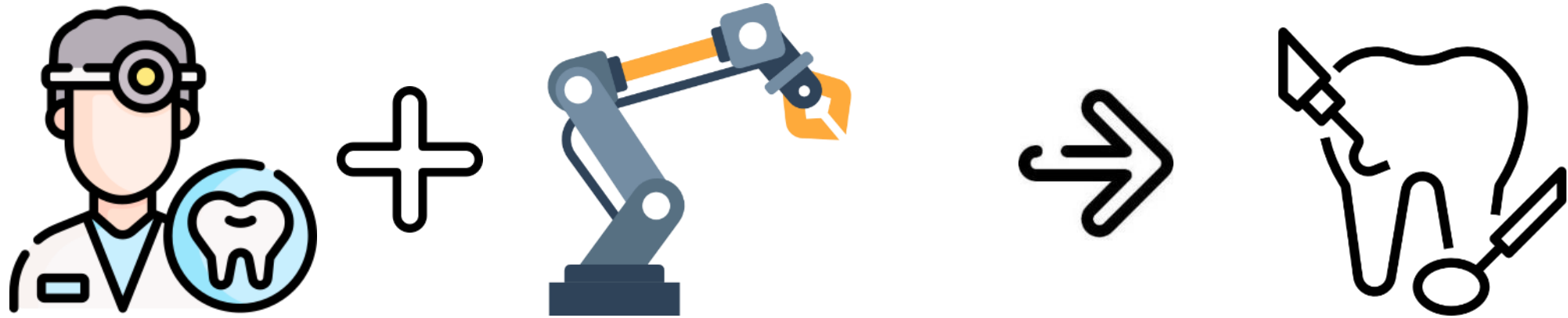


AI(인공지능)로 구강검진 부산 스타트업 기술에 베트남도 반했다

‘큐티티’서 개발 ‘이아포’ 솔루션, 호찌민 의료전시회 소개돼 호평



- 전 세계 디지털 헬스케어 시장의 **급격한 규모 증가**
- 여러 산업분야에 이어 서비스 시장까지 열리면서 **협동로봇 시장은 가파른 성장세가 예측됨**
- 광범위해진 로봇의 사용 가능성으로 인해 **다양한 산업에서 로봇에 대한 수요가 늘고 있음**
- 고령화에 따른 인력수급 부족과 무인화, 자동화로 인한 **개발 가능성 증가**



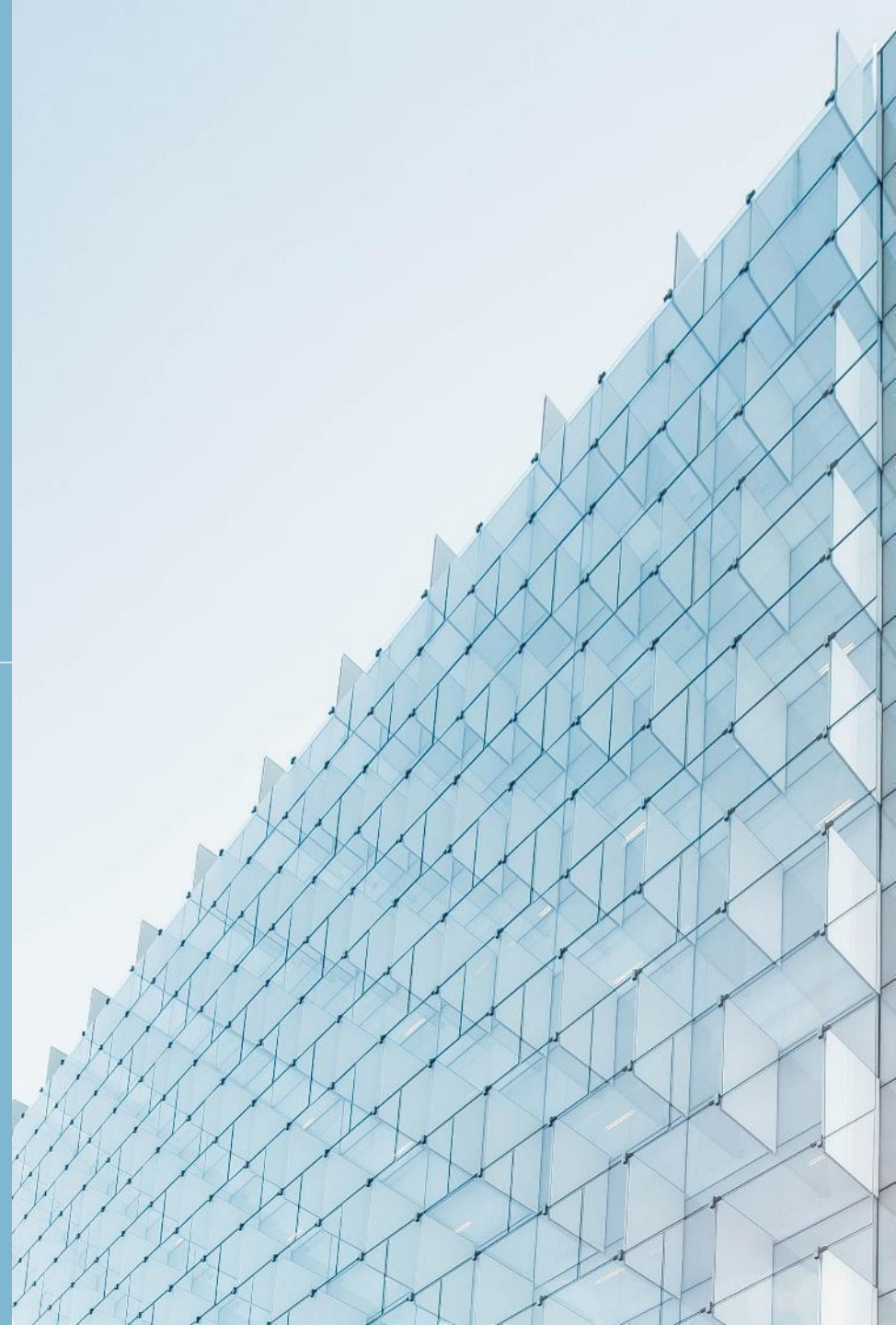
구현계획

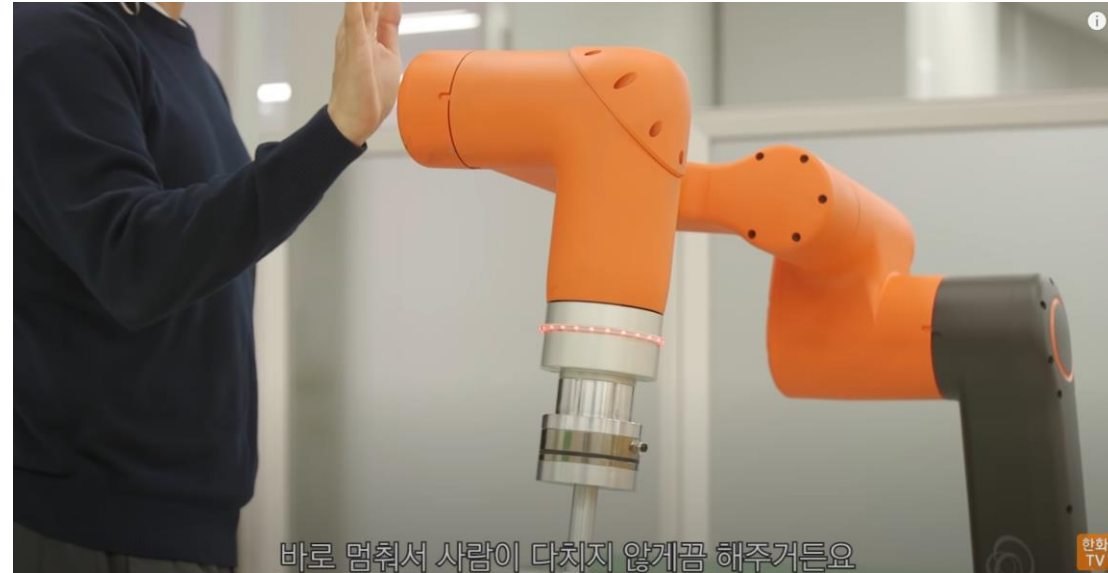
- 제공받은 치아 데이터를 파이썬 자동화 코드를 통해 분류함
- 치아 데이터셋을 딥러닝으로 학습 후 검출 과정을 통해 초기 치아 좌표 추출
- 받은 좌표로 이동 후 카메라를 통해 치아의 정확한 위치와 각도를 계산하여 세부 위치 전달
- 전달받은 좌표 값으로 협동로봇(Cobot)가 움직여 의사의 치료를 보조함

Part 2,

프로젝트 소개 및 단계별 진행과정

- 협동로봇 소개
- 사용 알고리즘
- 모델 데이터
- 프로젝트 흐름도
- 데이터 전처리 과정
- 데이터 학습
- 학습된 데이터로 새로운 이미지 예측





About - cobot

- 인간과의 직접적인 상호 작용을 위해 설계된 로봇
 - 일반 로봇은 다소 자율적으로 움직이도록 만들어짐
- ↓
- 협동로봇은 사람이 어떤 작업을 성공적으로 수행할 수 있게 도움

Part 2, 데이터 전처리 과정

- QTT측에서 제공받은 데이터를 1차적으로 분류하여 a(치아우식), b(치석, 보철물), c(치아번호)로 폴더를 분리함
- 1차 데이터 中 a를 선택 후 train / val / test 로 분리함



Part 2, Labelme convert to coco format

- 학습을 위해 labelme 데이터 셋을 coco 형식으로 변환하였음

ouuput > a091.json

이름

001_00095_01_a091

001_00097_05_a091

001_00100_01_a091

001_00100_02_a091

001_00100_03_a091

001_00100_05_a091

001_00100_06_a091

001_00100_07_a091

001_00100_10_a091

001_00100_11_a091

001_00100_12_a091

001_00106_01_a091

001_00106_02_a091

001_00106_03_a091

001_00106_04_a091

001_00106_05_a091

001_00106_07_a091

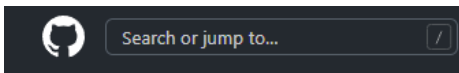
001_00106_08_a091

001_00095_01_a091 - Windows 메모장

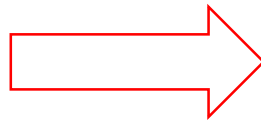
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
{
  "version": "5.0.2",
  "flags": {},
  "shapes": [
    {
      "label": "GI1",
      "points": [
        [
          106.7037037037037,
          204.18518518518516
        ],
        [
          106.7037037037037,
          222.7037037037037
        ],
        [
          130.28571428571428,
          228.57142857142858
        ],
        [
          135.8979591836735,
          219.3877551020408
        ],
        [
          157.83673469387753,
          221.9387755102041
        ],
        [
          176.71428571428572,
          224.48979591836735
        ],
        [
          185.8979591836735,
          204.18518518518516
        ]
      ]
    }
  ]
}
```

<labelme>



fcakyon / labelme2coco Public
forked from XCRobot/Labelme2Coco

[illegible]

test	2022-09-30 오전 12:59	JSON File
train	2022-09-30 오전 12:58	JSON File
val	2022-09-30 오전 12:59	JSON File

<coco>



Detectron2

Detectron2 란?

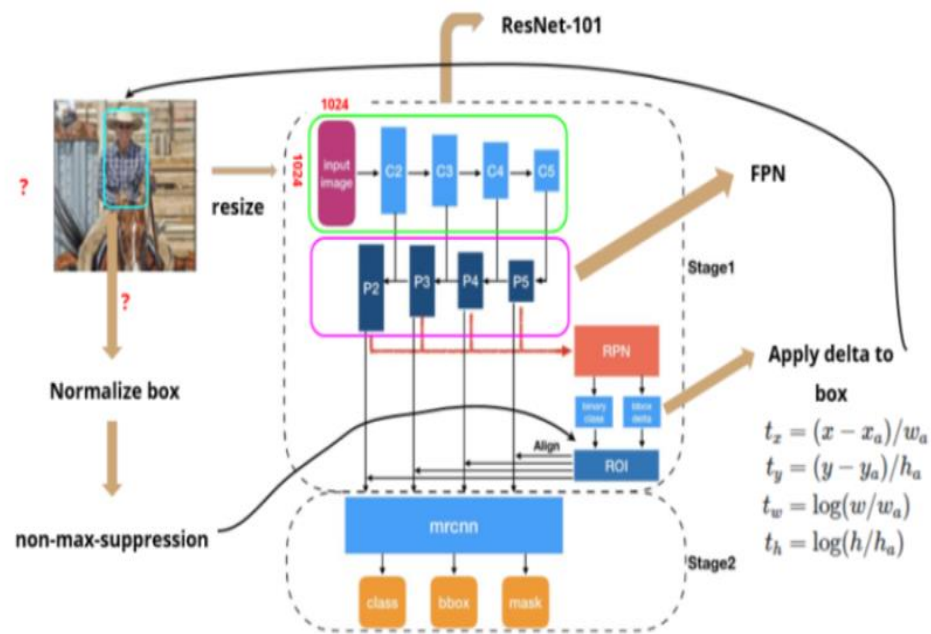
- Facebook(현 Meta) 인공지능 연구자들이 만든 pytorch 기반 object detection과 semantic segemanation을 위한 training inference 플랫폼



The Meta logo, consisting of a blue infinity symbol followed by the word "Meta" in a dark blue sans-serif font.

Part 2, 사용 알고리즘 (Mask R-CNN)

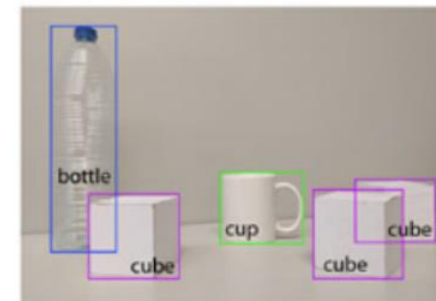
- Fast R-CNN에서 Instance Segmentation이 가능하도록 Mask Branch를 추가한 모델임
- ROI-Align 방법을 통해 픽셀 연산이 가능하고 이를 통해 Segmentation을 표현할 수 있음



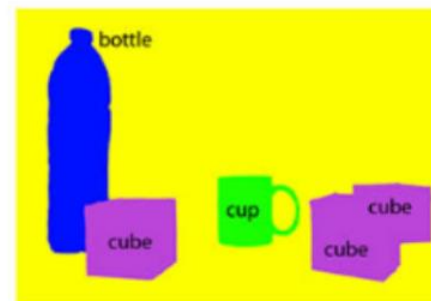
<Mask R-CNN architecture>



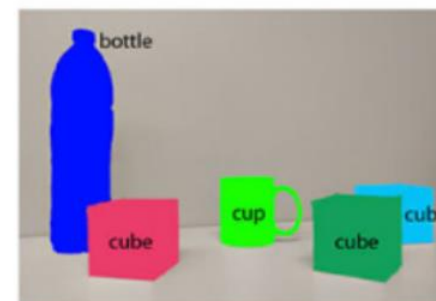
(a) Image classification



(b) Object localization



(c) Semantic segmentation



(d) Instance segmentation

Part 2, 데이터 학습

- detectron2 라이브러리를 이용하여 이미지와 json 파일을 학습하여 모델 데이터를 추출하는 코드를 생성함

```
class MyTrainer(DefaultTrainer):
    @classmethod
    def build_evaluator(cls, cfg, dataset_name, output_folder=None):
        if output_folder is None:
            output_folder = os.path.join(cfg.OUTPUT_DIR, "inference")
        return COCOEvaluator(dataset_name, cfg, True, output_folder)

    def build_hooks(self):
        hooks = super().build_hooks()
        hooks.insert(-1, LossEvalHook(
            cfg.TEST.EVAL_PERIOD,
            self.model,
            build_detection_test_loader(
                self.cfg,
                self.cfg.DATASETS.TEST[0],
                DatasetMapper(self.cfg, True)
            )
        ))
        return hooks

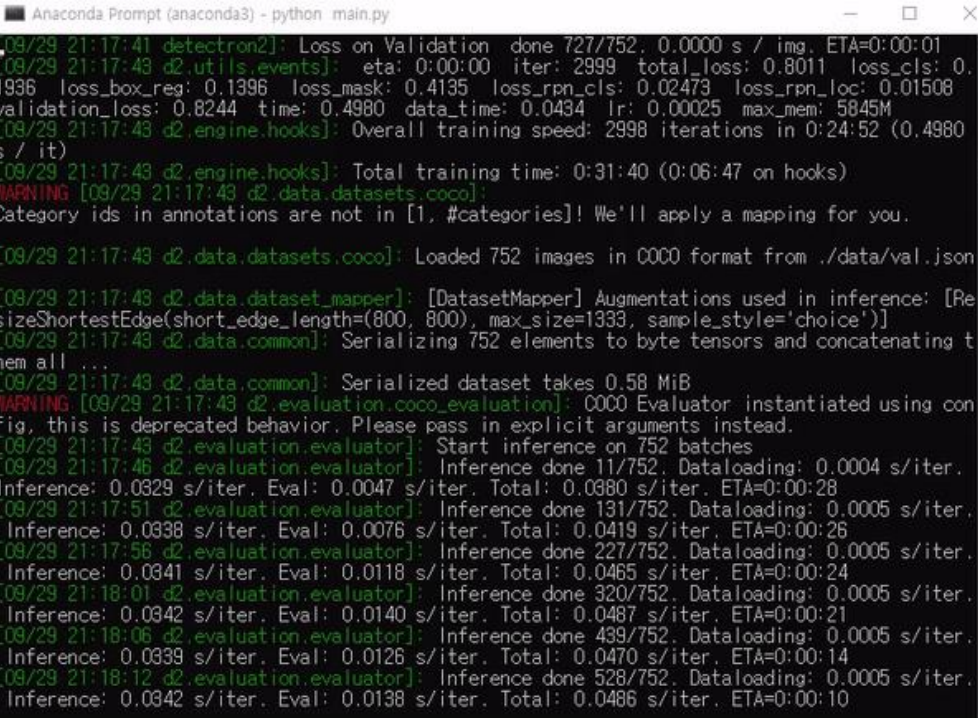
if __name__ == '__main__':
    run()

register_coco_instances("train", {}, "./data/train.json", "./data/train")
register_coco_instances("val", {}, "./data/val.json", "./data/val")
register_coco_instances("test", {}, "./data/test.json", "./data/test")

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")) #mask_rcnn_R_50_FPN_3x.yaml
cfg.DATASETS.TRAIN = ("train",)
cfg.DATASETS.TEST = ("val",)
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 8
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 3000
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 10
cfg.TEST.EVAL_PERIOD = 500

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = MyTrainer(cfg) #DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

<main.py>



```
Anaconda Prompt (anaconda3) - python main.py
09/29 21:17:41 detectron2]: Loss on Validation done 727/752. 0.0000 s / img. ETA=0:00:01
09/29 21:17:43 d2.utils.events]: eta: 0:00:00 iter: 2999 total_loss: 0.8011 loss_cls: 0.
996 loss_box_reg: 0.1396 loss_mask: 0.4135 loss_rpn_cls: 0.02473 loss_rpn_loc: 0.01508
validation_loss: 0.8244 time: 0.4980 data_time: 0.0434 lr: 0.00025 max_mem: 5845M
09/29 21:17:43 d2.engine.hooks]: Overall training speed: 2998 iterations in 0:24:52 (0.4980
s / it)
09/29 21:17:43 d2.engine.hooks]: Total training time: 0:31:40 (0:06:47 on hooks)
WARNING [09/29 21:17:43 d2.data.datasets.coco]:
Category ids in annotations are not in [1, #categories]! We'll apply a mapping for you.
09/29 21:17:43 d2.data.datasets.coco]: Loaded 752 images in COCO format from ./data/val.json
09/29 21:17:43 d2.data.dataset_mapper]: [DatasetMapper] Augmentations used in inference: [Re
sizeShortestEdge(short_edge_length=(800, 800), max_size=1333, sample_style='choice')]
09/29 21:17:43 d2.data.common]: Serializing 752 elements to byte tensors and concatenating t
hem all ...
09/29 21:17:43 d2.data.common]: Serialized dataset takes 0.58 MiB
WARNING [09/29 21:17:43 d2.evaluation.coco_evaluation]: COCO Evaluator instantiated using con
fig, this is deprecated behavior. Please pass in explicit arguments instead.
09/29 21:17:43 d2.evaluation.evaluator]: Start inference on 752 batches
09/29 21:17:46 d2.evaluation.evaluator]: Inference done 11/752. Dataloading: 0.0004 s/iter.
Inference: 0.0329 s/iter. Eval: 0.0047 s/iter. Total: 0.0380 s/iter. ETA=0:00:28
09/29 21:17:51 d2.evaluation.evaluator]: Inference done 131/752. Dataloading: 0.0005 s/iter.
Inference: 0.0338 s/iter. Eval: 0.0076 s/iter. Total: 0.0419 s/iter. ETA=0:00:26
09/29 21:17:56 d2.evaluation.evaluator]: Inference done 227/752. Dataloading: 0.0005 s/iter.
Inference: 0.0341 s/iter. Eval: 0.0118 s/iter. Total: 0.0465 s/iter. ETA=0:00:24
09/29 21:18:01 d2.evaluation.evaluator]: Inference done 320/752. Dataloading: 0.0005 s/iter.
Inference: 0.0342 s/iter. Eval: 0.0140 s/iter. Total: 0.0487 s/iter. ETA=0:00:21
09/29 21:18:06 d2.evaluation.evaluator]: Inference done 439/752. Dataloading: 0.0005 s/iter.
Inference: 0.0339 s/iter. Eval: 0.0126 s/iter. Total: 0.0470 s/iter. ETA=0:00:14
09/29 21:18:12 d2.evaluation.evaluator]: Inference done 528/752. Dataloading: 0.0005 s/iter.
Inference: 0.0342 s/iter. Eval: 0.0138 s/iter. Total: 0.0486 s/iter. ETA=0:00:10
```

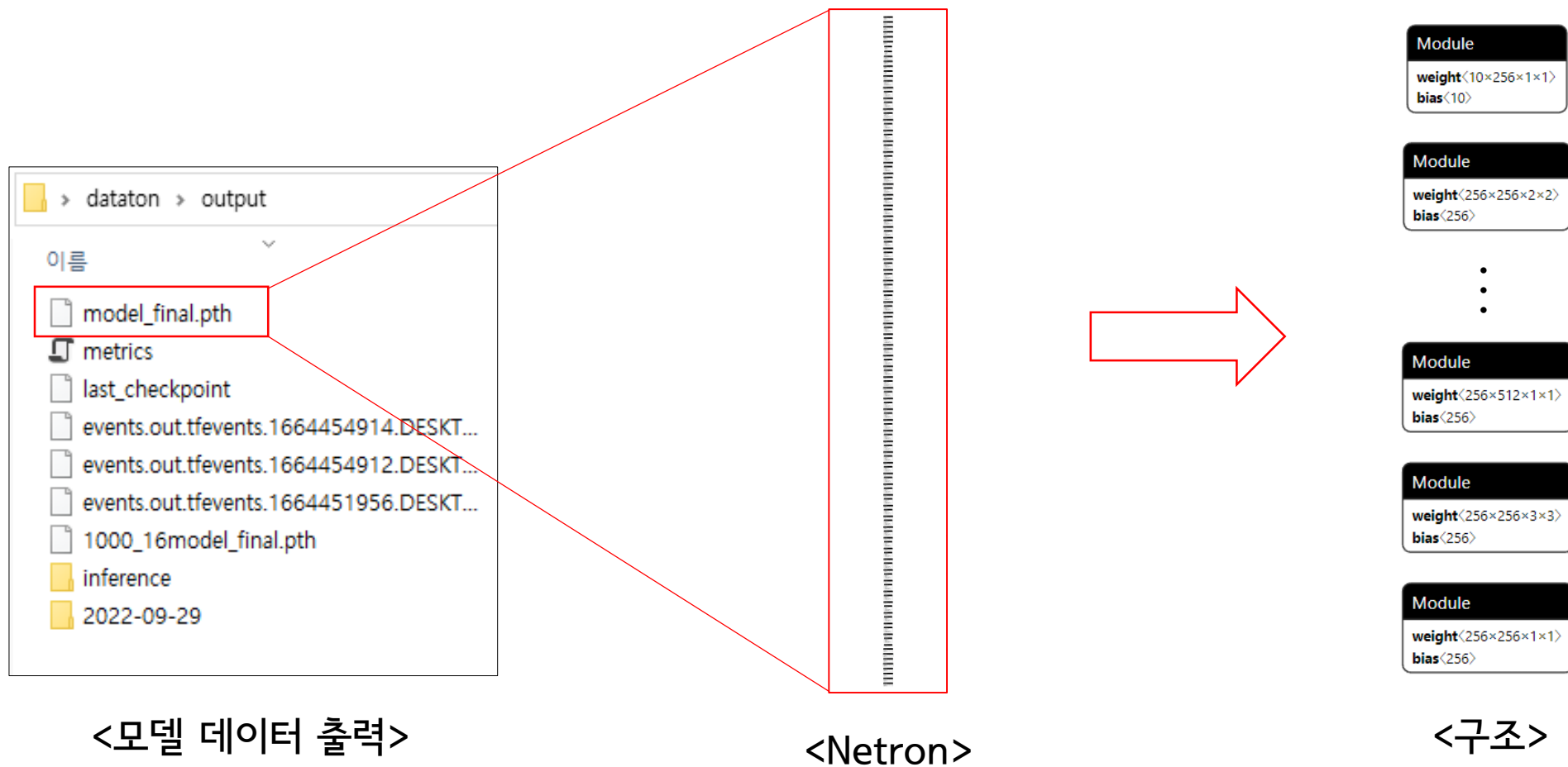
GPU

<training>

NVIDIA GeForce
RTX 3080Ti

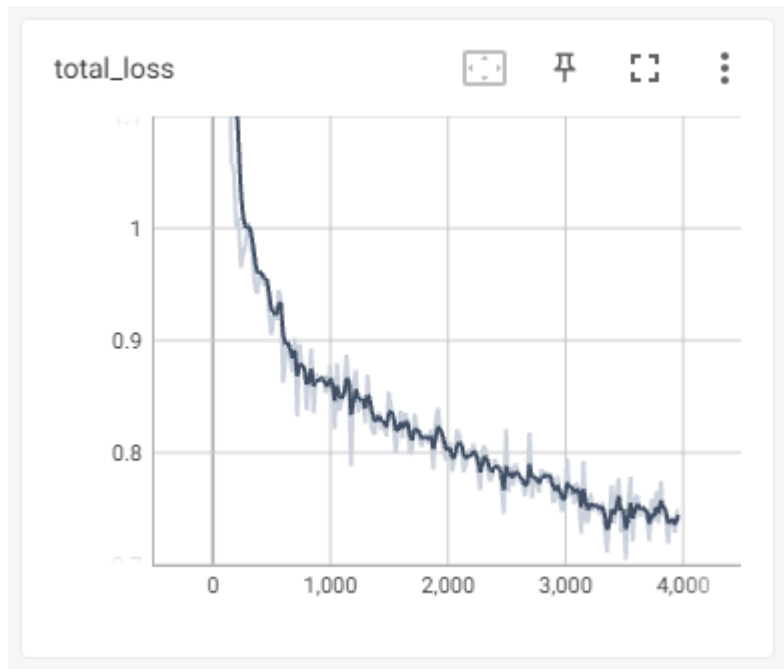
Part 2, 모델 데이터

- 학습 결과 output 폴더에 모델 파일(model_final.pth)이 생성됨
- Netron 사이트를 통해 네트워크 모델을 시각화함으로써 구조를 확인함

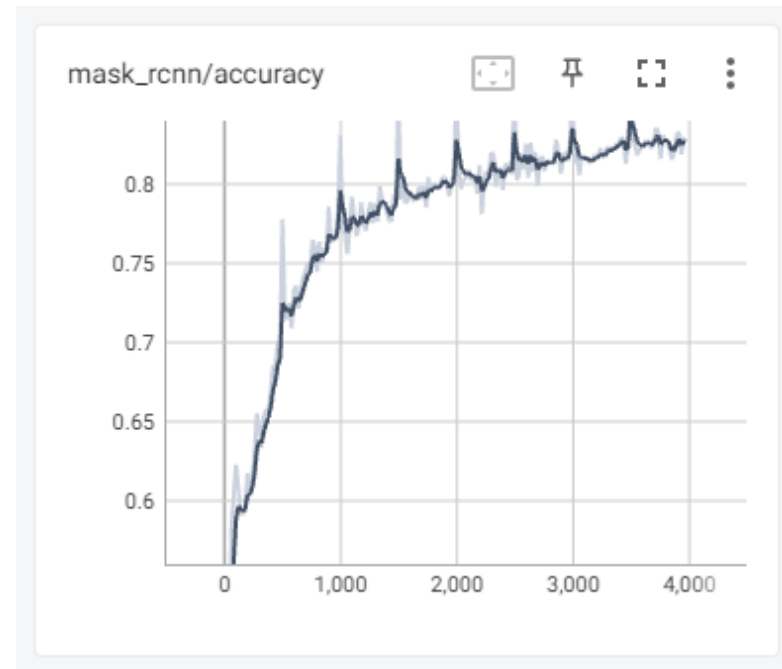


Part 2, 라이브러리를 통한 학습과정 그래프 검출

- 텐서보드를 통해 학습 과정을 그래프로 확인



<Total_loss가 점점 떨어지는 것을 확인>



<Accuracy는 점점 증가하는 것을 확인>

Part 2, 학습된 모델로 새로운 이미지 예측

- 모델 데이터로 바탕으로 입력 이미지에 대해 객체 분류를 진행하는 코드를 생성함

```
hehe - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg
from detectron2 import model_zoo
from detectron2.data.datasets import register_coco_instances
from detectron2.engine import DefaultPredictor
import os, cv2
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog, DatasetCatalog
from detectron2.utils.visualizer import ColorMode
import matplotlib.pyplot as plt

cfg = get_cfg()
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "1000_16model_final.pth")
cfg.DATASETS.TEST = ("my_dataset_test",)
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.7 # set the testing threshold for this model
predictor = DefaultPredictor(cfg)
test_metadata = MetadataCatalog.get("my_dataset_test")

from detectron2.utils.visualizer import ColorMode
import glob

for imageName in glob.glob('./data/test/JPEGImages/*.jpg'):
    im = cv2.imread(imageName)
    outputs = predictor(im)["panoptic_seg"]
    v = Visualizer(im[:, :, ::-1],
                  metadata=test_metadata,
                  scale=0.5
    )
    out = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    plt.imshow(out.get_image())
    plt.show()
```

detect.py

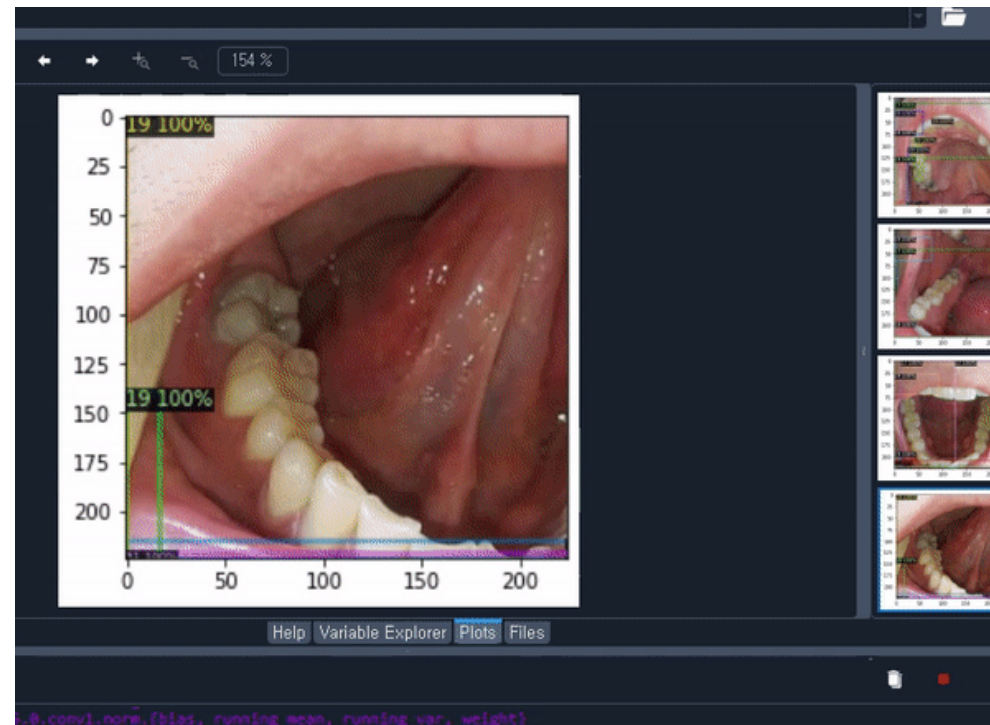
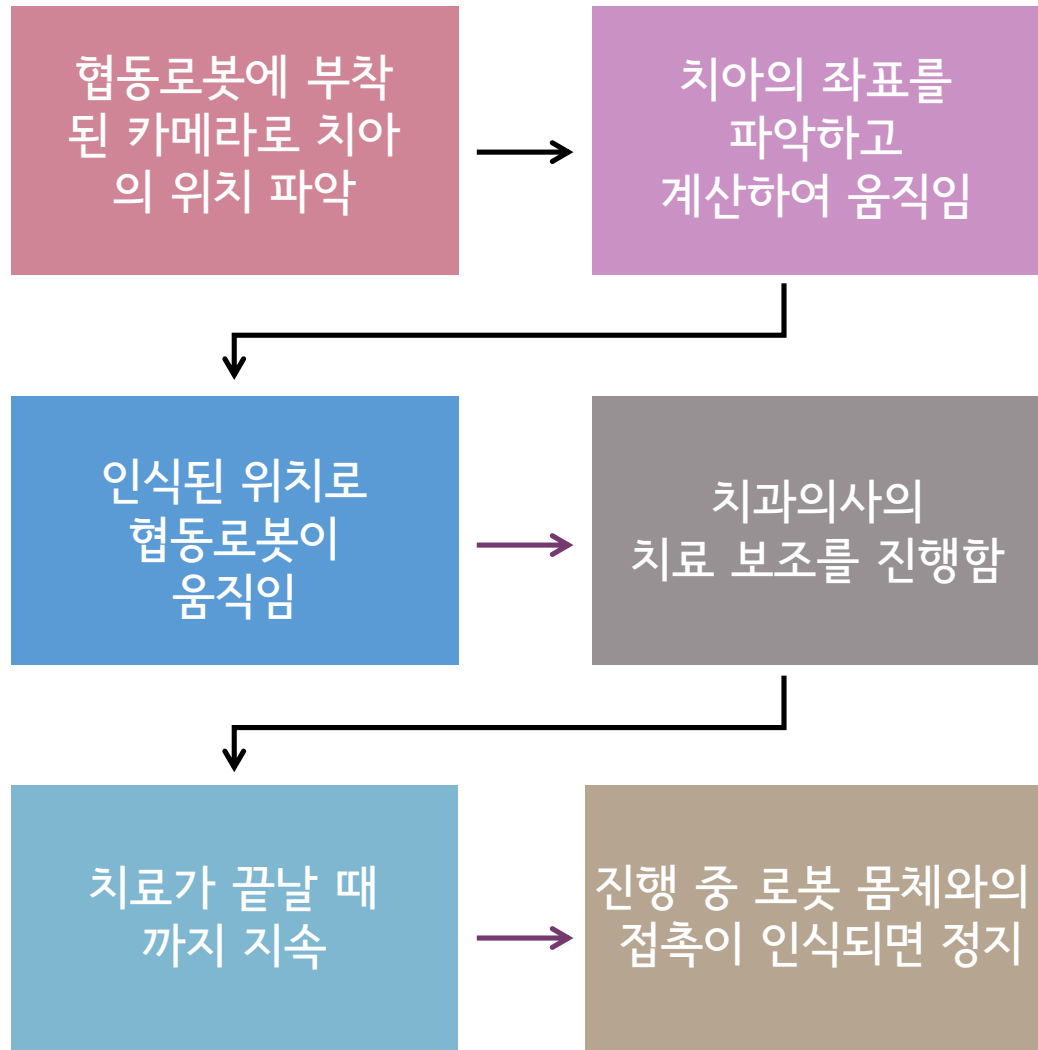


image
classification

Part 2,
프로젝트 흐름도



<참고영상>



<https://youtu.be/PCdcTgE3kDA>

Part 3,

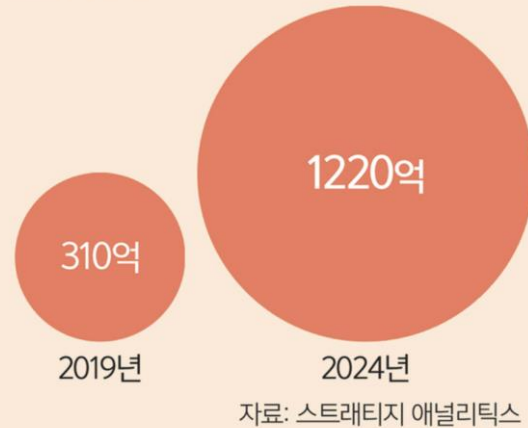
기대효과 및 느낀 점

- 기대효과
- 프로젝트를 진행하며 느낀 점



전세계 로봇시장 규모

(단위: 달러)



주요 기업들의 로봇 사업

삼성	사내 로봇 TF '로봇 사업화' 상설조직 로봇사업팀으로 격상. 인터랙션 로봇 '삼성 봇 아이', 가사보조 로봇 '삼성 봇 핸디', 보행보조 로봇 '젠크스', 음식서빙 로봇 '삼성봇 서빙' 개발
현대차	소프트뱅크로부터 보스턴다이나믹스 인수. 4족보행 로봇개 '스팟', 직립보행이 가능한 휴머노이드 로봇 '아틀라스' 개발
LG	2018년 로봇사업센터로 로봇사업 통합. 산업용 로봇 개발·제조기업 로보스타 등 인수. LG 자사 로봇 '클로이'를 청소봇, 가이드봇, 살균봇, 바리스타봇 등으로 적극 활용
KT	'AI 방역로봇' 선보여. 'AI호텔로봇' 'AI케어로봇' '바리스타로봇' 등 로봇 서비스 플랫폼 확장

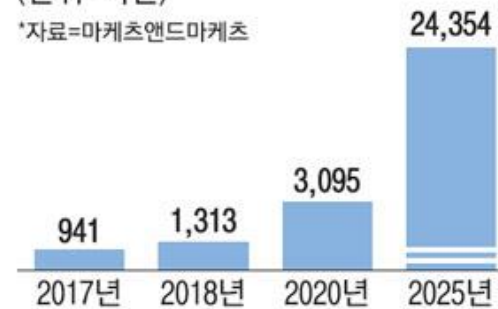
자료: 업계 종합

예상되는 효과

- 로봇시장 가속화 및 상용화 증대
- 국내 AI 헬스케어 시장 규모의 증대

국내 AI 헬스케어 시장 규모 (단위=억원)

*자료=마켓츠앤드마켓츠



- 디지털 생태계를 향한 스마트시티 프로젝트와 정부 투자 확대
로 이어져 더 다양한 종류의 로봇들이 개발되고 상용화될 것임
- 고령화 시대에 따른 라이프 케어의 중요성이 높아지면서 건강
관리를 위한 R&D 분야에 집중 투자 증대

느낀 점

- **박주환** : R-CNN 에 대해서는 알고 있었지만 이를 파인튜닝하여 다른 프로그램과 연동하는 과정은 처음이었다. 구현하는 과정에서 시행착오가 있어서 목표에 완벽하게 도달하지는 못하였지만 AI학습에 대한 방향성을 잡아본 좋은 경험이었다.
- **장성혁** : 처음 주제를 받았을 때에는 3일안에 어떻게 진행하면 좋을지 막막하였지만, 짧은 시간 많은 데이터와 인공지능 학습, 라벨링 등 새로운 지식을 쌓을 수 있었던 잊지못할 경험이 될 것 같다.
- **오세현** : 프로젝트를 진행하면서, detection2를 위해 아나콘다, opencv등 새로운 지식을 쌓을 수 있었습니다. 시간이 좀 더 있다면 인식범위를 늘려 다른 프로젝트도 해보고 싶습니다.

애로사항 및 추후계획

- 추후 협동 로봇 뿐만 아닌 다양한 로봇에 인공지능 알고리즘을 적용하여 연구를 진행해보고 싶다.
- 깃 허브에서 인공지능 알고리즘을 가져와서 사용하는 과정에서 Fine tuning할 때 세부적인 프로그램 이해가 부족하여 수정에 어려움이 있었음
- 로봇과 학생들이 주 구성원이다 보니, 인공지능에 대한 이해도가 부족했는데 데이터톤을 통해 한 층 더 자신감을 가질 수 있게 되었음



감사합니다!

Q&A

