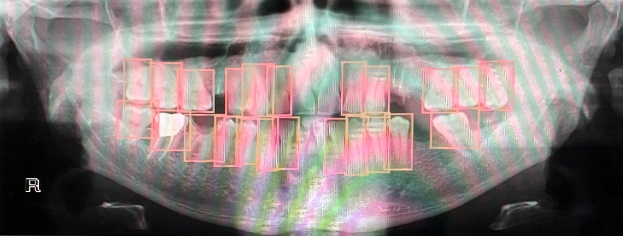
1. Yolo로 개별 치아 crop하기

**Install**

|  |
| --- |
| %cd yourpath/Team1/yolov5  !pip install -r requirements.txt |

**Usage**

1. **개별 치아로 라벨링된 데이터로 yolo 학습하기**



개별 치아로 라벨링된 이미지 (/yolov5/data/panorama)

yolov5/data/coco128.yaml에 있는 train, val 및 test 파일 경로를 수정해야합니다.

또한 train.txt 및 val.txt 안에 있는 데이터 경로를 수정해야합니다.

|  |
| --- |
| import torch  import glob  %cat yourpath/Team1/yolov5/data/coco128.yaml  os.chdir(‘yourpath/Team1/yolov5’)  !python train.py --img 416 --batch 16 --epochs 50 –data /yourpath/Team1/yolov5/data/coco128.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name your\_results\_name |

1. **학습된 yolo로 입 전체 X-ray 이미지 crop하기**

학습된 모델로 이미지(yolov5/data/unlabeled\_jpg\_te)를 개별 치아로 crop해줍니다.

crop된 치아는 yolov5/data/teethCrop폴더에 저장됩니다.

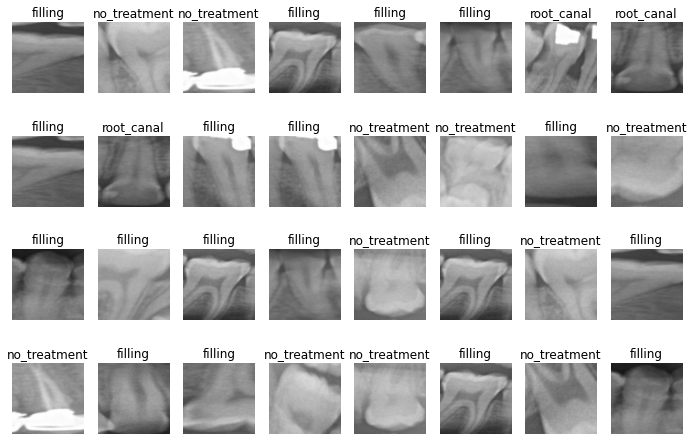
|  |
| --- |
| from crop\_img import crop  path = ‘yourpath/Team1’ #ex)C:/Users/Team1  model = torch.hub.load('.', 'custom', path = path+'/yolov5/runs/train/ your\_results\_name/weights/best.pt', source='local')  img\_list = glob.glob(path+'/yolov5/data/unlabeled\_jpg\_te/\*.jpg')  crop(path, img\_list, model) |
|  |

1. Resnet으로 치아 우식 분류 학습하기

**Usage**

1. **augmentain**

우선 치아 우식을 라벨링하고, 라벨링된 폴더의 데이터를 8:2 비율로 train data set (/resnet/data/labeled)과 test data set(/resnet/data/labeled\_test)으로 분리해주었습니다.



치아 우식 라벨링된 이미지

그리고 train data set의 데이터를 flip 기능으로 data augmentation을 해주었습니다.

여기서 path1은 라벨링된 파일들이 저장되어 있는 폴더이고, path2는 augmentation된 파일들이 저장될 폴더 경로입니다.

Augmentation 함수를 사용하면 augmentation 폴더에 augmentation된 데이터+원본 데이터가 합쳐진 train data set이 형성됩니다.

|  |
| --- |
| %cd yourpath/Team1/resnet  from augmentation\_f import augmentation  path = ‘yourpath’ #ex)C:/Users  augmentation(path) |
|  |

1. **Resnet Supervised learning**

augmentation의 파일의 데이터를 resnet 모델 및 crossvalidation을 이용해 학습해줍니다. (resnet/saved\_models/supervised\_weights\_crossval1에 미리 학습된 모델 가중치가 있습니다.)

|  |
| --- |
| %cd yourpath/Team1/resnet  from model import res\_train  path = ‘yourpath’ #ex)C:/Users  weight\_name = ‘supervised\_1’ #학습될 가중치 이름  res\_train(path, weight\_name) |

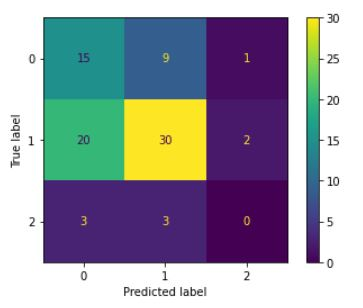
1. **Resnet Semi-supervised learning**

사전 학습된 가중치와 unlabeled dataset을 이용해, Semi-supervised learning을 진행합니다. (resnet/saved\_models/Final3에 미리 학습된 모델 가중치가 있습니다.)

|  |
| --- |
| %cd yourpath/Team1/resnet  from model import semi\_train  path = ‘yourpath’ #ex)C:/Users  weight\_name1 = ‘supervised\_weights\_crossval1’ #사전 훈련된 가중치 이름  weight\_name2 = ‘ssl\_1' #Semi-supervised learning으로 학습될 가중치 이름  semi\_train(path, weight\_name1, weight\_name2) |

1. **모델 평가하기 (confusion matrix)**

평가하기 원하는 모델 가중치의 confusion matrix를 확인합니다.

(Final3의 confusion matrix)

|  |
| --- |
| %cd yourpath/Team1/resnet  from model import confusion\_m  path = ‘yourpath’ #ex)C:/Users  weight\_name = ‘Final3’ #평가 모델 가중치  confusion\_m(path, weight\_name) |

1. 학습된 모델로 치아 견적 산출하기

**Install**

|  |
| --- |
| %cd yourpath/Team1/yolov5  !pip install -r requirements.txt |

**Usage**

1. 한 사람의 개별 치아 분류

견적을 내 줄 사람의 입 전체 데이터를 학습된 yolo 모델로 crop을 해준 후, 데이터들을 (/resnet/data/person)폴더에 저장해줍니다. 그리고 model.py의 predictList 함수를 사용하면, 사람의 개별 치아에 대한 클래스 값 리스트를 받을 수 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| %cd yourpath/Team1/resnet  from model import predictList  path = ‘yourpath’ #ex)C:/Users  weight\_name = ‘Final3’ #모델 가중치  person = ‘person1’#견적받을 사람명  p\_list = predictList(path, weight\_name, person)  classifications=[[]]  for i,j in p\_list:  classifications[0].append(j)  print(classifications) |
|  |

1. 최종 견적 산출

원본 입 전체 이미지에 치아 우식 클래스를 그려주고 최종 견적을 산출합니다.

|  |
| --- |
| %cd yourpath/Team1  from PIL import Image  import matplotlib.pyplot as plt  import torch  from final import box\_save, final\_cost  path = ‘yourpath/Team1’  img = Image.open(path+'/yolov5/data/unlabeled\_jpg\_te/78.jpg')  model = torch.hub.load('.', 'custom', path=path+'/yolov5/runs/train/ your\_results\_name/weights/best.pt', source='local')  coordinates = box\_save(path, img, model)  final\_cost(path, img, classifications, coordinates) |

텍스트이(가) 표시된 사진

자동 생성된 설명