지능화 캡스톤 프로젝트

프로젝트 #2 결과 발표

2023. 6. 14

충북대학교 산업인공지능학과

[3조] 김현기, 원윤재

INDUSTRIAL AI RESEARCH CENTER

수행방법

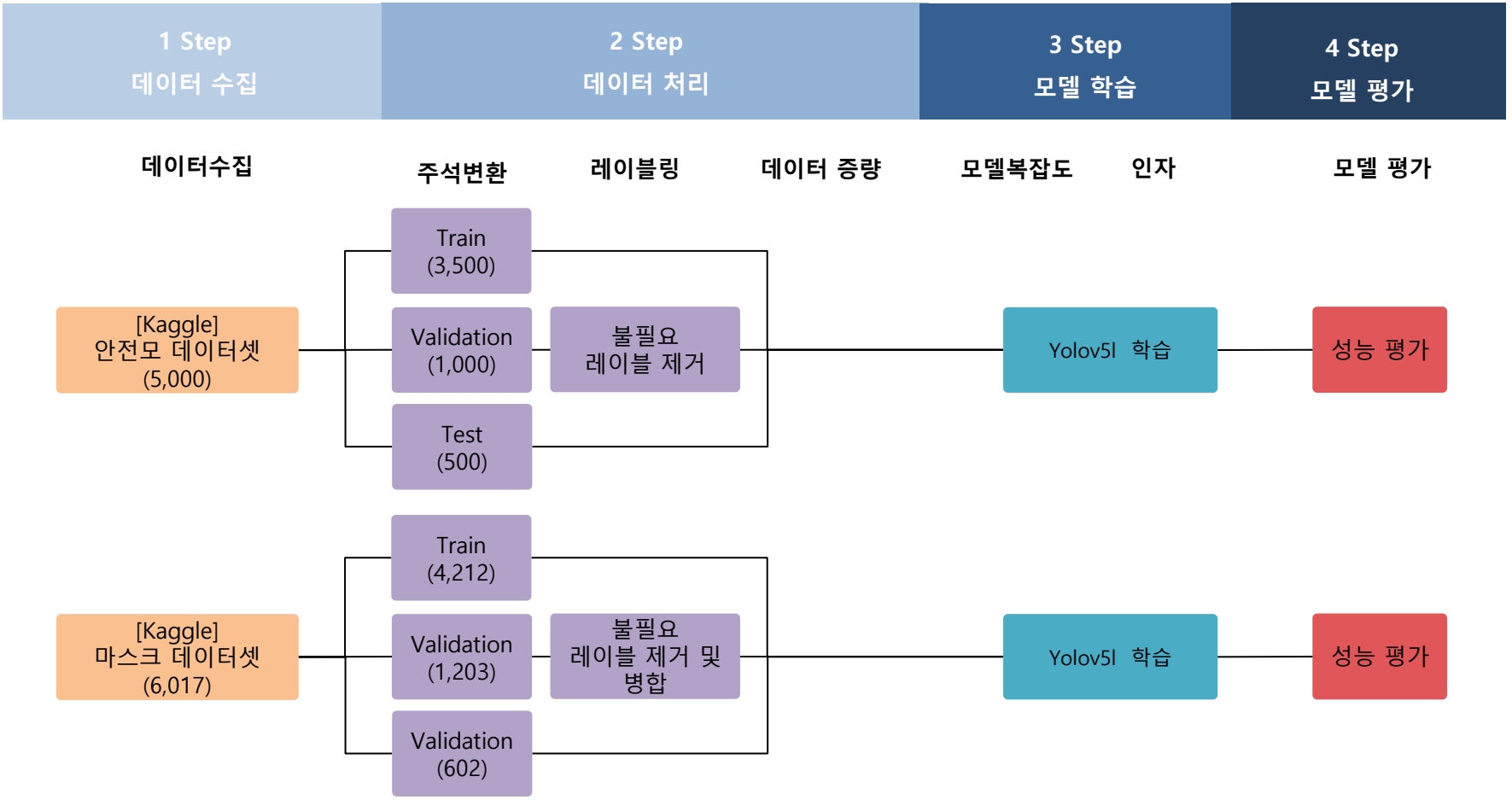
수행방법

- 같은 회사에 재직하여 수시로 내용 공유 및 실험 진행

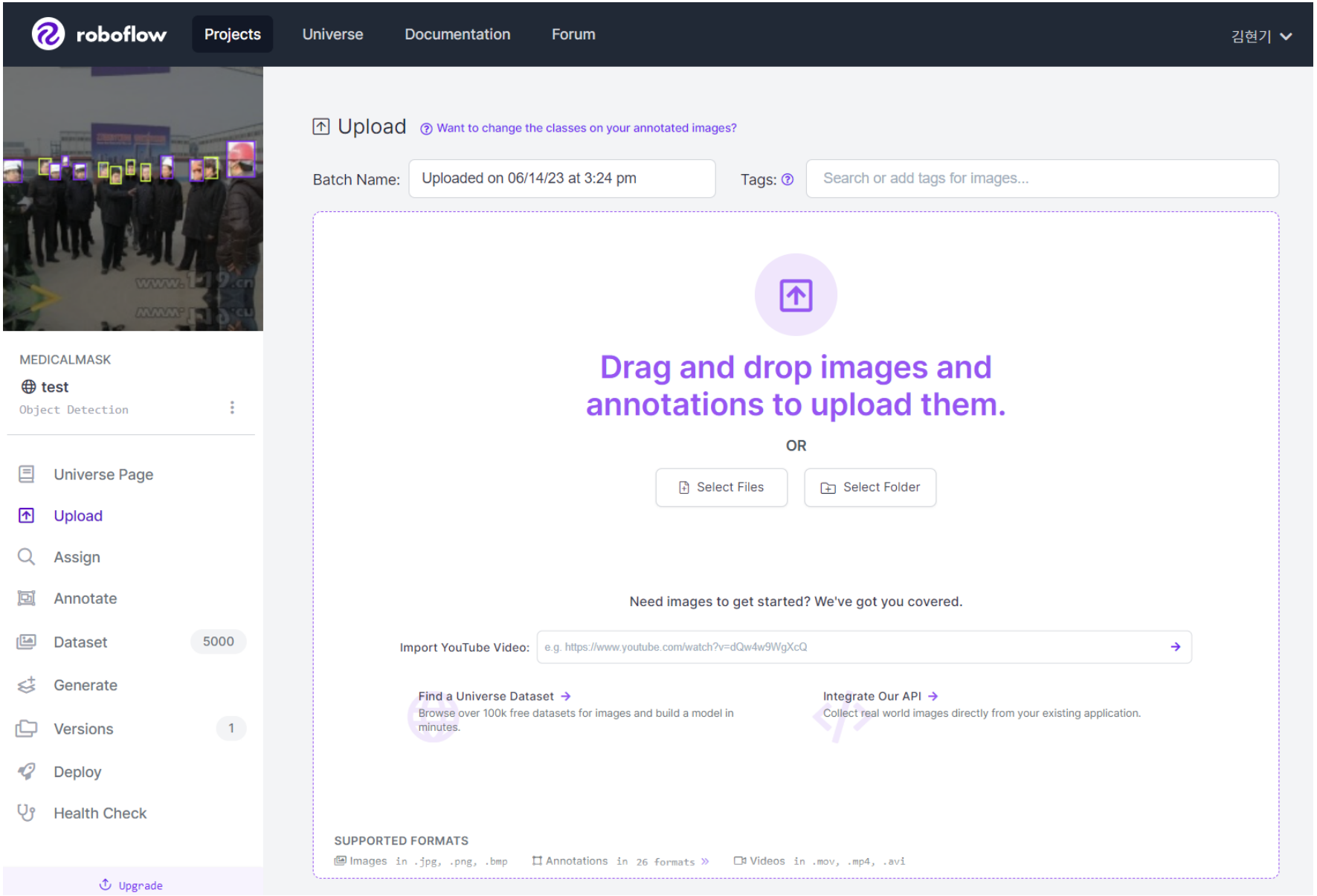
업무분장

이름	수행내용	비고
김현기	• 데이터셋 전처리 • Yolov5 학습환경 구성 • 주제발표	
원윤재	• 데이터셋 취득 • Yolov5 학습환경 구성 • 학습결과 분석 및 개선	

모델 개발 프로세스


1 Step
데이터 수집
2 Step
데이터 처리
3 Step
모델 학습
4 Step
모델 평가
데이터수집
주석변환
레이블링
데이터 증량
모델복잡도
인자
모델 평가
[Kaggle]
안전모 데이터셋
(5,000)
Train
(3,500)
Validation
(1,000)
Test
(500)
불필요
레이블 제거
Yolov5l 학습
성능 평가
[Kaggle]
마스크 데이터셋
(6,017)
Train
(4,212)
Validation
(1,203)
Validation
(602)
불필요
레이블 제거 및
병합
Yolov5l 학습
성능 평가

데이터셋 전처리 – Roboflow Project



데이터셋 전처리 – Annotation 전처리

```
{
    "FileName": "0002.png",
    "NumOfAnno": 1,
    "Annotations": [
        {
            "isProtected": false,
            "ID": 984460083838631424,
            "BoundingBox": [
                332,
                9,
                590,
                371
            ],
            "classname": "face_no_mask",
            "Confidence": 1,
            "Attributes": {}
        }
    ]
}
```

[JSON 형식]

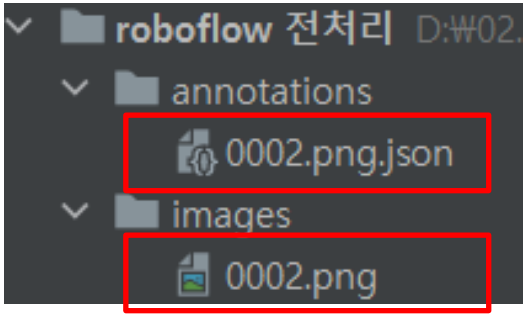
전처리

```
<annotation>
    <filename>0002.jpg</filename>
    <size>
        <width>740</width>
        <height>416</height>
    </size>
    <object>
        <name>face_no_mask</name>
        <bndbox>
            <xmin>332</xmin>
            <ymin>9</ymin>
            <xmax>590</xmax>
            <ymax>371</ymax>
        </bndbox>
    </object>
</annotation>
```

[Pascal VOC 형식]

>>이미지 size 속성 필요

데이터셋 전처리 – Annotation 전처리



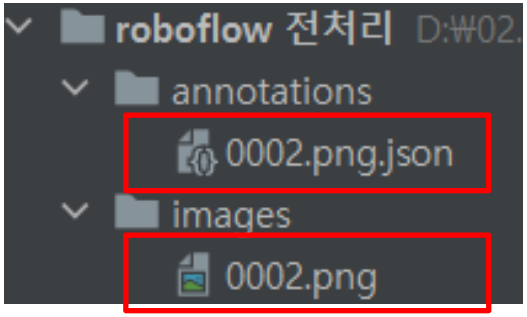
[데이터셋 구성 예시]

*쌍을 이루고 있음
- 0002.png.json
- 0002.png

```python
# 디렉토리 내의 JSON 파일을 순환하며 처리
for json_file in os.listdir(annotation_directory):
    if json_file.endswith(".json"):
        json_path = os.path.join(annotation_directory, json_file)

        # JSON 파일 읽기
        with open(json_path) as file:
            data = json.load(file)

        file_name = os.path.splitext(data["FileName"])[0] + ".jpg"
        image_path = os.path.join(image_directory, file_name)

        if os.path.isfile(image_path):
            image = Image.open(image_path)
            width, height = image.size
            create_pascal_voc_xml(file_name, width, height, data["Annotations"])
```

[대상 이미지파일 크기 추출 및 json파일 매핑]

데이터셋 전처리 – Annotation 전처리



[데이터셋 구성 예시]

*쌍을 이루고 있음
- 0002.png.json
- 0002.png

```python
def create_pascal_voc_xml(filename, width, height, annotations):
    root = ET.Element("annotation")

    filename_elem = ET.SubElement(root, "filename")
    filename_elem.text = filename

    size_elem = ET.SubElement(root, "size")
    width_elem = ET.SubElement(size_elem, "width")
    width_elem.text = str(width)
    height_elem = ET.SubElement(size_elem, "height")
    height_elem.text = str(height)

    for annotation in annotations:
        object_elem = ET.SubElement(root, "object")

        name_elem = ET.SubElement(object_elem, "name")
        name_elem.text = annotation["classname"]

        bndbox_elem = ET.SubElement(object_elem, "bndbox")
        xmin_elem = ET.SubElement(bndbox_elem, "xmin")
        xmin_elem.text = str(annotation["BoundingBox"][0])
        ymin_elem = ET.SubElement(bndbox_elem, "ymin")
        ymin_elem.text = str(annotation["BoundingBox"][1])
        xmax_elem = ET.SubElement(bndbox_elem, "xmax")
        xmax_elem.text = str(annotation["BoundingBox"][2])
        ymax_elem = ET.SubElement(bndbox_elem, "ymax")
        ymax_elem.text = str(annotation["BoundingBox"][3])

    tree = ET.ElementTree(root)
    fileStr = filename+".xml"
    tree.write(os.path.join(result_directory,fileStr))
```

[XML 파일 생성]

데이터셋 전처리 – Roboflow Dataset

roboflow | Projects | Universe | Documentation | Forum | 김현기

MEDICALMASK
test
Object Detection

- Universe Page
- Upload
- Assign
- Annotate
- Dataset 5000
- Generate
- Versions 1
- Deploy
- Health Check

test Dataset

Generate New Version

VERSIONS

2023-06-09 5:43pm
v1 Jun 9, 2023

Generating New Version

Prepare your images and data for training by compiling them into a version.
Experiment with different configurations to achieve better training results.

Source Images
Images: 5,000
Classes: 3
Unannotated: 0

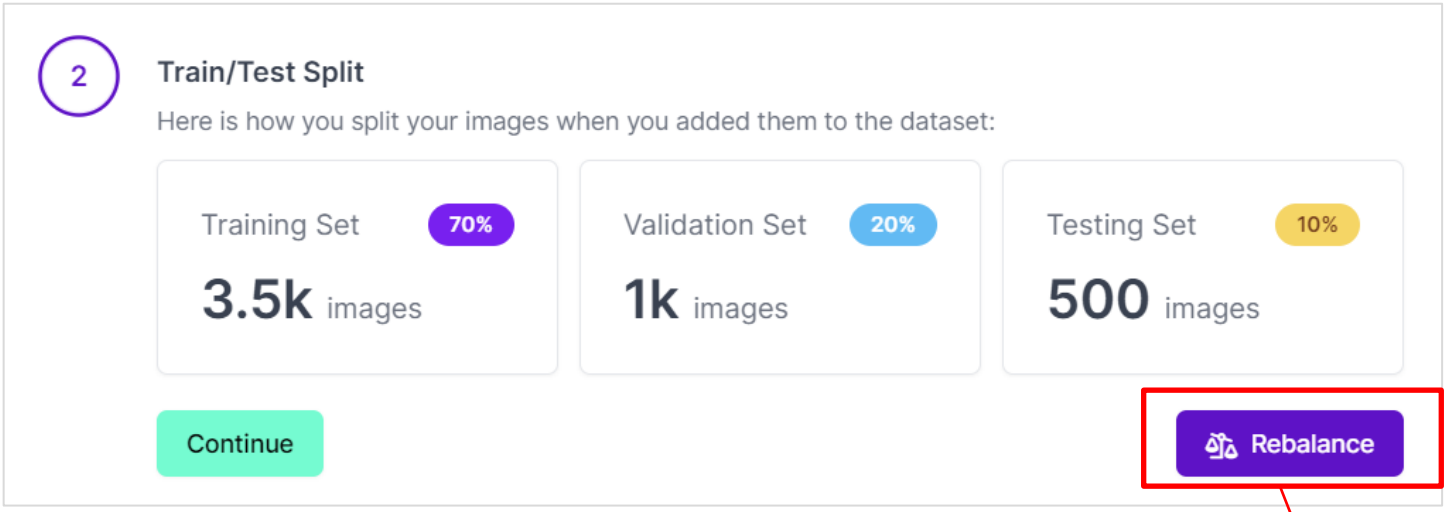
Train/Test Split
Training Set: 3.5k images
Validation Set: 1k images
Testing Set: 500 images

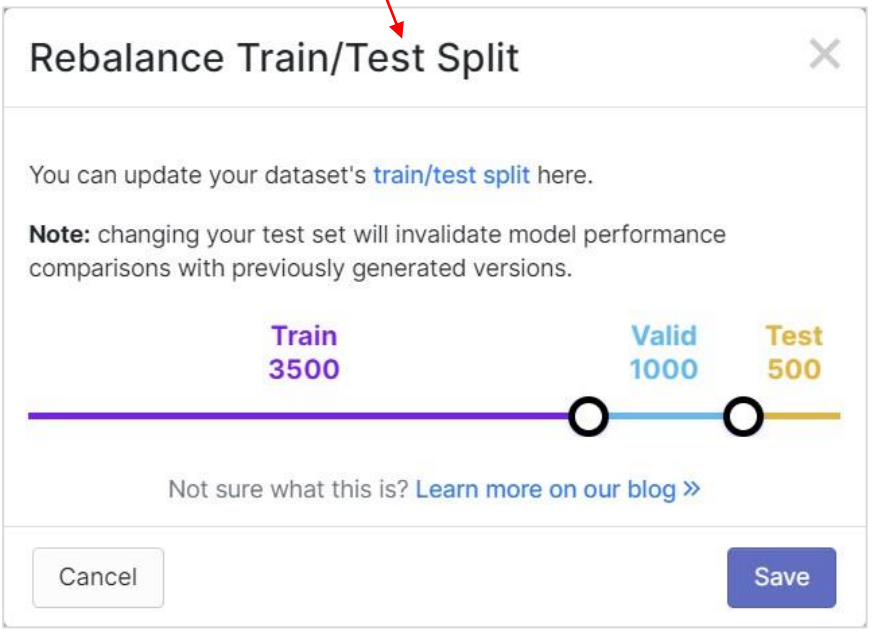
Preprocessing
Auto-Orient: Applied
Resize: Stretch to 640×640

Augmentation
Turned Off

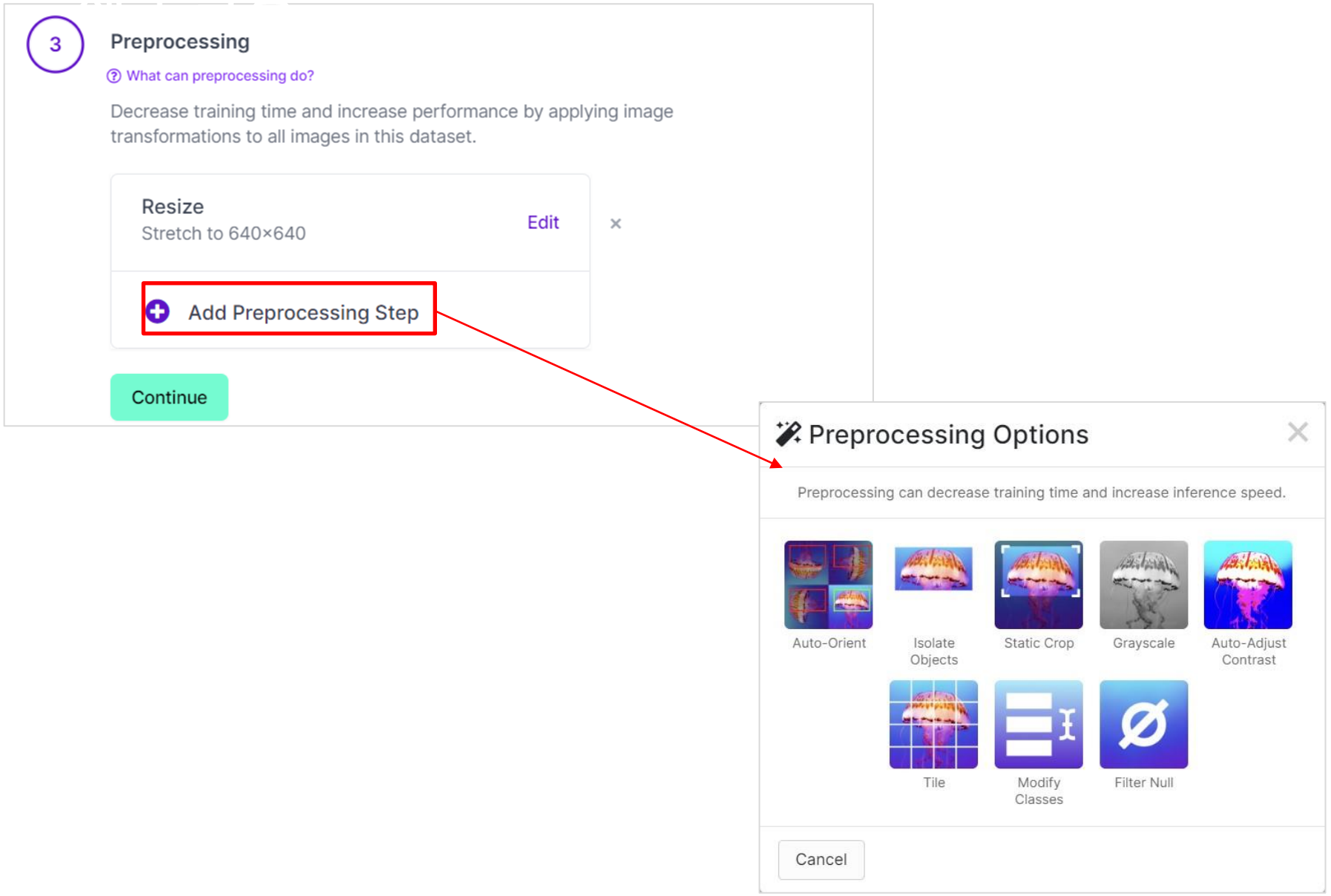
5 **Generate**
Review your selections then click "Generate" to create a moment-in-time snapshot of your dataset with the applied preprocessing steps.

데이터셋 전처리 – 데이터 분할

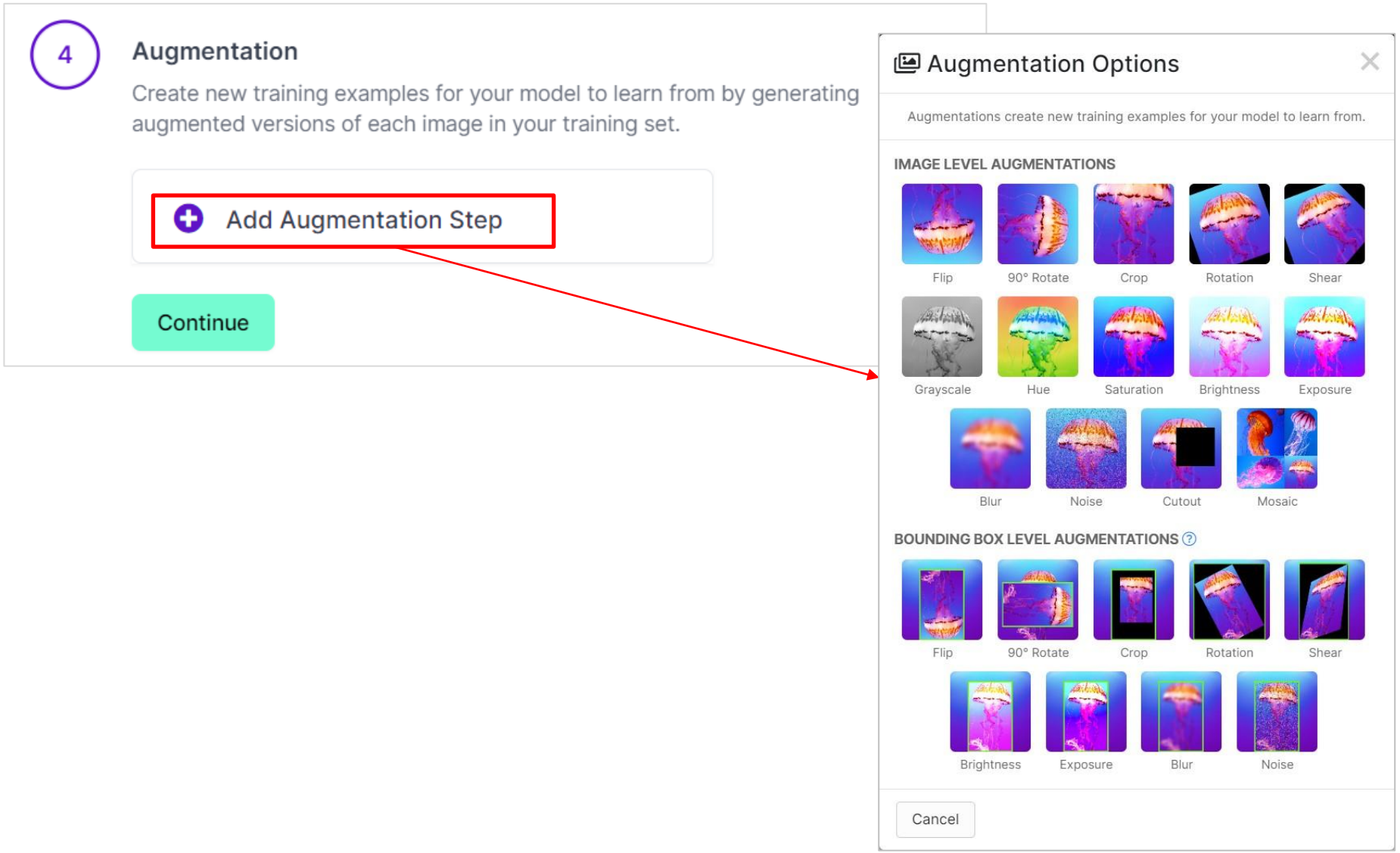




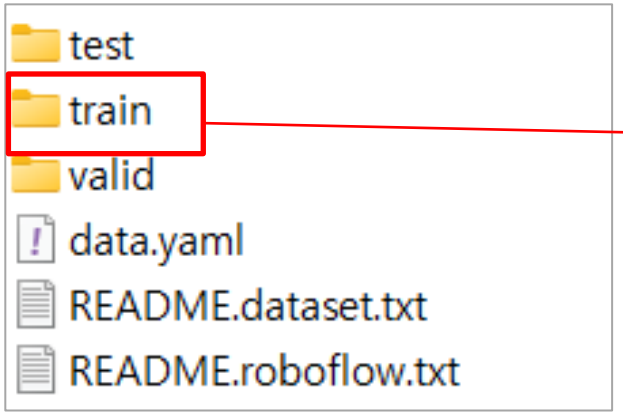
데이터셋 전처리 – 데이터 전처리



데이터셋 전처리 – 데이터 증량



데이터셋 – 데이터셋 처리 결과

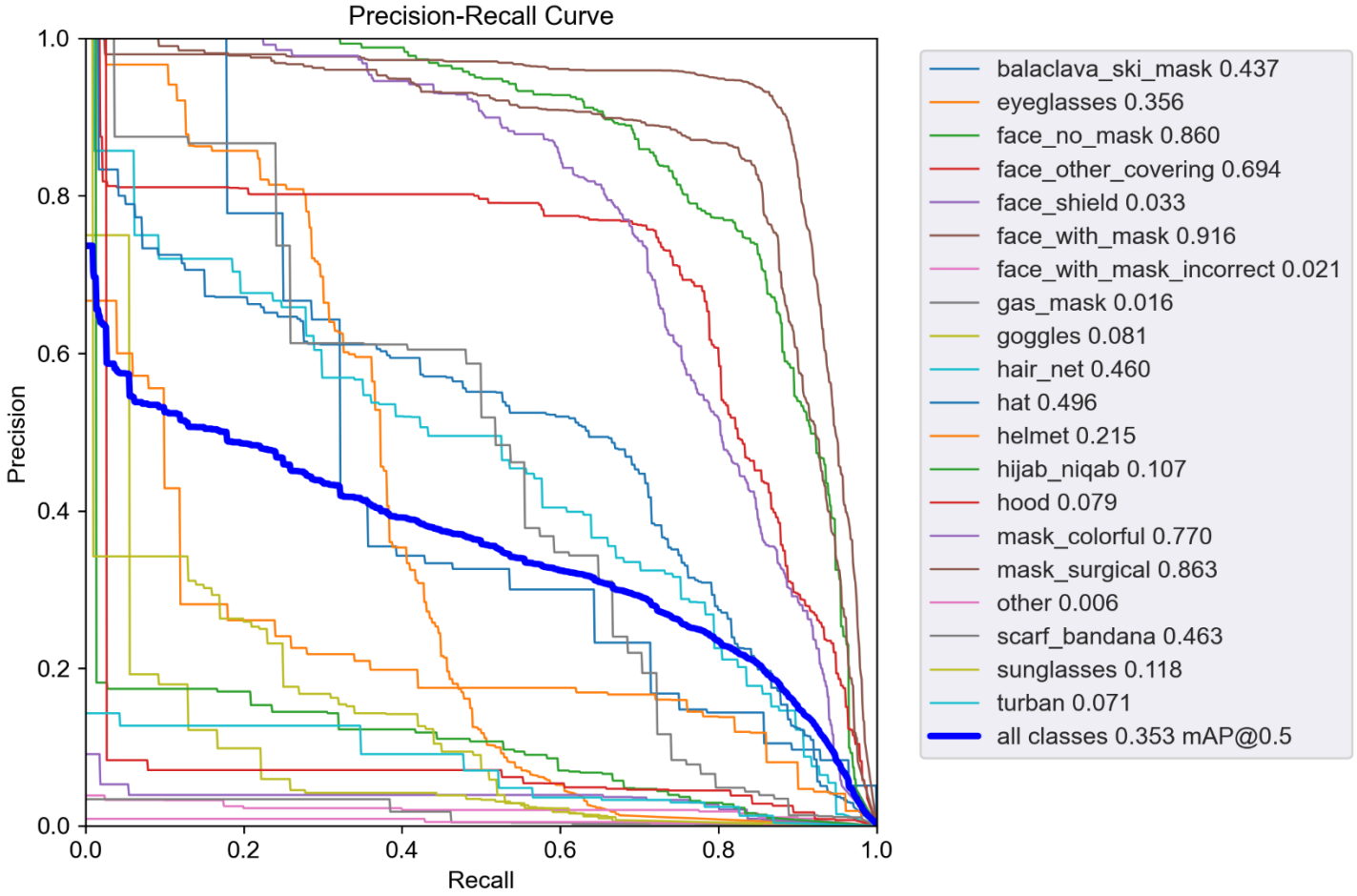


[데이터셋]


```
3 0.49140625 0.5006944444444444 0.3 0.6152777777777778
10 0.464453125 0.20069444444444445 0.35859375 0.3736111111111111
13 0.50859375 0.5090277777777777 0.3859375 0.6736111111111112
```

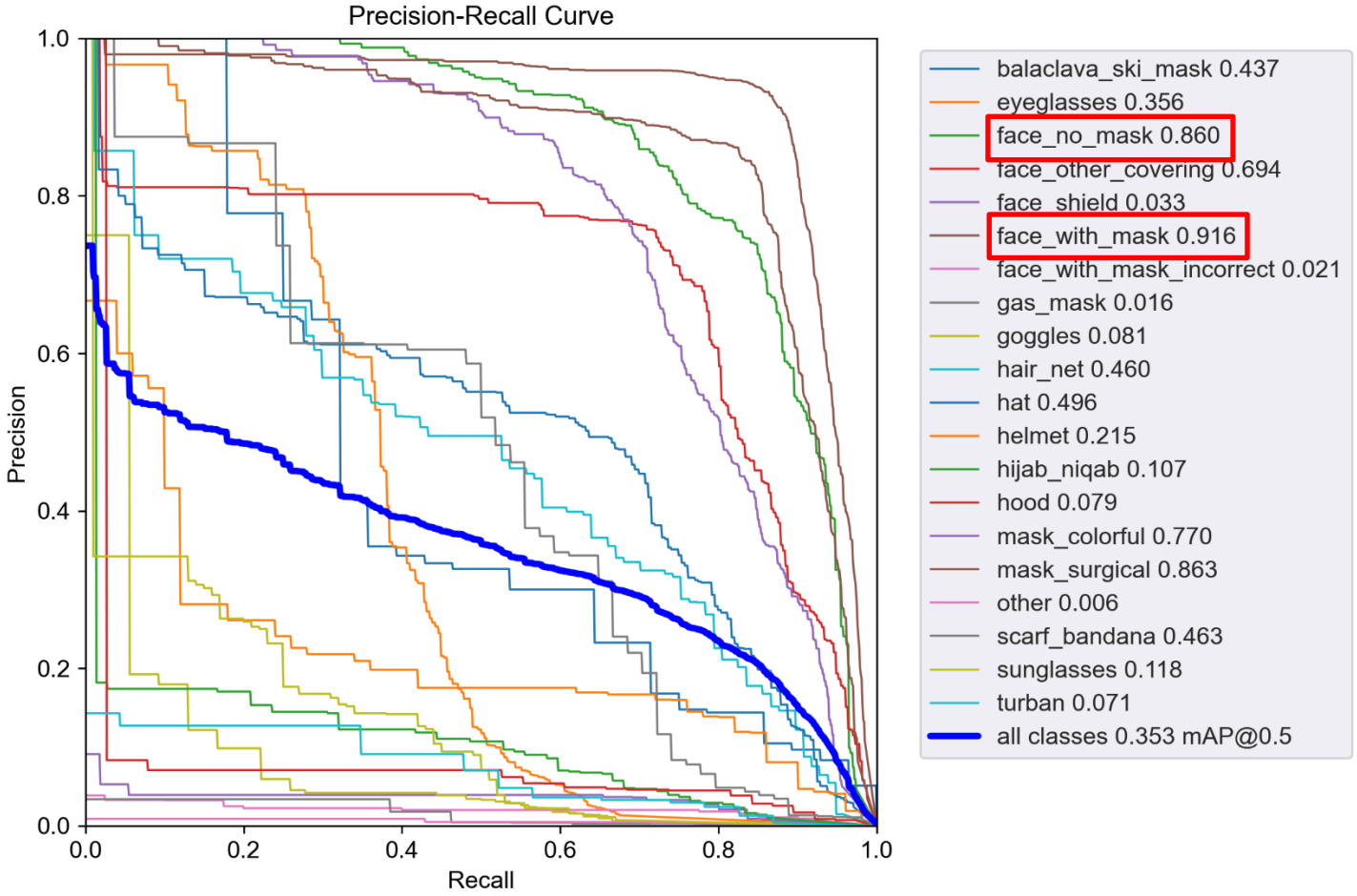
[labels 데이터 형식]

데이터셋 – 이슈사항



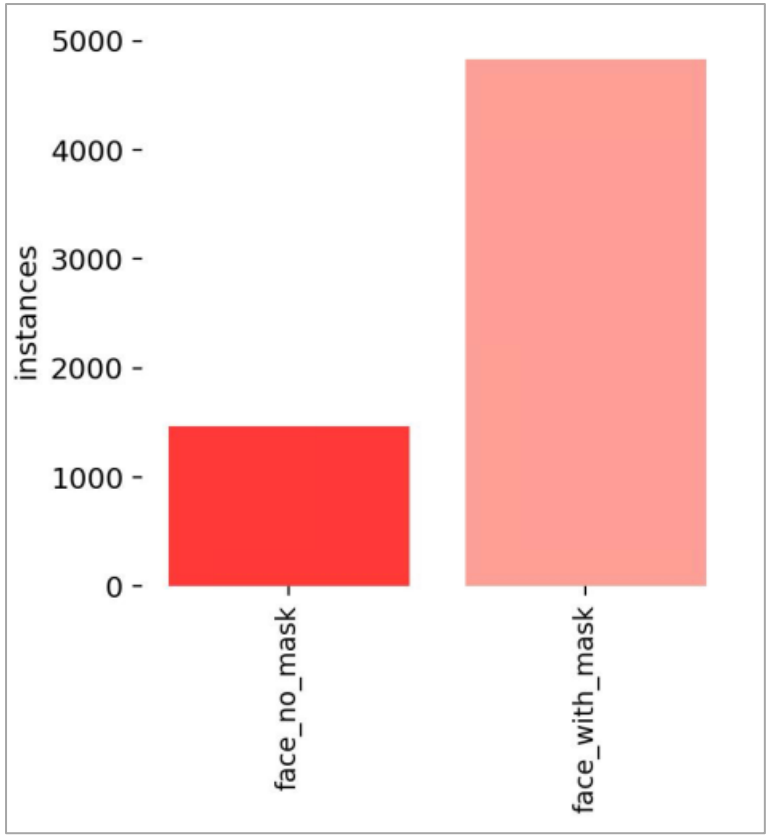
[세분화된 클래스]

데이터셋 – 이슈사항



[세분화된 클래스]

데이터셋 – 이슈사항



[데이터셋 인스턴스 분포]

Class	Images	Instances	P	R	mAP50	mAP50-95
all	1000	3297	0.403	0.247	0.204	0.0422
face_no_mask	1000	2033	0.448	0.0944	0.128	0.0285
face_with_mask	1000	1264	0.358	0.4	0.28	0.0559

[검증 결과]

데이터셋 – 이슈사항

```python
for filename in os.listdir(directory):
    if filename.endswith(".txt"):
        file_path = os.path.join(directory, filename)

        with open(file_path, 'r') as file:
            lines = file.readlines()

        # 숫자를 변경하고 필터링하여 새로운 리스트 생성
        filtered_lines = []
        for line in lines:
            parts = line.split()
            if len(parts) > 0:
                first_num = int(parts[0])
                if first_num == 2 or first_num == 3:
                    parts[0] = '0'
                elif first_num == 5 or first_num == 0 or first_num == 5 or first_num == 6 or first_num == 7 \
                        or first_num == 14 or first_num == 15:
                    parts[0] = '1'
                else:
                    continue
            filtered_lines.append(' '.join(parts) + '\n')

        with open(file_path, 'w') as file:
            file.writelines(filtered_lines)
```

[주석 전처리 py]

```yaml
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 20
names: ['balaclava_ski_mask',
        'eyeglasses',
        'face_no_mask',
        'face_other_covering',
        'face_shield',
        'face_with_mask',
        'face_with_mask_incorrect',
        'gas_mask',
        'goggles',
        'hair_net',
        'hat',
        'helmet',
        'hijab_niqab',
        'hood',
        'mask_colorful',
        'mask_surgical',
        'other',
        'scarf_bandana',
        'sunglasses',
        'turban']
```

[마스크 X 데이터 병합]

데이터셋 – 이슈사항

```python
for filename in os.listdir(directory):
    if filename.endswith(".txt"):
        file_path = os.path.join(directory, filename)

        with open(file_path, 'r') as file:
            lines = file.readlines()

        # 숫자를 변경하고 필터링하여 새로운 리스트 생성
        filtered_lines = []
        for line in lines:
            parts = line.split()
            if len(parts) > 0:
                first_num = int(parts[0])
                if first_num == 2 or first_num == 3:
                    parts[0] = '0'
                elif first_num == 5 or first_num == 0 or first_num == 5 or first_num == 6 or first_num == 7 \
                        or first_num == 14 or first_num == 15:
                    parts[0] = '1'
                else:
                    continue
            filtered_lines.append(' '.join(parts) + '\n')

        with open(file_path, 'w') as file:
            file.writelines(filtered_lines)
```

[주석 전처리 py]

```yaml
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 20
names: ['balaclava_ski_mask',
        'eyeglasses',
        'face_no_mask',
        'face_other_covering',
        'face_shield',
        'face_with_mask',
        'face_with_mask_incorrect',
        'gas_mask',
        'goggles',
        'hair_net',
        'hat',
        'helmet',
        'hijab_niqab',
        'hood',
        'mask_colorful',
        'mask_surgical',
        'other',
        'scarf_bandana',
        'sunglasses',
        'turban']
```

[마스크 O 데이터 병합]

데이터셋 – 이슈사항

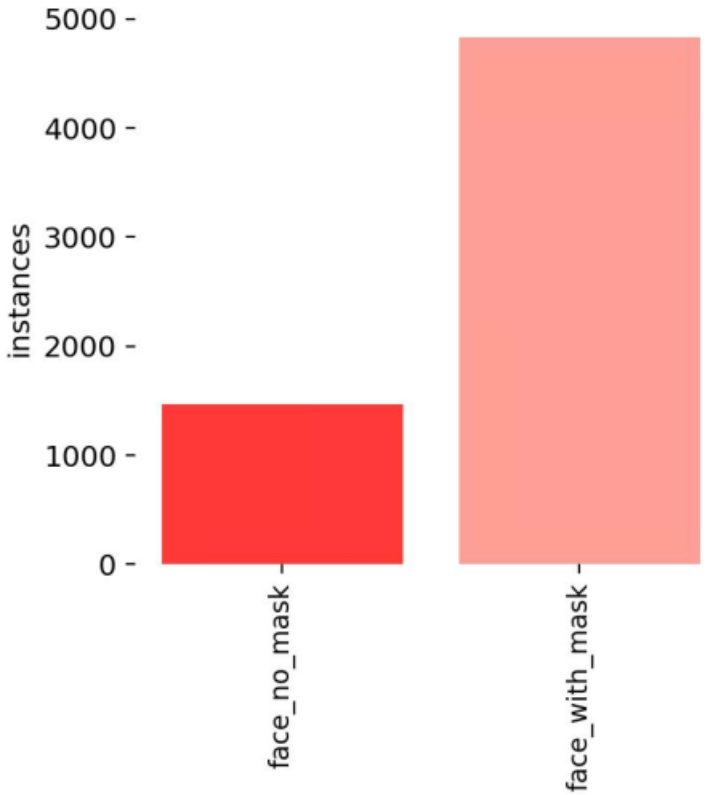
```python
for filename in os.listdir(directory):
    if filename.endswith(".txt"):
        file_path = os.path.join(directory, filename)

        with open(file_path, 'r') as file:
            lines = file.readlines()

        # 숫자를 변경하고 필터링하여 새로운 리스트 생성
        filtered_lines = []
        for line in lines:
            parts = line.split()
            if len(parts) > 0:
                first_num = int(parts[0])
                if first_num == 2 or first_num == 3:
                    parts[0] = '0'
                elif first_num == 5 or first_num == 0 or first_num == 5 or first_num == 6 or first_num == 7 \
                        or first_num == 14 or first_num == 15:
                    parts[0] = '1'
                else:
                    continue
            filtered_lines.append(' '.join(parts) + '\n')

        with open(file_path, 'w') as file:
            file.writelines(filtered_lines)
```

[주석 전처리 py]

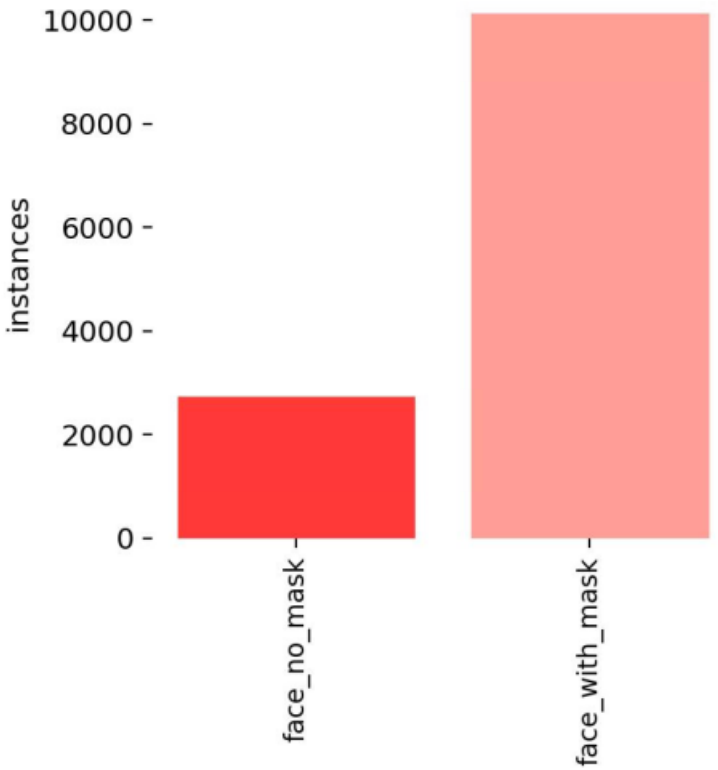
```yaml
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 20
names: ['balaclava_ski_mask',
        'eyeglasses',
        'face_no_mask',
        'face_other_covering',
        'face_shield',
        'face_with_mask',
        'face_with_mask_incorrect',
        'gas_mask',
        'goggles',
        'hair_net',
        'hat',
        'helmet',
        'hijab_niqab',
        'hood',
        'mask_colorful',
        'mask_surgical',
        'other',
        'scarf_bandana',
        'sunglasses',
        'turban']
```

[불필요 클래스 제거]

데이터셋 – 이슈사항



[주석 전처리 전]



[주석 전처리 후]

[1500건 > 2900건으로 증가된 face_no_mask]

모델 학습

딥러닝 학습 환경

하드웨어	
CPU	Intel® Core™ i9-9960X CPU @ 3.10GHz
RAM	32GB
GPU	NVDIA GeForce GTX 2080 TI 10GB

소프트웨어	
Python	3.8.6
Pytorch	1.9.0+cu111
Yolo	v5

학습 파라미터	
Image size	416
Batch size	16
Epoch	50
Weights	yolov5l

모델 학습

학습 방법

[Helmet]

```
(venv) C:₩Users₩kseclab₩Desktop₩new_cbnu₩project_2₩yolov5-master-git₩yolov5>python train.py --img 416 --batch 16 --epochs 50 --data ./datasets/helmet/data.yaml --
weights yolov5l.pt --cache
train: weights=yolov5l.pt, cfg=, data=./datasets/helmet/data.yaml, hyp=data₩hyps₩hyp.scratch-low.yaml, epochs=50, batch_size=16, imgsz=416, rect=False, resume=Fal
se, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=ram, image_weights=False, device=, multi_scale=False, single_cls=Fal
se, optimizer=SGD, sync_bn=False, workers=8, project=runs₩train, name=exp, exist_ok=False, quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0]
, save_period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_interval=-1, artifact_alias=latest
Auto packing the repository in background for optimum performance.
See "git help gc" for manual housekeeping.
Enumerating objects: 11180, done.
Counting objects: 100% (11020/11020), done.
Delta compression using up to 32 threads
Compressing objects: 100% (10978/10978), done.
Writing objects: 100% (11020/11020), done.
Total 11020 (delta 2), reused 0 (delta 0), pack-reused 0
Removing duplicate objects: 100% (256/256), done.
Command 'git fetch origin' timed out after 5 seconds
YOLOv5  v7.0-178-ga199480b Python-3.8.6 torch-1.9.0+cu111 CUDA:0 (NVIDIA GeForce RTX 2080 Ti, 11264MiB)
```

$ python train.py --img 416 --batch 16 --epoch 50 --data ./datasets/helmet/data.yaml --weights yolov5l.pt --cache

[Mask]

```
(venv) C:₩Users₩kseclab₩Desktop₩new_cbnu₩project_2₩yolov5-master-git₩yolov5>python train.py --img 416 --batch 16 --epochs 50 --data ./datasets/data.yaml --weights
 yolov5l.pt --cache
train: weights=yolov5l.pt, cfg=, data=./datasets/data.yaml, hyp=data₩hyps₩hyp.scratch-low.yaml, epochs=50, batch_size=16, imgsz=416, rect=False, resume=False, nos
ave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=ram, image_weights=False, device=, multi_scale=False, single_cls=False, opt
imizer=SGD, sync_bn=False, workers=8, project=runs₩train, name=exp, exist_ok=False, quad=False, cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0], save_
period=-1, seed=0, local_rank=-1, entity=None, upload_dataset=False, bbox_interval=-1, artifact_alias=latest
github: up to date with https://github.com/ultralytics/yolov5
YOLOv5  v7.0-178-ga199480 Python-3.8.6 torch-1.9.0+cu111 CUDA:0 (NVIDIA GeForce RTX 2080 Ti, 11264MiB)
```

$ python train.py --img 416 --batch 16 --epoch 50 --data ./datasets/data.yaml --weights yolov5l.pt --cache

모델 학습

학습 결과(계속)

[Helmet]

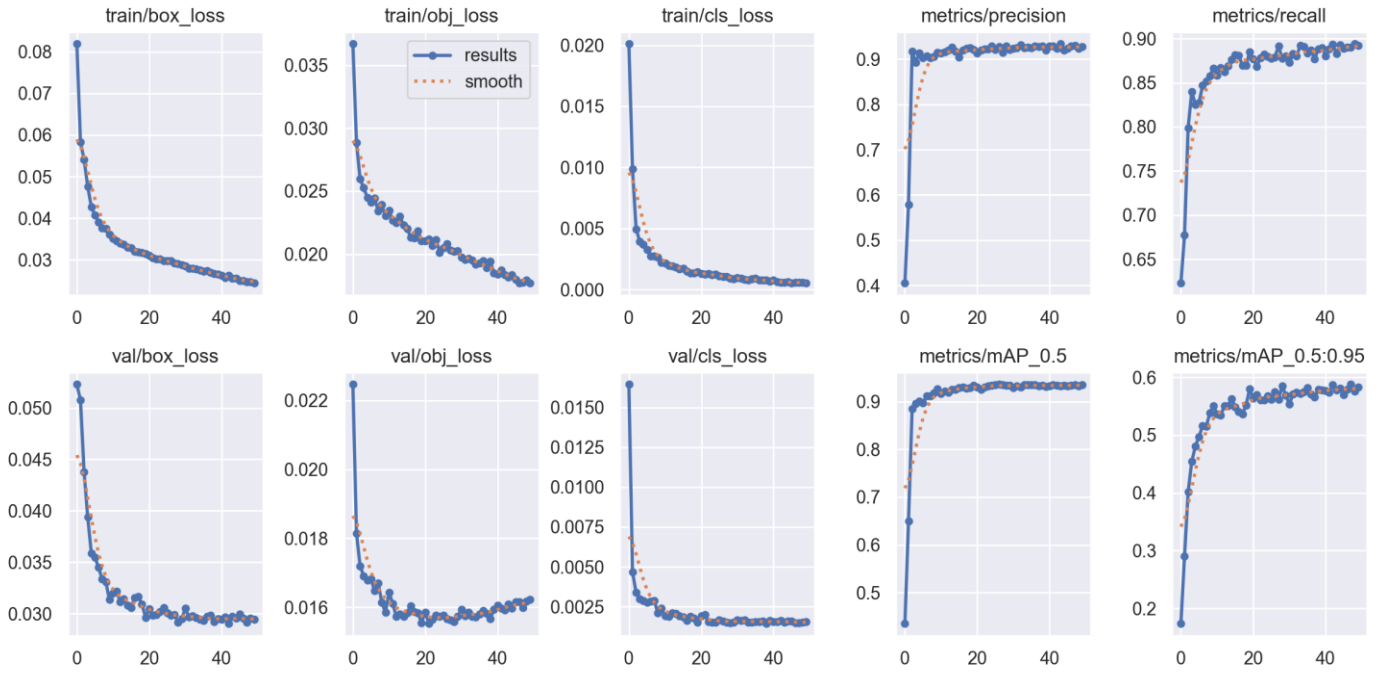
```
50 epochs completed in 0.690 hours.
Optimizer stripped from runs₩train₩exp21₩weights₩last.pt, 92.8MB
Optimizer stripped from runs₩train₩exp21₩weights₩best.pt, 92.8MB

Validating runs₩train₩exp21₩weights₩best.pt...
Fusing layers...
Model summary: 267 layers, 46113663 parameters, 0 gradients, 107.7 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|          | 32/32 [00:11<00:00,  2.81it/s]
                   all       1000       5043      0.931       0.89      0.935      0.588
                  head       1000       1315      0.914      0.876      0.913      0.569
                helmet       1000       3728      0.947      0.904      0.957      0.607
Results saved to runs₩train₩exp21
```

학습 시간

- **0.69시간(41분 24초)**

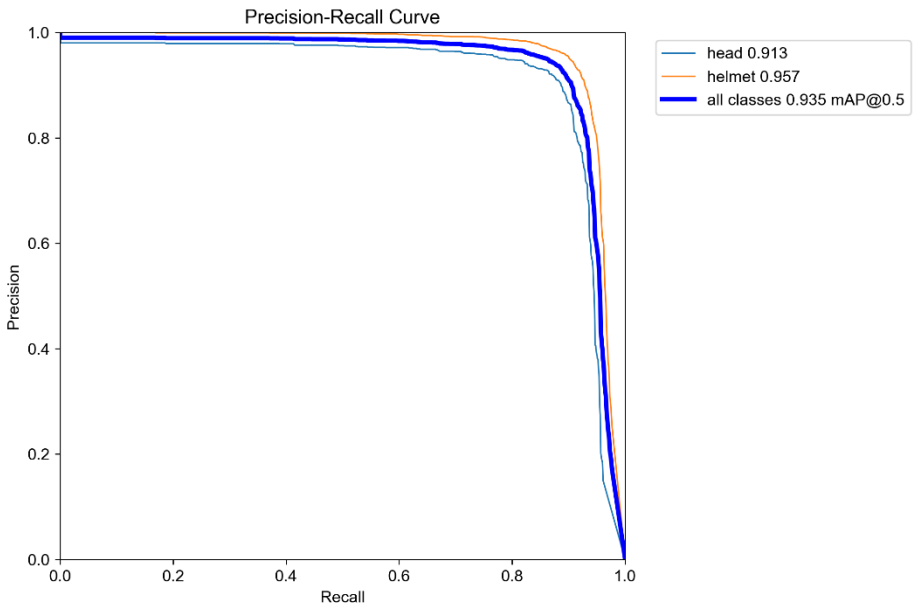
정확도

- **All: 93.5%**
- **Head: 91.3%**
- **Helmet: 95.7%**

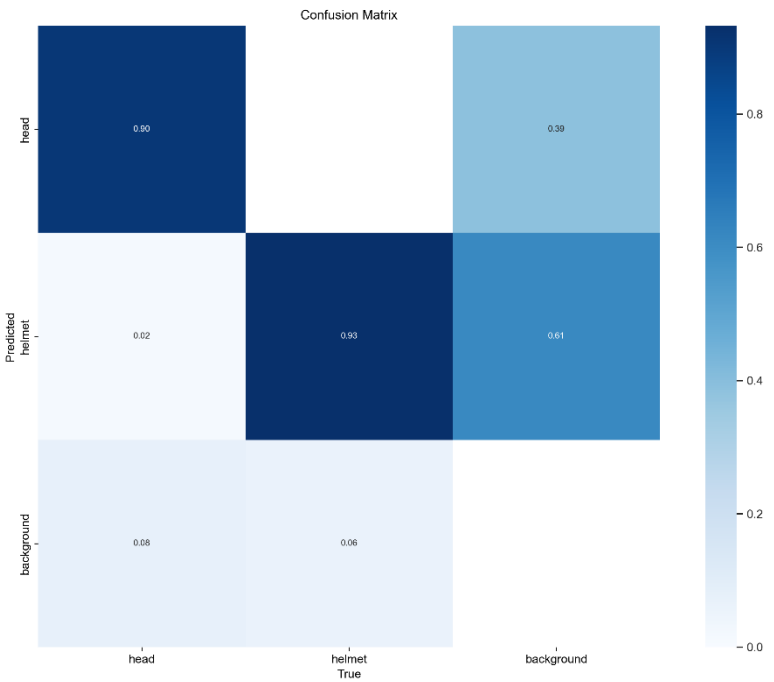


모델 학습

학습 결과(계속)



[PR Curve]



[Confusion Metrix]

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	1000	4317	0.899	0.83	0.883	0.474
head	1000	997	0.933	0.689	0.807	0.425
helmet	1000	3320	0.866	0.97	0.958	0.524

[검증 결과]

모델 학습

학습 결과(계속)

[Mask1]

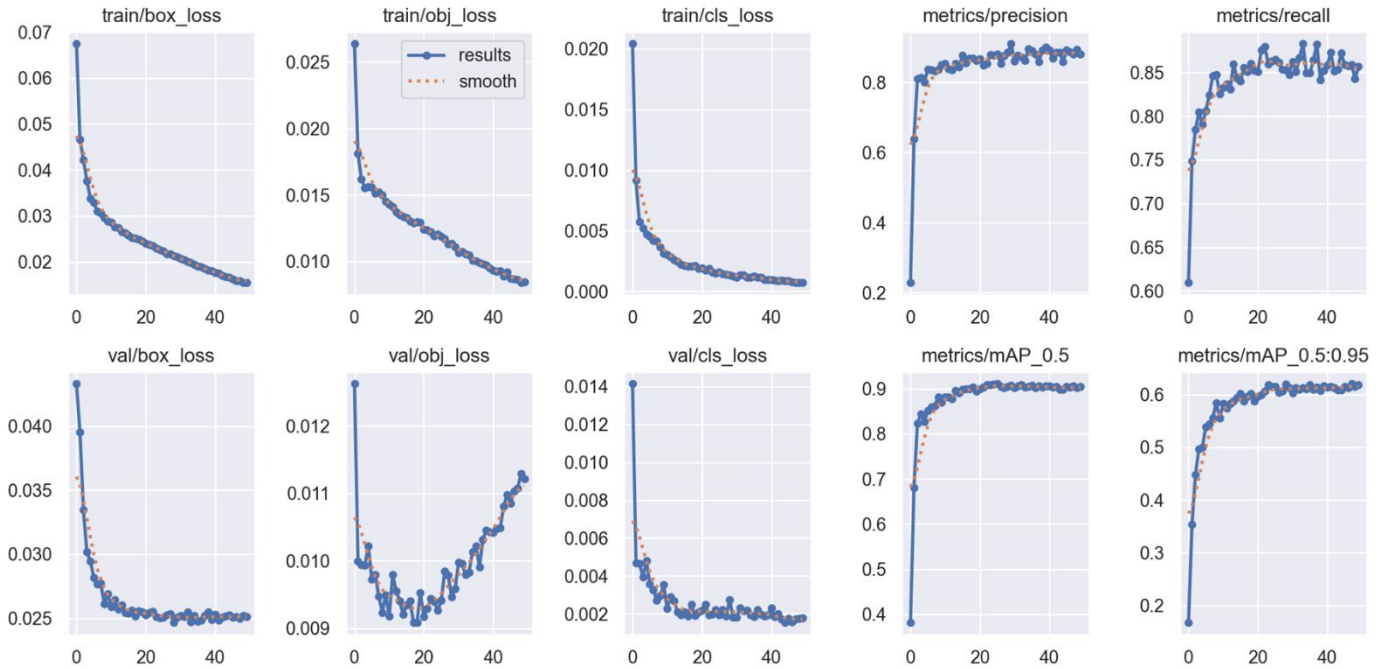
```
50 epochs completed in 0.811 hours.
Optimizer stripped from runs₩train₩exp20₩weights₩last.pt, 92.7MB
Optimizer stripped from runs₩train₩exp20₩weights₩best.pt, 92.7MB

Validating runs₩train₩exp20₩weights₩best.pt...
Fusing layers...
Model summary: 267 layers, 46113663 parameters, 0 gradients, 107.7 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|          | 38/38 [00:10<00:00,  3.66it/s]
                   all       1203       1749      0.881       0.86      0.906       0.62
          face_no_mask       1203        419      0.833      0.816      0.873      0.614
        face_with_mask       1203       1330       0.93      0.903      0.939      0.626
Results saved to runs₩train₩exp20
```

학습 시간

- **0.811시간(48분 40초)**

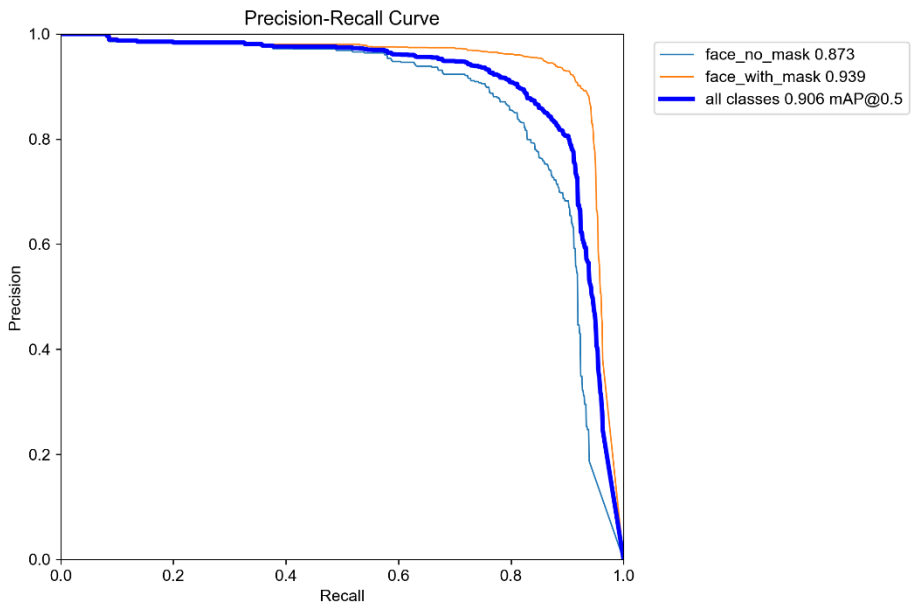
정확도

- **All: 90.6%**
- **No Mask: 87.3%**
- **With Mask: 93.9%**

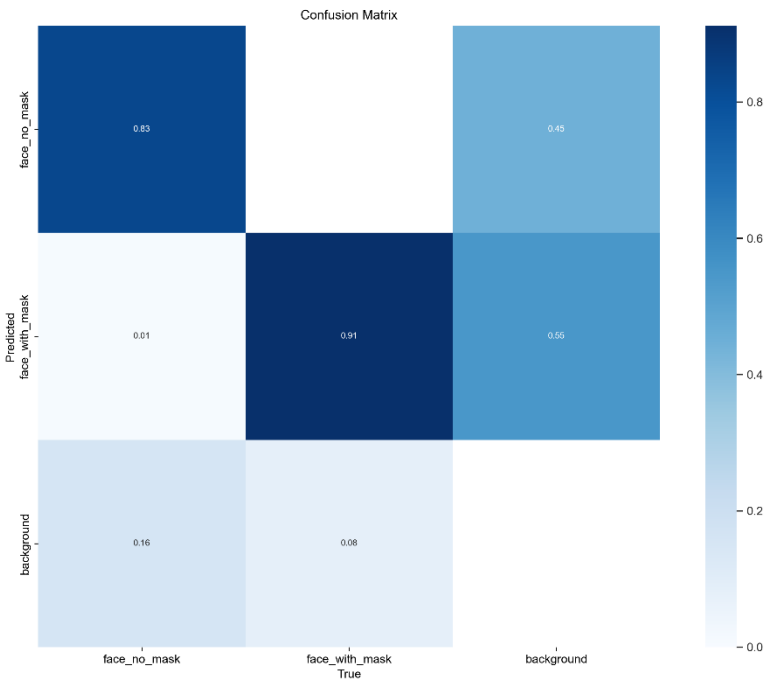


모델 학습

학습 결과



[PR Curve]



[Confusion Metrix]

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	1000	3297	0.403	0.247	0.204	0.0422
face_no_mask	1000	2033	0.448	0.0944	0.128	0.0285
face_with_mask	1000	1264	0.358	0.4	0.28	0.0559

[검증 결과]

모델 학습

학습 결과(계속)

[Mask2]

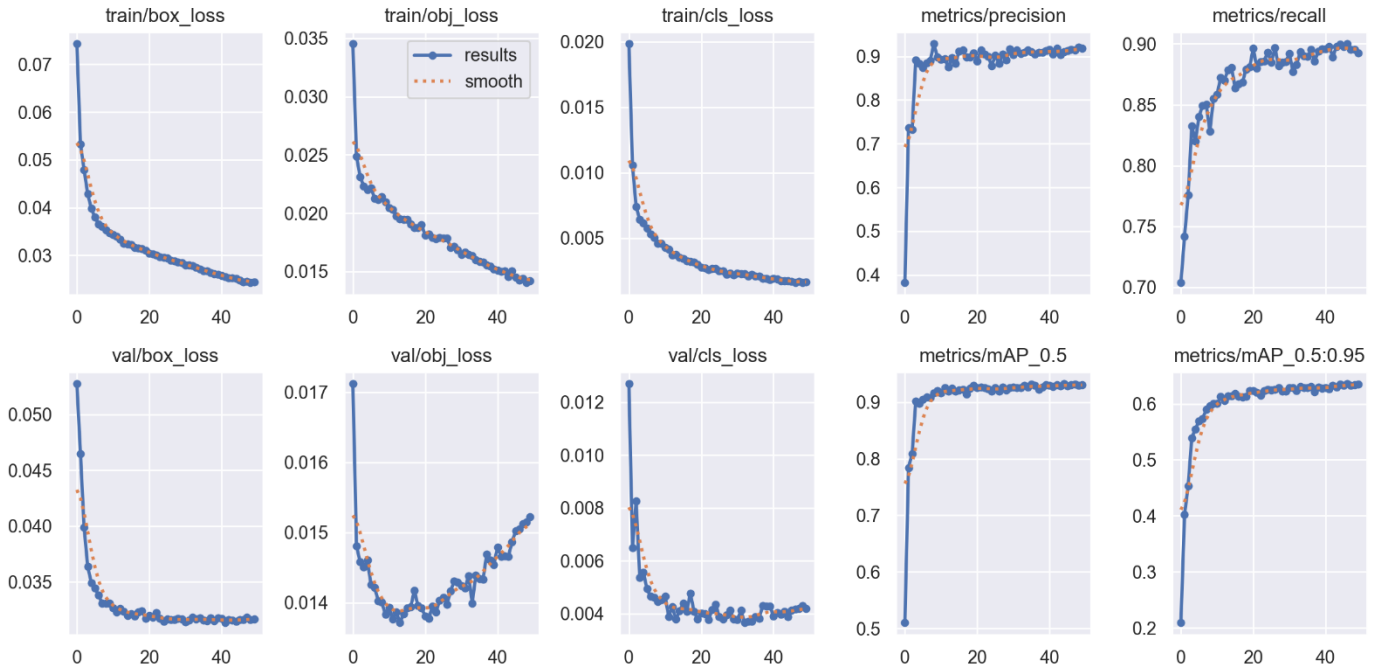
```
50 epochs completed in 0.837 hours.
Optimizer stripped from runs₩train₩exp25₩weights₩last.pt, 92.7MB
Optimizer stripped from runs₩train₩exp25₩weights₩best.pt, 92.7MB

Validating runs₩train₩exp25₩weights₩best.pt...
Fusing layers...
Model summary: 267 layers, 46113663 parameters, 0 gradients, 107.7 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|          | 38/38 [00:12<00:00,  3.01it/s]
                   all       1203       3522      0.915        0.9       0.93      0.636
          face_no_mask       1203        749        0.9      0.888      0.918      0.625
        face_with_mask       1203       2773       0.93      0.912      0.943      0.646
Results saved to runs₩train₩exp25
```

학습 시간

- **0.837시간(50분 22초)**

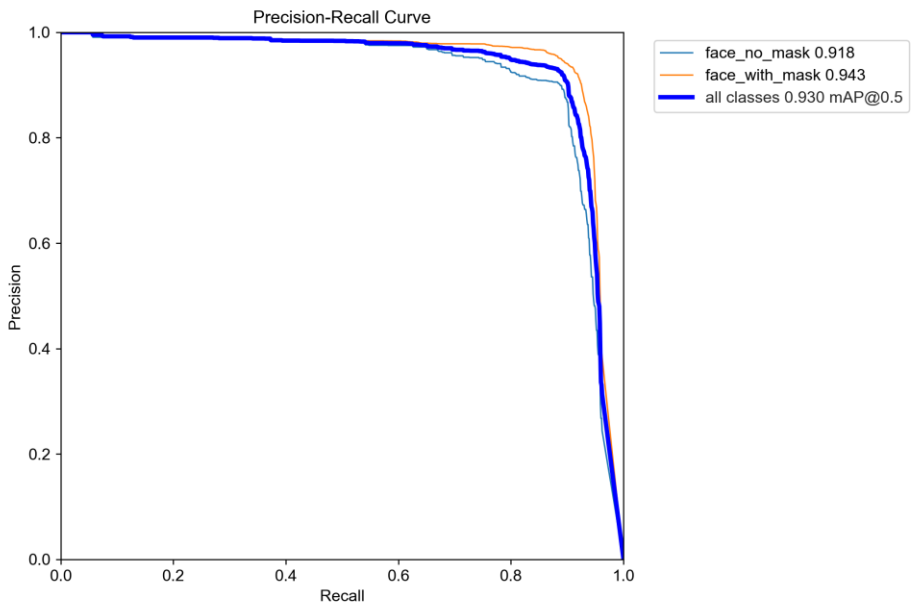
정확도

- **All: 93%**
- **No Mask: 91.8%**
- **With Mask: 94.3%**

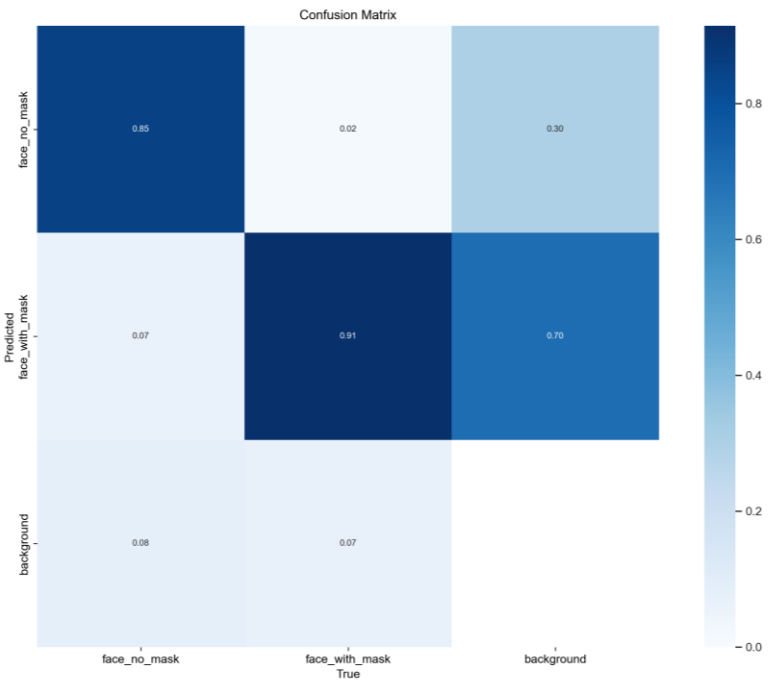


모델 학습

학습 결과



[PR Curve]

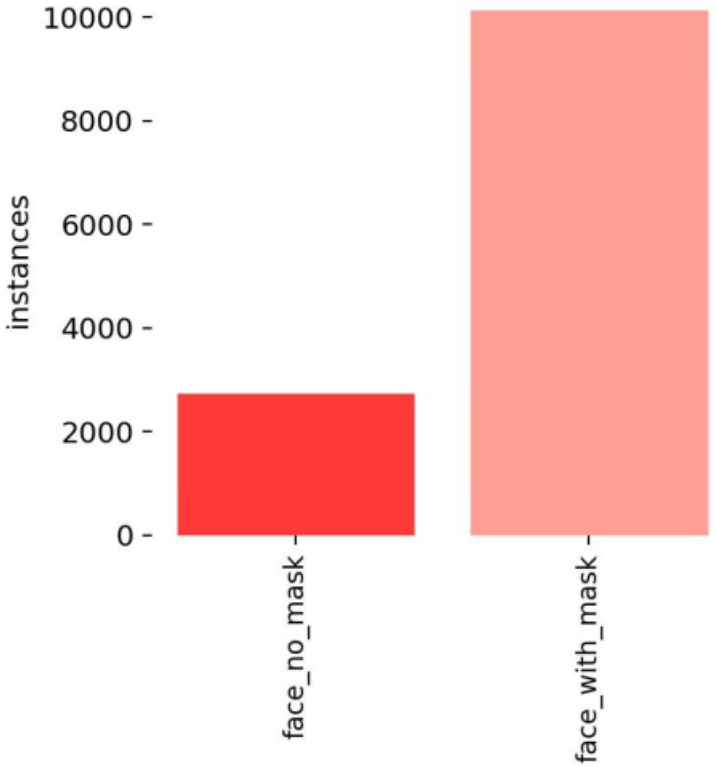


[Confusion Metrix]

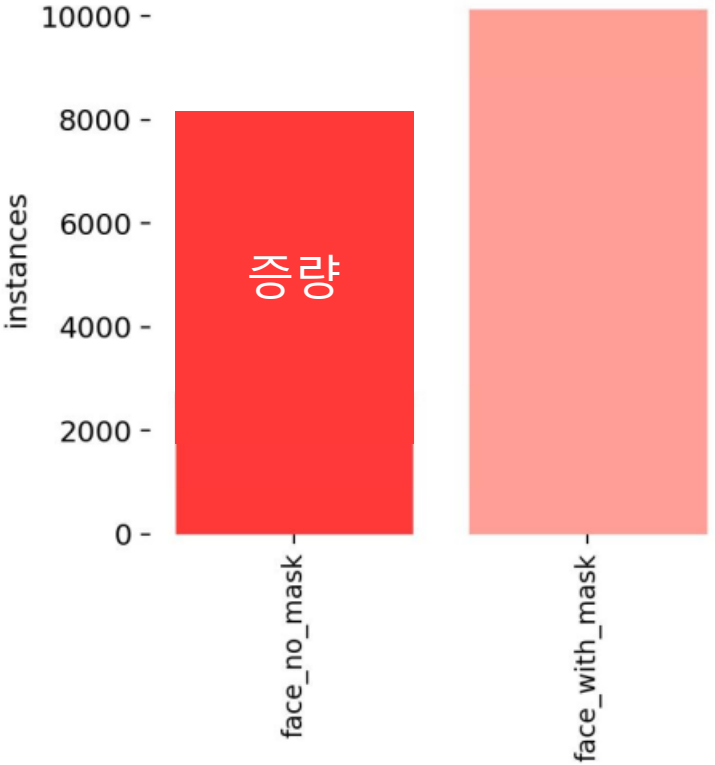
Class	Images	Instances	P	R	mAP50	mAP50-95
all	1000	3297	0.276	0.405	0.226	0.0555
face_no_mask	1000	2033	0.181	0.194	0.0878	0.0191
face_with_mask	1000	1264	0.37	0.615	0.365	0.0918

[검증 결과]

개선 사항



[주석 전처리 후]



[증량]

감사합니다