

지능화 캡스톤 프로젝트

프로젝트 #2 결과 발표

2022. 6. 15

충북대학교 산업인공지능학과

[21-7조] 봉은정, 김원우

수행방법 및 기여도

수행방법

- 각 팀원이 자유롭게 전체 프로젝트 구현
- 데이터셋 구성, 사용 모델 등 구현 방법 공유
- 테스트 결과 비교

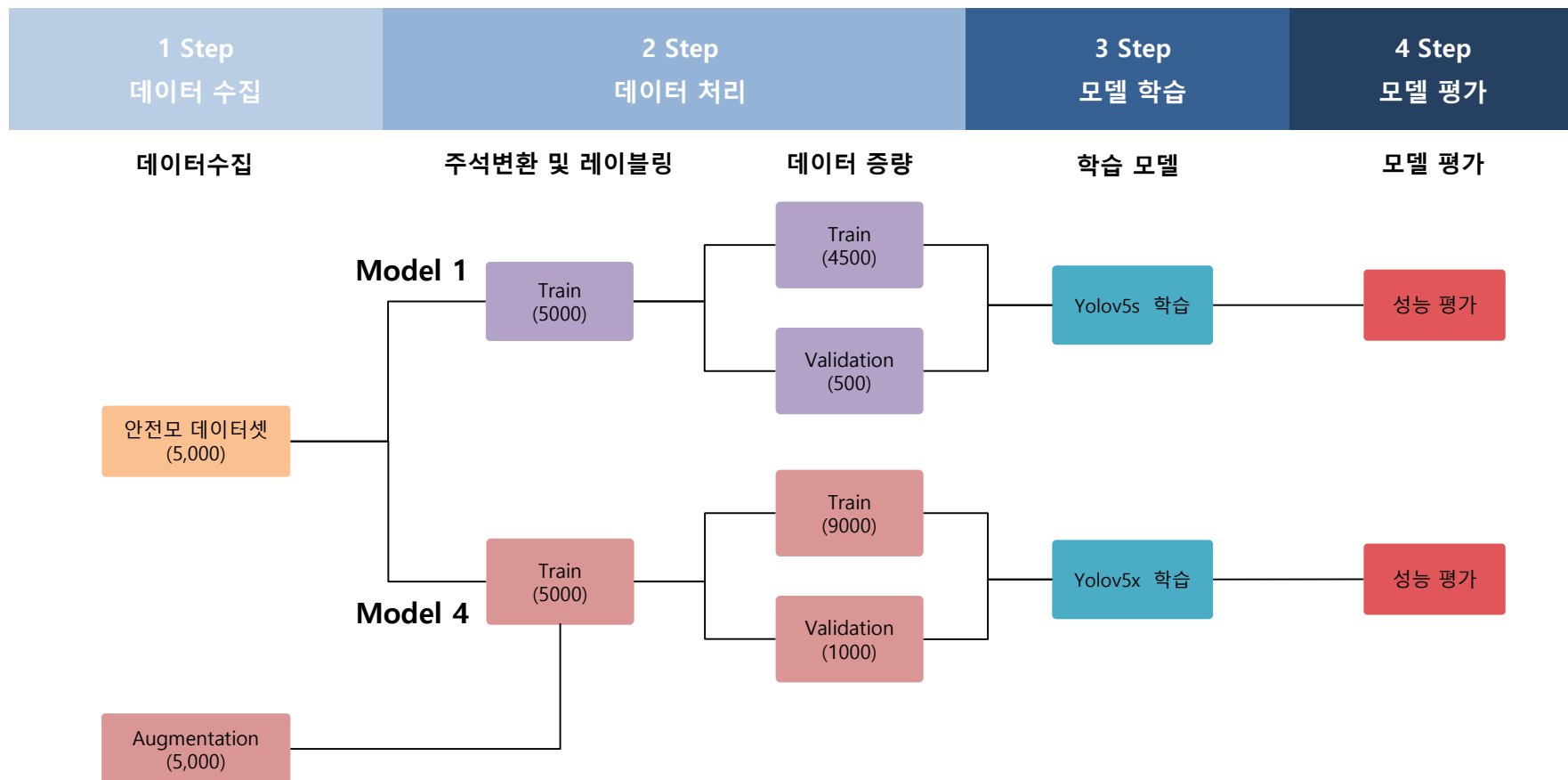
업무분장 및 기여도

이름	비중	수행내용			비고
		프로젝트#1	프로젝트#2	주제발표	
봉은정	50%	<ul style="list-style-type: none">• 데이터 증량• 코딩 및 학습• 자료 작성 및 발표	<ul style="list-style-type: none">• 데이터 레이블링• yolov5 모델 학습• 자료 작성 및 발표	<ul style="list-style-type: none">• yolov5 학습 방법 조사• 자료 작성 및 발표	
김원우	50%	<ul style="list-style-type: none">• 데이터 증량• 코딩 및 학습• 자료 작성 및 발표	<ul style="list-style-type: none">• 데이터 레이블링• yolov5 모델 학습• 자료 작성 및 발표	<ul style="list-style-type: none">• yolov5 추론, 검증 방법 조사• 자료 작성 및 발표	

모델 개발 프로세스

우수성/차별성

- 데이터셋 및 학습 모델을 변경해가며 모델 학습 진행
- 테스트 결과 mAP@.5, mAP@.5:.95가 높은 모델 2개 선정



데이터셋 구성

Data 레이블링

1) XML 파일의 정보를 txt 형식으로 변환

- 이미지 당 1개의 Annotation 파일 생성
- Python xml 모듈 이용하여 구문 분석
- Center, Width, Height를 0~1 사이의 값으로 치환

```
<object>
  <name>helmet</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <occluded>0</occluded>
  <difficult>0</difficult>
  <bndbox>
    <xmin>357</xmin>
    <ymin>116</ymin>
    <xmax>404</xmax>
    <ymax>175</ymax>
  </bndbox>
</object>
```

XML 파일

```
1 0.914663 0.349760 0.112981 0.141827
1 0.051683 0.396635 0.084135 0.091346
1 0.634615 0.379808 0.052885 0.091346
1 0.748798 0.391827 0.055288 0.086538
1 0.305288 0.397837 0.052885 0.069712
1 0.216346 0.397837 0.048077 0.069712
0 0.174279 0.379808 0.050481 0.067308
0 0.801683 0.383413 0.055288 0.088942
0 0.443510 0.411058 0.045673 0.072115
0 0.555288 0.400240 0.043269 0.074519
0 0.500000 0.383413 0.038462 0.064904
1 0.252404 0.360577 0.033654 0.048077
0 0.399038 0.393029 0.043269 0.064904
```

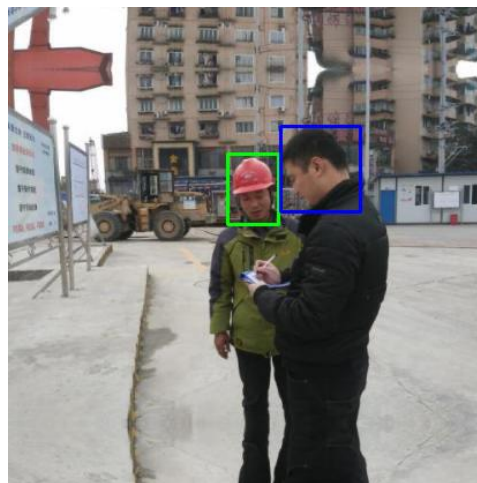
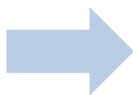
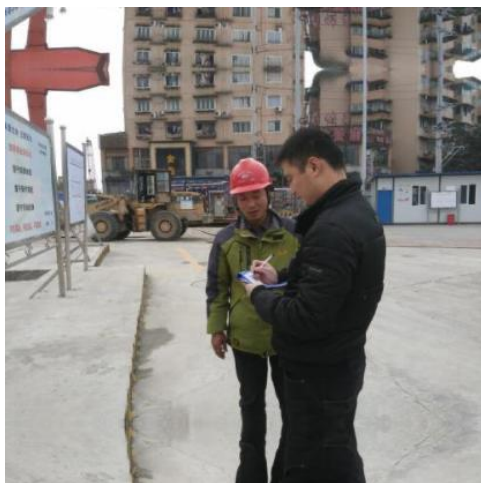
txt 파일

데이터셋 구성

Data 레이블링

2) 이미지 검수

- yolomark, Labellmg 등의 주석 도구 활용
- Person 클래스 제거
- 이미지 재라벨링



데이터셋 구성

Data Augmentation

- Python imgaug 모듈 활용
 - GaussianBlur 적용하여 데이터 5000장 추가

```
import imgaug.augmenters as iaa

class Augmentation:
    def __init__(self, origPath, destPath):
        self.origPath = origPath
        self.destPath = destPath
        self.fileList = os.listdir(self.origPath)

    # Augment
    self.blur = iaa.GaussianBlur(sigma=(1.0, 1.2)).to_deterministic()

    def Blur(self):
        for f in self.fileList:
            if f.endswith(".png"):
                Image = cv2.imread(self.origPath + f)
                BlurImage = self.blur.augment_images([Image])[0]
                cv2.imwrite(self.destPath + 'blur_' + f, BlurImage)

                textName = f.split(".png")[0] + '.txt'
                Text = self.origPath + textName
                BlurText = self.destPath + 'blur_' + textName
                shutil.copy(Text, BlurText)

def Main():
    origPath = "dataset/Project2_Dataset/"
    destPath = "dataset/Project2_Dataset_blur/"
    Augmentation(origPath, destPath).Blur()
```

코드



Blur 이미지

모델 학습

딥러닝 학습 환경

- PC 사양
 - CPU : Intel core 10GEN, i9-10900K @ 3.70GHz
 - RAM : 32GB
 - GPU : NVIDIA GeForce RTX 3090 24GB X 2
- requirements.txt 목록 설치
 - Pytorch = 1.7.0 / Tensorflow = 2.4.1
- UBUNTU 18.04 / 20.04
- Anaconda , Virtualenv 가상환경 사용

모델 학습

학습 방법

	Model 1	Model 2	Model 3	Model 4
Epochs	30	200	240	200
Batch size	16	128	128	64
architecture	YOLOv5s	YOLOv5s	YOLOv5m	YOLOv5x
Layers	213	276	290	444

- Layer 수 조정 방법

- Depth multiple의 값을 수정하여 CSP 모듈의 반복 횟수를 조정

YOLOv5s

```
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# Parameters
nc: 2 # number of classes
depth_multiple: 0.53 # model depth multiple
width_multiple: 0.50 # layer channel multiple
```

YOLOv5m

```
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# Parameters
nc: 2 # number of classes
depth_multiple: 0.85 # model depth multiple
width_multiple: 0.75 # layer channel multiple
```