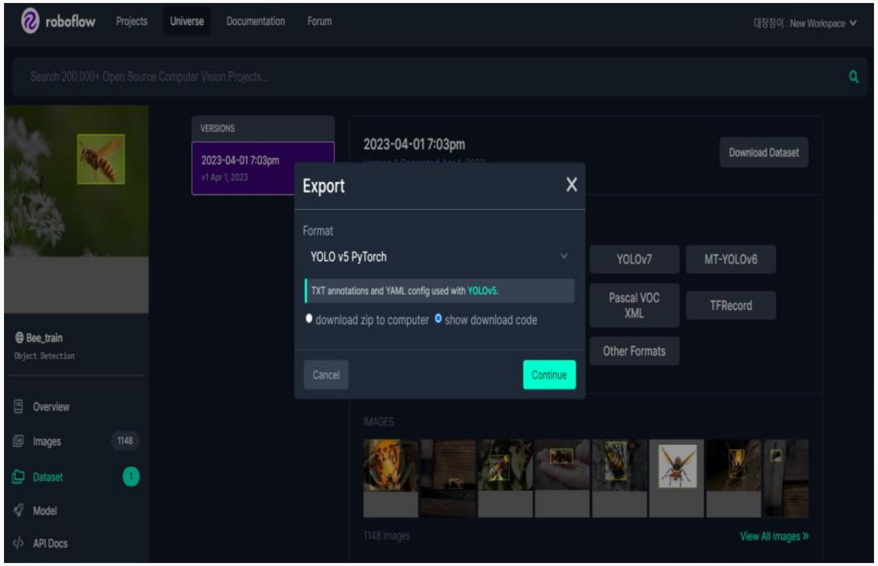
# 수집된 이미지를 바탕으로 객체인식 실습

**이준혁 (2023250419)**
산업인공지능학과

# 진행순서

1. Datasets 가져오기
2. Yolo 클론 설치 및 프로그램 패키지 설치
3. 이미지 저장 리스트생성 및 이미지 개수확인
4. 트레이닝셋과 밸리데이션 셋 나누기
5. 위치 변경
6. 트레이닝 시키기
7. 트레이닝 완료하고 확인하기
8. 이미지 넣고 추론하기
9. 내가 원하는 이미지를 넣고 확인하기

# 1. 데이터셋 가져오기 (1)



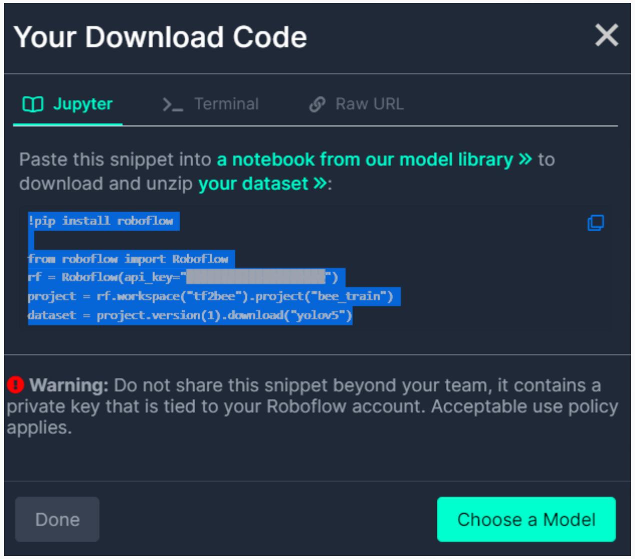
Export

Format

YOLO v5 PyTorch

TXT annotations and YAML config used with YOLOv5.

○ download zip to computer ● show download code



Your Download Code

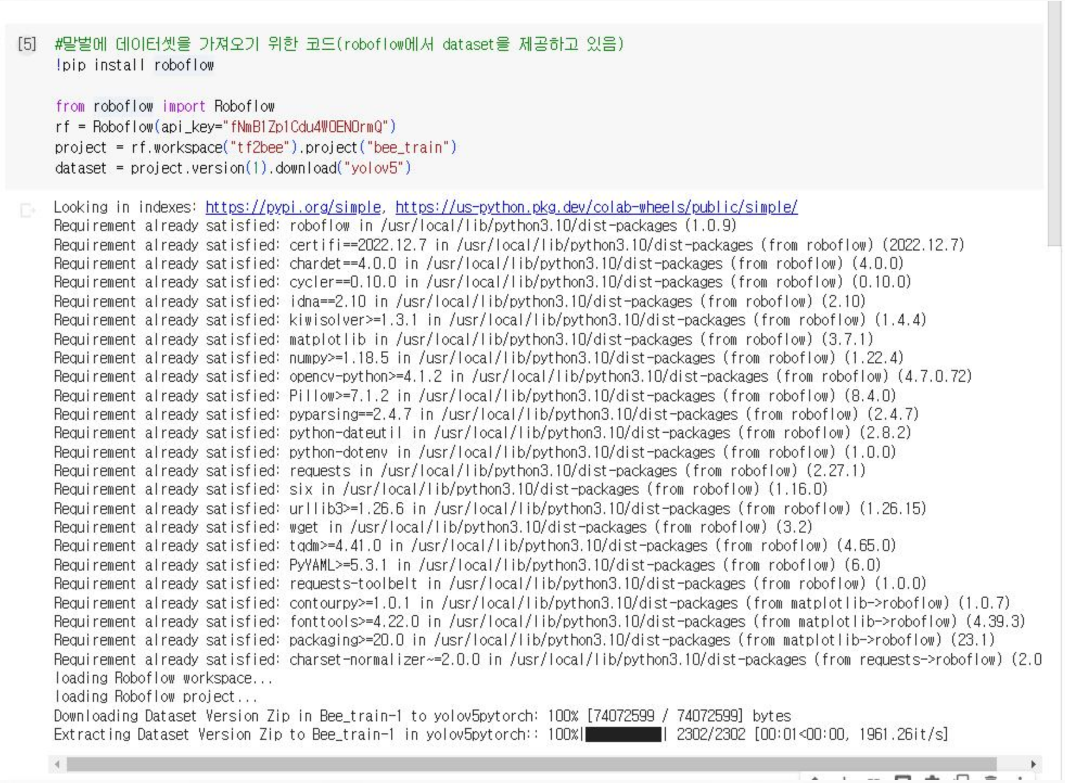
Paste this snippet into a notebook from our model library » to download and unzip your dataset »:

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="[redacted]")
project = rf.workspace("tf2bee").project("bee_train")
dataset = project.version(1).download("yolov5")
```

**Warning:** Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

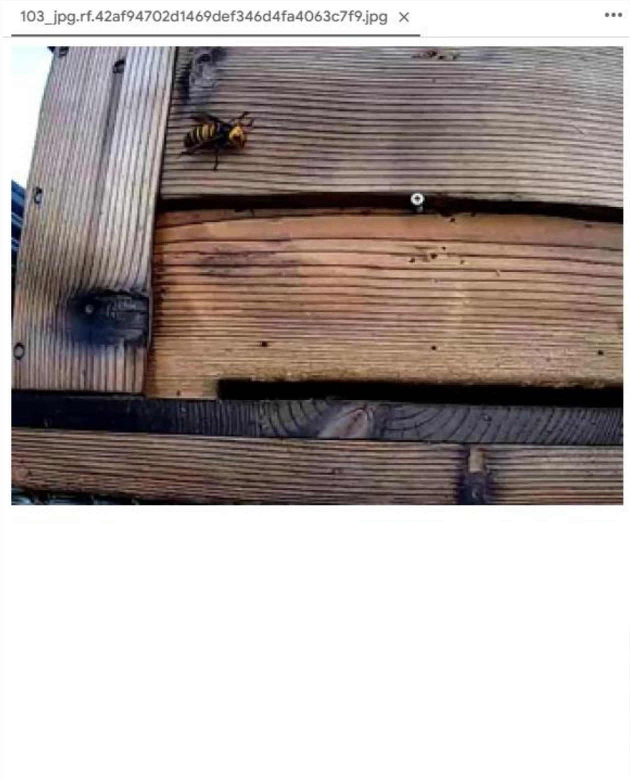
로보플로우사이트에서 데이터셋 구하기

# 1. 데이터셋 가져오기 (2)



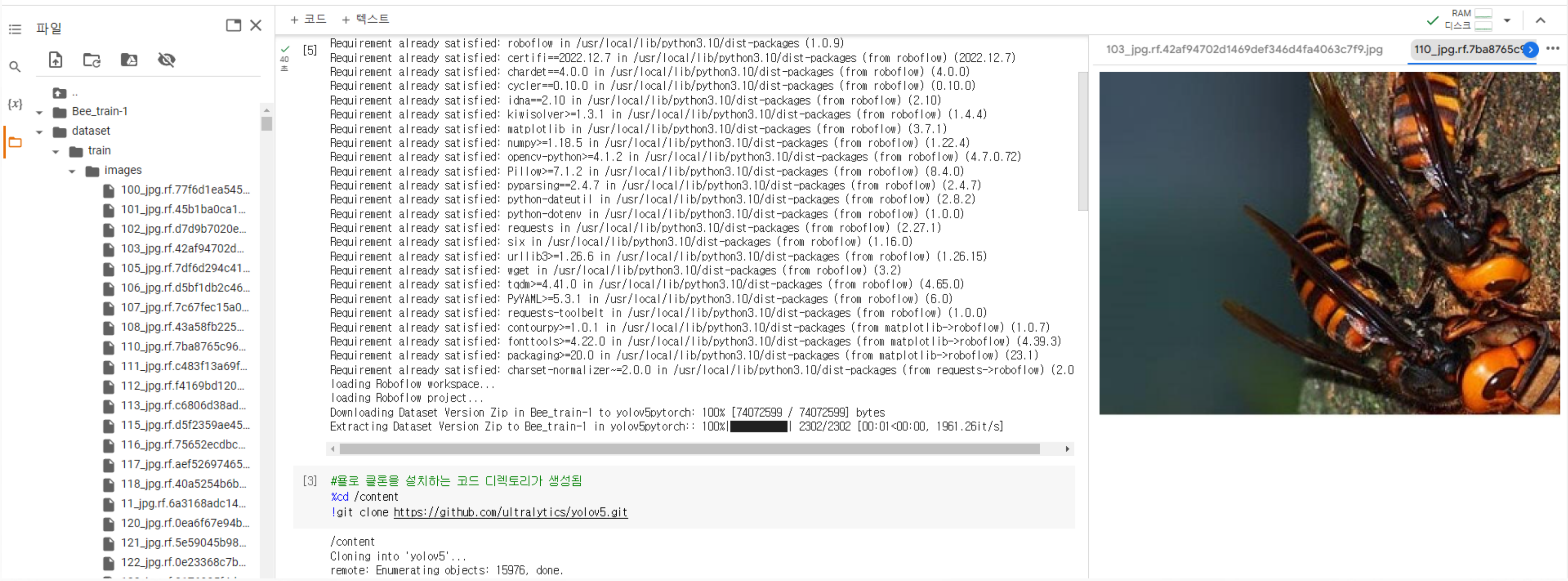
```
[5] #말벌에 데이터셋을 가져오기 위한 코드(roboflow에서 dataset을 제공하고 있음)
    !pip install roboflow

    from roboflow import Roboflow
    rf = Roboflow(api_key="fNmB1Zp1Cdu4WOENOrmQ")
    project = rf.workspace("tf2bee").project("bee_train")
    dataset = project.version(1).download("yolov5")
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: roboflow in /usr/local/lib/python3.10/dist-packages (1.0.9)
Requirement already satisfied: certifi==2022.12.7 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2022.12.7)
Requirement already satisfied: chardet==4.0.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.0.0)
Requirement already satisfied: cycler==0.10.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.10.0)
Requirement already satisfied: idna==2.10 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.10)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.22.4)
Requirement already satisfied: opencv-python>=4.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.7.0.72)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (8.4.0)
Requirement already satisfied: pyparsing==2.4.7 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.4.7)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.27.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.26.15)
Requirement already satisfied: wget in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.2)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.65.0)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.0.7)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.39.3)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (23.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (2.0
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Bee_train-1 to yolov5pytorch: 100% [74072599 / 74072599] bytes
Extracting Dataset Version Zip to Bee_train-1 in yolov5pytorch:: 100%|██████████| 2302/2302 [00:01<00:00, 1961.26it/s]
```



데이터셋 주소 입력하고 이미지파일 확인

# 1. 데이터셋 가져오기 (3)



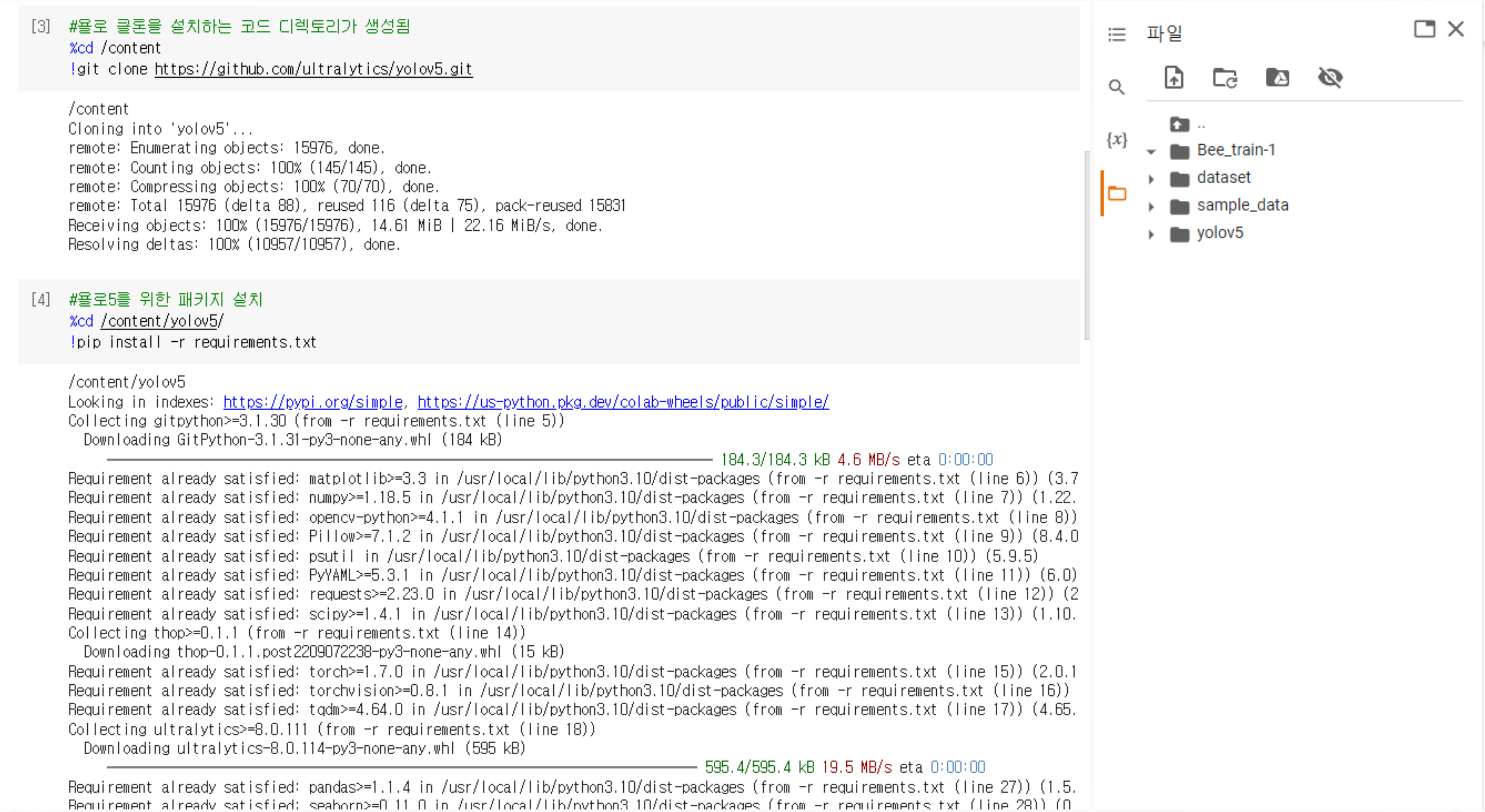
```
[5] Requirement already satisfied: roboflow in /usr/local/lib/python3.10/dist-packages (1.0.9)
    Requirement already satisfied: certifi==2022.12.7 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2022.12.7)
    Requirement already satisfied: chardet==4.0.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.0.0)
    Requirement already satisfied: cycler==0.10.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.10.0)
    Requirement already satisfied: idna==2.10 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.10)
    Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.4)
    Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7.1)
    Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.22.4)
    Requirement already satisfied: opencv-python>=4.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.7.0.72)
    Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (8.4.0)
    Requirement already satisfied: pyparsing==2.4.7 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.4.7)
    Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
    Requirement already satisfied: python-dotenv in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.27.1)
    Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
    Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.26.15)
    Requirement already satisfied: wget in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.2)
    Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.65.0)
    Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0)
    Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
    Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.0.7)
    Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.39.3)
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (23.1)
    Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (2.0
    loading Roboflow workspace...
    loading Roboflow project...
    Downloading Dataset Version Zip in Bee_train-1 to yolov5pytorch: 100% [74072599 / 74072599] bytes
    Extracting Dataset Version Zip to Bee_train-1 in yolov5pytorch:: 100%|██████████| 2302/2302 [00:01<00:00, 1961.26it/s]
```

```
[3] #욜로 클론을 설치하는 코드 디렉토리가 생성됨
    %cd /content
    !git clone https://github.com/ultralytics/yolov5.git
```

```
/content
Cloning into 'yolov5'...
remote: Enumerating objects: 15976, done.
```

데이터셋 주소 입력하고 이미지파일 확인

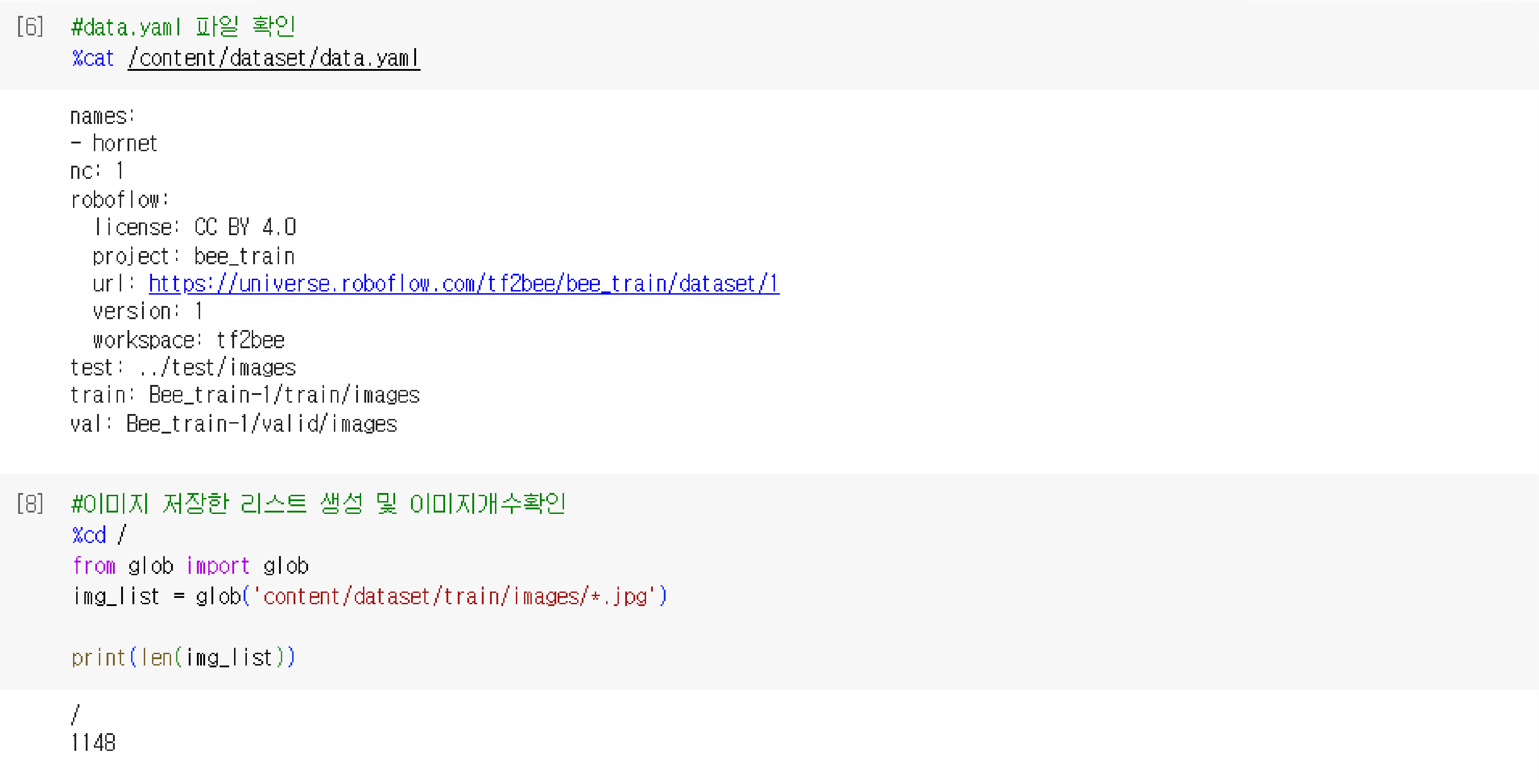
# 2. 욜로 환경 설치



```
[3] #욜로 클론을 설치하는 코드 디렉토리가 생성됨
    %cd /content
    !git clone https://github.com/ultralytics/yolov5.git

    /content
    Cloning into 'yolov5'...
    remote: Enumerating objects: 15976, done.
    remote: Counting objects: 100% (145/145), done.
    remote: Compressing objects: 100% (70/70), done.
    remote: Total 15976 (delta 88), reused 116 (delta 75), pack-reused 15831
    Receiving objects: 100% (15976/15976), 14.61 MiB | 22.16 MiB/s, done.
    Resolving deltas: 100% (10957/10957), done.

[4] #욜로5를 위한 패키지 설치
    %cd /content/yolov5/
    !pip install -r requirements.txt

    /content/yolov5
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Collecting gitpython>=3.1.30 (from -r requirements.txt (line 5))
      Downloading GitPython-3.1.31-py3-none-any.whl (184 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 184.3/184.3 kB 4.6 MB/s eta 0:00:00
    Requirement already satisfied: matplotlib>=3.3 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 6)) (3.7
    Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 7)) (1.22.
    Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 8))
    Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 9)) (8.4.0
    Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 10)) (5.9.5)
    Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 11)) (6.0)
    Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 12)) (2
    Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 13)) (1.10.
    Collecting thop>=0.1.1 (from -r requirements.txt (line 14))
      Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
    Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 15)) (2.0.1
    Requirement already satisfied: torchvision>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 16))
    Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 17)) (4.65.
    Collecting ultralytics>=8.0.111 (from -r requirements.txt (line 18))
      Downloading ultralytics-8.0.114-py3-none-any.whl (595 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 595.4/595.4 kB 19.5 MB/s eta 0:00:00
    Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 27)) (1.5.
    Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 28)) (0
```

욜로 클론 설치, 패키지 설치

# 3. 이미지저장 리스트생성 및 이미지 개수확인



```
[6]  #data.yaml 파일 확인
     %cat /content/dataset/data.yaml
```

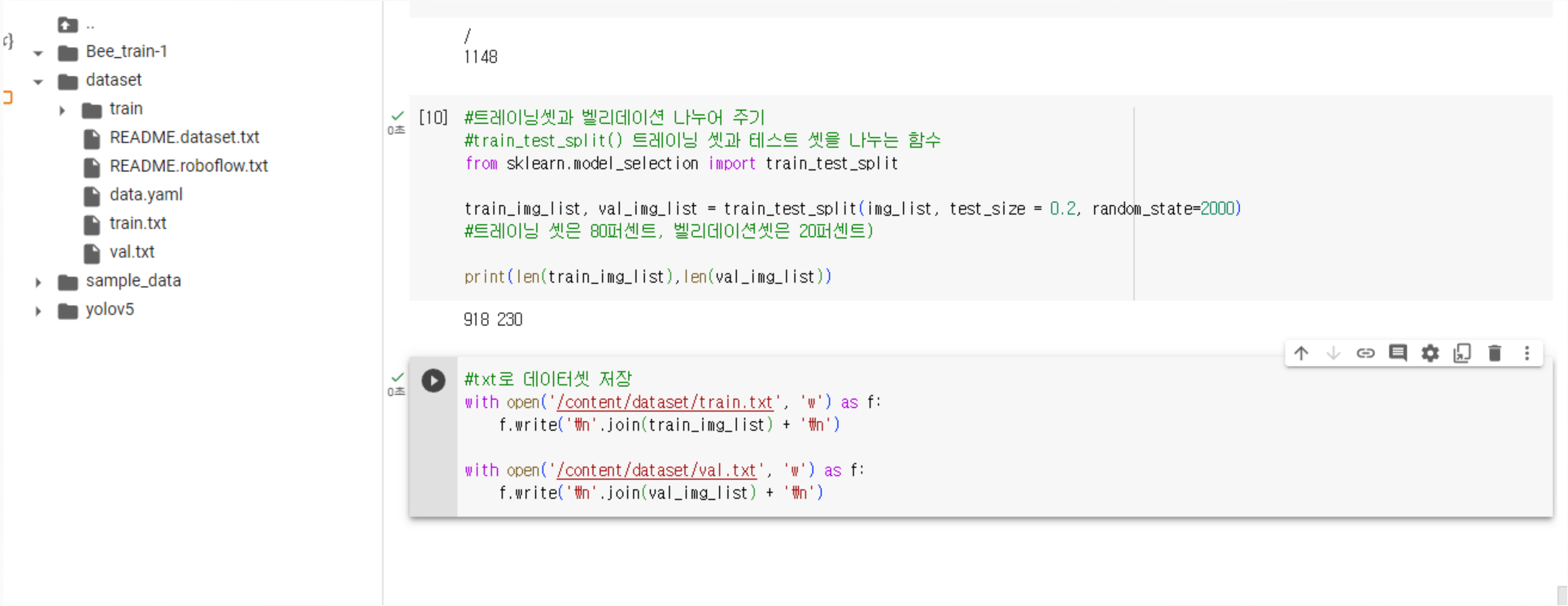
```
names:
- hornet
nc: 1
roboflow:
  license: CC BY 4.0
  project: bee_train
  url: https://universe.roboflow.com/tf2bee/bee_train/dataset/1
  version: 1
  workspace: tf2bee
test: ../test/images
train: Bee_train-1/train/images
val: Bee_train-1/valid/images
```

```
[8]  #이미지 저장한 리스트 생성 및 이미지개수확인
     %cd /
     from glob import glob
     img_list = glob('content/dataset/train/images/*.jpg')

     print(len(img_list))
```

```
/
1148
```

이미지 리스트 및 개수확인

# 4. 트레이닝셋과 밸리데이션 셋나누기 및 텍스트 파일로 데이터셋 저장



```
/
1148
```

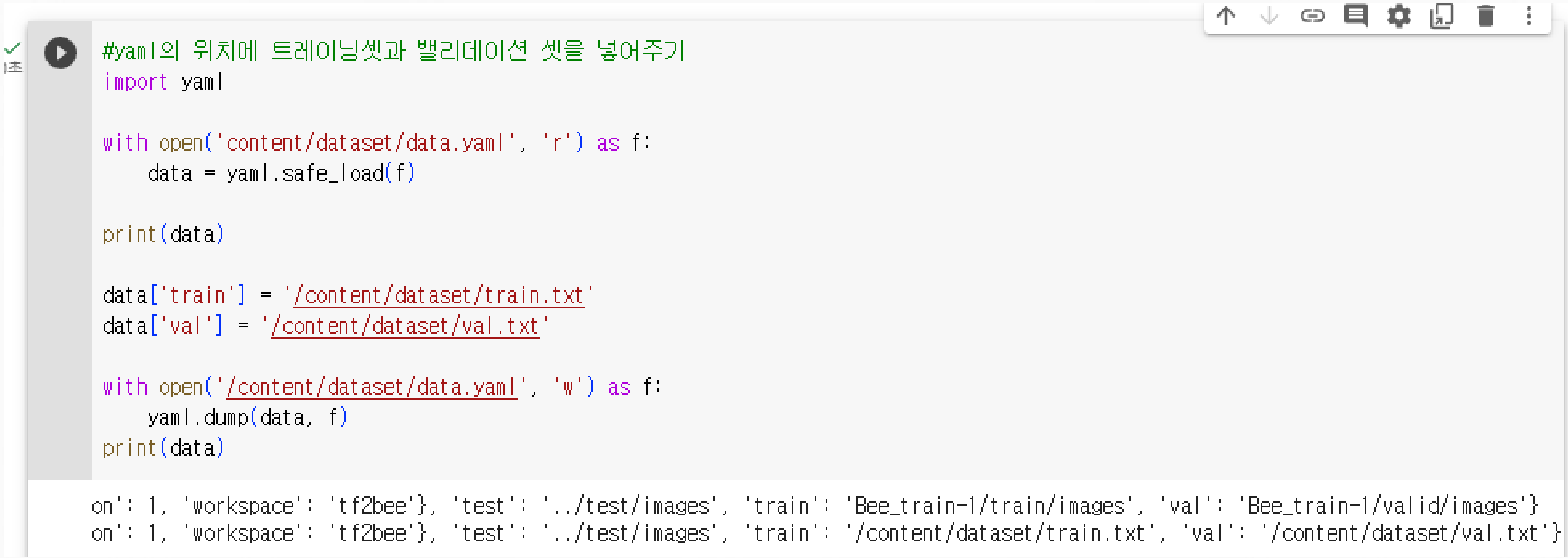
```
[10] #트레이닝셋과 벨리데이션 나누어 주기
     #train_test_split() 트레이닝 셋과 테스트 셋을 나누는 함수
     from sklearn.model_selection import train_test_split

     train_img_list, val_img_list = train_test_split(img_list, test_size = 0.2, random_state=2000)
     #트레이닝 셋은 80퍼센트, 벨리데이션셋은 20퍼센트)

     print(len(train_img_list),len(val_img_list))
```

```
918 230
```

```
#txt로 데이터셋 저장
with open('/content/dataset/train.txt', 'w') as f:
    f.write('₩n'.join(train_img_list) + '₩n')

with open('/content/dataset/val.txt', 'w') as f:
    f.write('₩n'.join(val_img_list) + '₩n')
```

Train, val 값 TXT로 저장

# 5. 위치변경

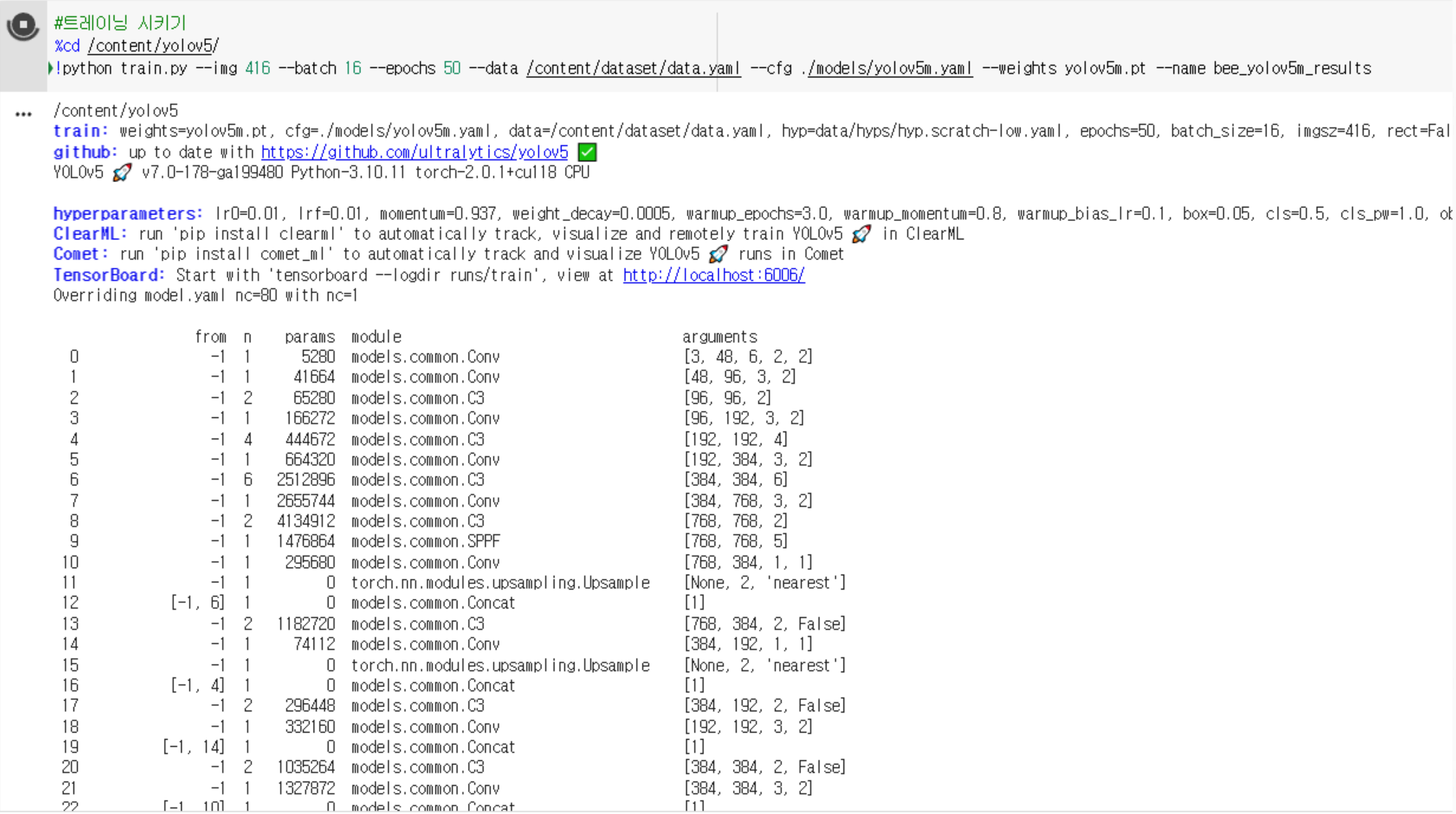


```python
#yaml의 위치에 트레이닝셋과 밸리데이션 셋을 넣어주기
import yaml

with open('content/dataset/data.yaml', 'r') as f:
    data = yaml.safe_load(f)

print(data)

data['train'] = '/content/dataset/train.txt'
data['val'] = '/content/dataset/val.txt'

with open('/content/dataset/data.yaml', 'w') as f:
    yaml.dump(data, f)
print(data)
```

```
on': 1, 'workspace': 'tf2bee'}, 'test': '../test/images', 'train': 'Bee_train-1/train/images', 'val': 'Bee_train-1/valid/images'}
on': 1, 'workspace': 'tf2bee'}, 'test': '../test/images', 'train': '/content/dataset/train.txt', 'val': '/content/dataset/val.txt'}
```

트레이닝 시키기 위해 위치지정

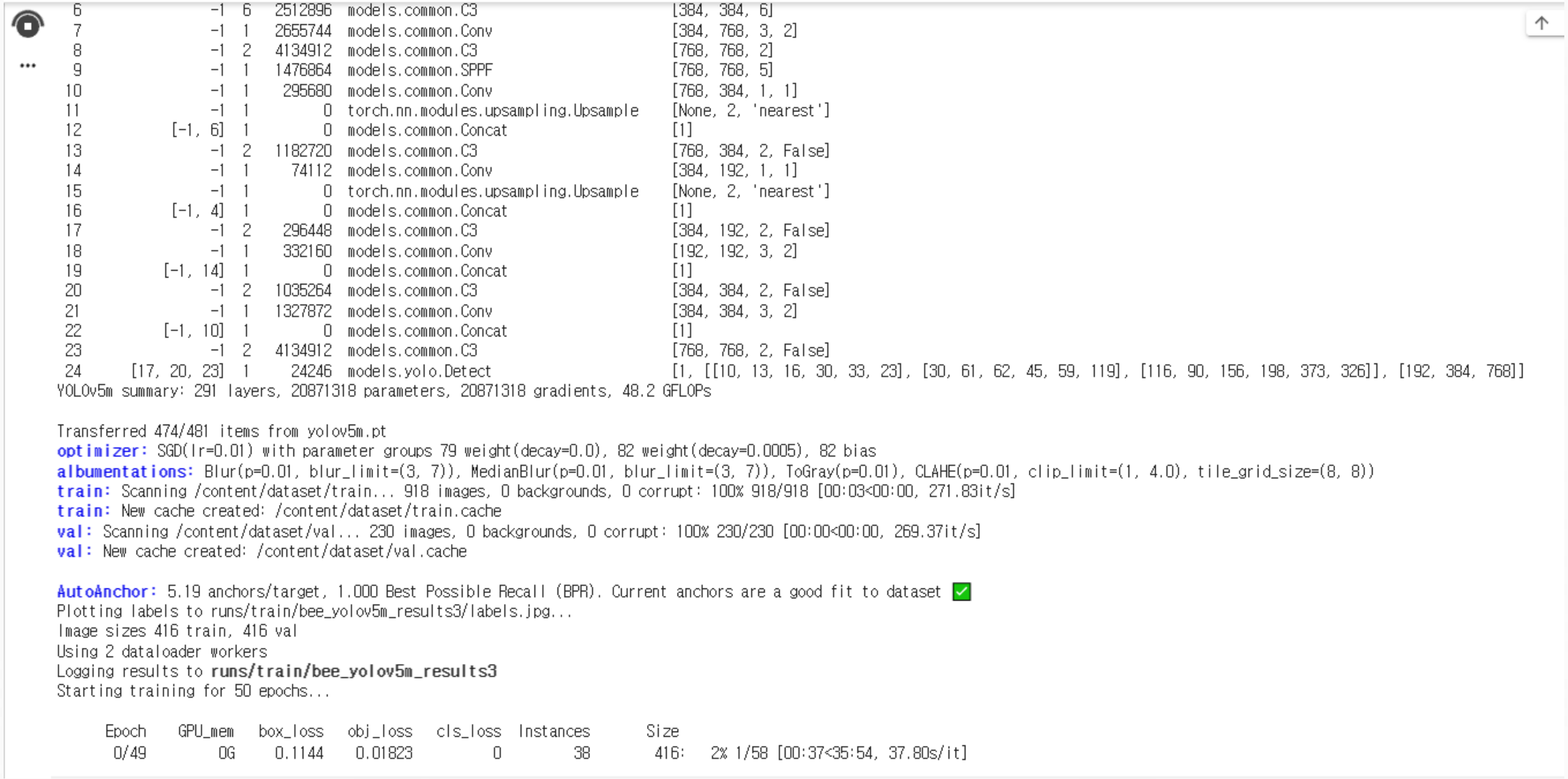
# 6. 트레이닝 시키기



```
#트레이닝 시키기
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 50 --data /content/dataset/data.yaml --cfg ./models/yolov5m.yaml --weights yolov5m.pt --name bee_yolov5m_results
```

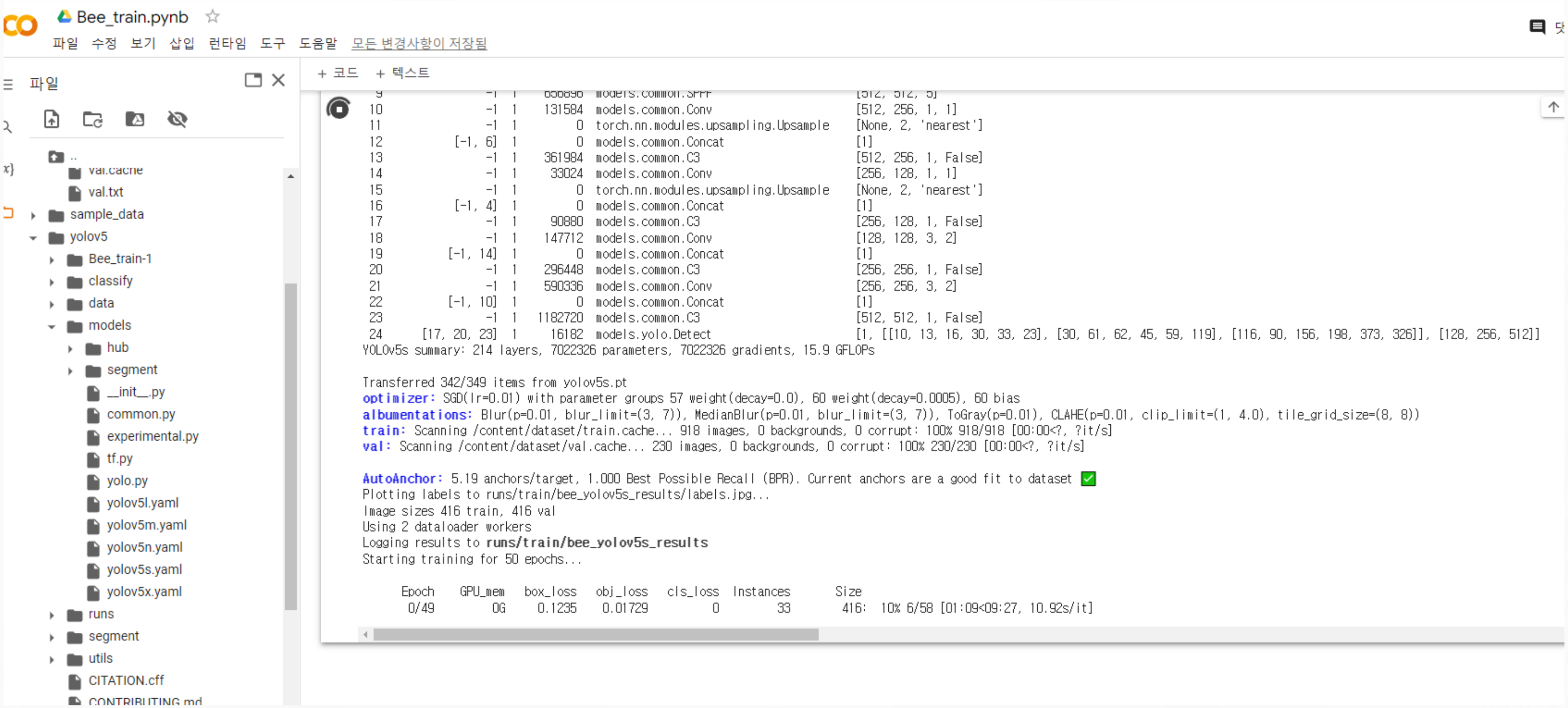
```
/content/yolov5
train: weights=yolov5m.pt, cfg=./models/yolov5m.yaml, data=/content/dataset/data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=50, batch_size=16, imgsz=416, rect=Fal
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v7.0-178-ga199480 Python-3.10.11 torch-2.0.1+cu118 CPU

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, ob
ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 🚀 in ClearML
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 🚀 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
Overriding model.yaml nc=80 with nc=1

                 from  n    params  module                                  arguments
  0                -1  1      5280  models.common.Conv                      [3, 48, 6, 2, 2]
  1                -1  1     41664  models.common.Conv                      [48, 96, 3, 2]
  2                -1  2     65280  models.common.C3                        [96, 96, 2]
  3                -1  1    166272  models.common.Conv                      [96, 192, 3, 2]
  4                -1  4    444672  models.common.C3                        [192, 192, 4]
  5                -1  1    664320  models.common.Conv                      [192, 384, 3, 2]
  6                -1  6   2512896  models.common.C3                        [384, 384, 6]
  7                -1  1   2655744  models.common.Conv                      [384, 768, 3, 2]
  8                -1  2   4134912  models.common.C3                        [768, 768, 2]
  9                -1  1   1476864  models.common.SPPF                      [768, 768, 5]
 10                -1  1    295680  models.common.Conv                      [768, 384, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  2   1182720  models.common.C3                        [768, 384, 2, False]
 14                -1  1     74112  models.common.Conv                      [384, 192, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  2    296448  models.common.C3                        [384, 192, 2, False]
 18                -1  1    332160  models.common.Conv                      [192, 192, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  2   1035264  models.common.C3                        [384, 384, 2, False]
 21                -1  1   1327872  models.common.Conv                      [384, 384, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
```

# 6. 트레이닝 시키기



```
  6                -1  6   2512896  models.common.C3                        [384, 384, 6]
  7                -1  1   2655744  models.common.Conv                      [384, 768, 3, 2]
  8                -1  2   4134912  models.common.C3                        [768, 768, 2]
  9                -1  1   1476864  models.common.SPPF                      [768, 768, 5]
 10                -1  1    295680  models.common.Conv                      [768, 384, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  2   1182720  models.common.C3                        [768, 384, 2, False]
 14                -1  1     74112  models.common.Conv                      [384, 192, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  2    296448  models.common.C3                        [384, 192, 2, False]
 18                -1  1    332160  models.common.Conv                      [192, 192, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  2   1035264  models.common.C3                        [384, 384, 2, False]
 21                -1  1   1327872  models.common.Conv                      [384, 384, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  2   4134912  models.common.C3                        [768, 768, 2, False]
 24      [17, 20, 23]  1     24246  models.yolo.Detect                      [1, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [192, 384, 768]]
YOLOv5m summary: 291 layers, 20871318 parameters, 20871318 gradients, 48.2 GFLOPs

Transferred 474/481 items from yolov5m.pt
optimizer: SGD(lr=0.01) with parameter groups 79 weight(decay=0.0), 82 weight(decay=0.0005), 82 bias
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
train: Scanning /content/dataset/train... 918 images, 0 backgrounds, 0 corrupt: 100% 918/918 [00:03<00:00, 271.83it/s]
train: New cache created: /content/dataset/train.cache
val: Scanning /content/dataset/val... 230 images, 0 backgrounds, 0 corrupt: 100% 230/230 [00:00<00:00, 269.37it/s]
val: New cache created: /content/dataset/val.cache

AutoAnchor: 5.19 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✅
Plotting labels to runs/train/bee_yolov5m_results3/labels.jpg... 
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/bee_yolov5m_results3
Starting training for 50 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       0/49         0G     0.1144    0.01823          0         38        416:   2% 1/58 [00:37<35:54, 37.80s/it]
```

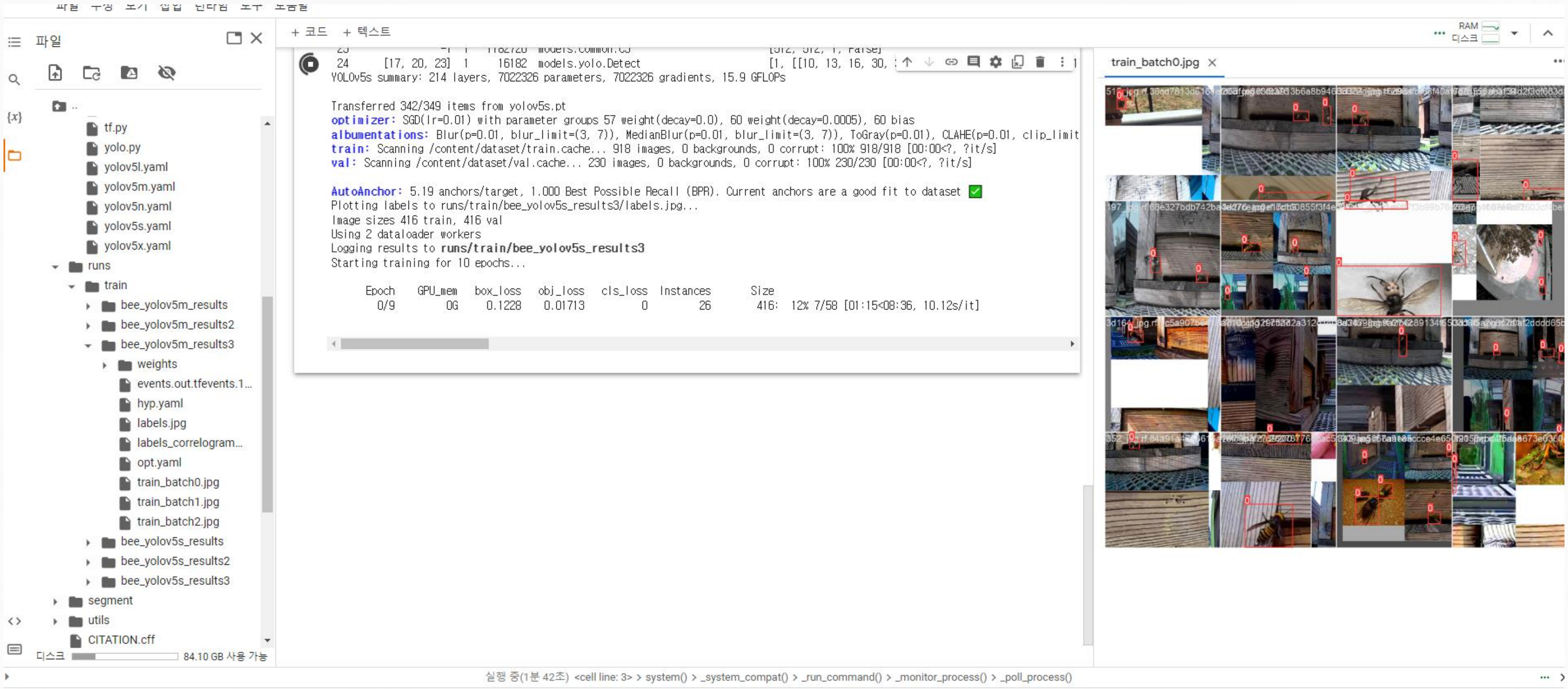
# 6. 트레이닝 시키기 모델 s로 변경, 학습 수 10번으로 변경



```
  9                -1  1    656896  models.common.SPPF                      [512, 512, 5]
 10                -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat                    [1]
 13                -1  1    361984  models.common.C3                        [512, 256, 1, False]
 14                -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat                    [1]
 17                -1  1     90880  models.common.C3                        [256, 128, 1, False]
 18                -1  1    147712  models.common.Conv                      [128, 128, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat                    [1]
 20                -1  1    296448  models.common.C3                        [256, 256, 1, False]
 21                -1  1    590336  models.common.Conv                      [256, 256, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat                    [1]
 23                -1  1   1182720  models.common.C3                        [512, 512, 1, False]
 24      [17, 20, 23]  1     16182  models.yolo.Detect                      [1, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]
YOLOv5s summary: 214 layers, 7022326 parameters, 7022326 gradients, 15.9 GFLOPs

Transferred 342/349 items from yolov5s.pt
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
train: Scanning /content/dataset/train.cache... 918 images, 0 backgrounds, 0 corrupt: 100% 918/918 [00:00<?, ?it/s]
val: Scanning /content/dataset/val.cache... 230 images, 0 backgrounds, 0 corrupt: 100% 230/230 [00:00<?, ?it/s]

AutoAnchor: 5.19 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✅
Plotting labels to runs/train/bee_yolov5s_results/labels.jpg...
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/bee_yolov5s_results
Starting training for 50 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       0/49         0G     0.1235    0.01729          0         33        416:  10% 6/58 [01:09<09:27, 10.92s/it]
```

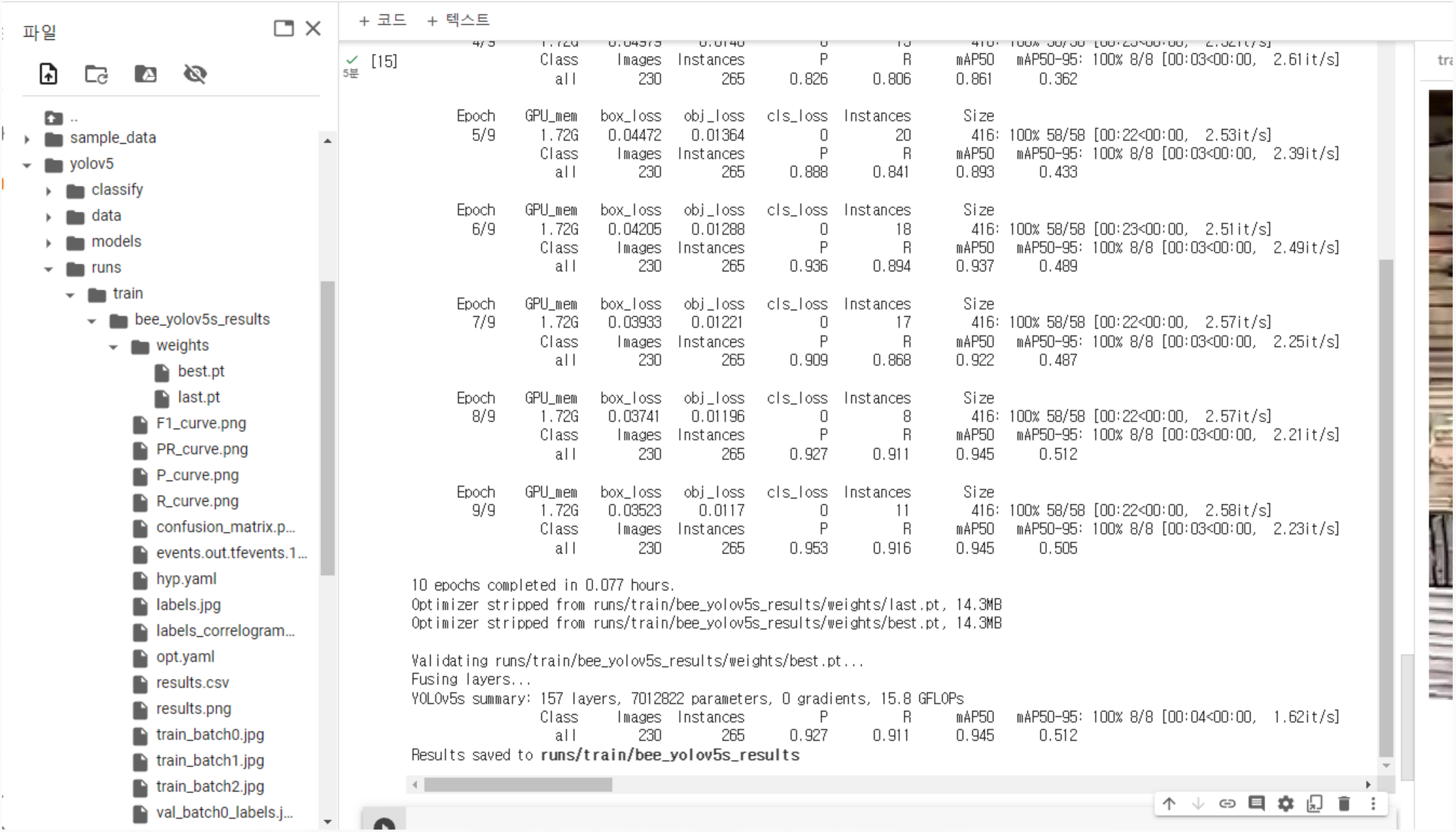
속도향상을 위해 코랩 GPU 활용 및 모델사이즈 s로 변경, 학습 수 10번으로 변경

# 6. 트레이닝 모델 배치활동 확인



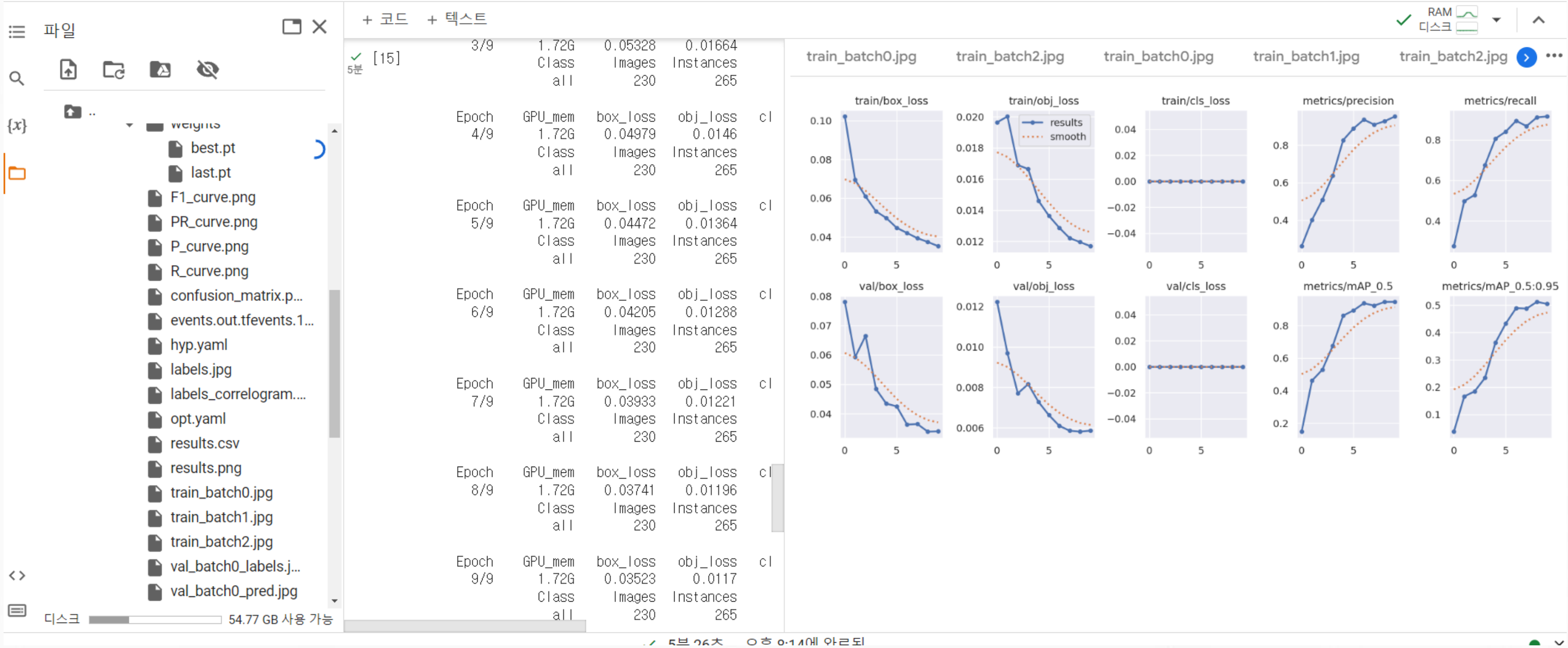
```
 24      [17, 20, 23]  1     16182  models.yolo.Detect                      [1, [[10, 13, 16, 30,
YOLOv5s summary: 214 layers, 7022326 parameters, 7022326 gradients, 15.9 GFLOPs

Transferred 342/349 items from yolov5s.pt
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit
train: Scanning /content/dataset/train.cache... 918 images, 0 backgrounds, 0 corrupt: 100% 918/918 [00:00<?, ?it/s]
val: Scanning /content/dataset/val.cache... 230 images, 0 backgrounds, 0 corrupt: 100% 230/230 [00:00<?, ?it/s]

AutoAnchor: 5.19 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✅
Plotting labels to runs/train/bee_yolov5s_results3/labels.jpg...
Image sizes 416 train, 416 val
Using 2 dataloader workers
Logging results to runs/train/bee_yolov5s_results3
Starting training for 10 epochs...

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
        0/9         0G     0.1228    0.01713          0         26        416:  12% 7/58 [01:15<08:36, 10.12s/it]
```

# 7. 트레이닝 완료하고 확인하기



```
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:03<00:00,  2.61it/s]
                   all        230        265      0.826      0.806      0.861      0.362

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
        5/9      1.72G    0.04472    0.01364          0         20        416: 100% 58/58 [00:22<00:00,  2.53it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:03<00:00,  2.39it/s]
                   all        230        265      0.888      0.841      0.893      0.433

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
        6/9      1.72G    0.04205    0.01288          0         18        416: 100% 58/58 [00:23<00:00,  2.51it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:03<00:00,  2.49it/s]
                   all        230        265      0.936      0.894      0.937      0.489

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
        7/9      1.72G    0.03933    0.01221          0         17        416: 100% 58/58 [00:22<00:00,  2.57it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:03<00:00,  2.25it/s]
                   all        230        265      0.909      0.868      0.922      0.487

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
        8/9      1.72G    0.03741    0.01196          0          8        416: 100% 58/58 [00:22<00:00,  2.57it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:03<00:00,  2.21it/s]
                   all        230        265      0.927      0.911      0.945      0.512

      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
        9/9      1.72G    0.03523     0.0117          0         11        416: 100% 58/58 [00:22<00:00,  2.58it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:03<00:00,  2.23it/s]
                   all        230        265      0.953      0.916      0.945      0.505

10 epochs completed in 0.077 hours.
Optimizer stripped from runs/train/bee_yolov5s_results/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/bee_yolov5s_results/weights/best.pt, 14.3MB

Validating runs/train/bee_yolov5s_results/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 8/8 [00:04<00:00,  1.62it/s]
                   all        230        265      0.927      0.911      0.945      0.512
Results saved to runs/train/bee_yolov5s_results
```

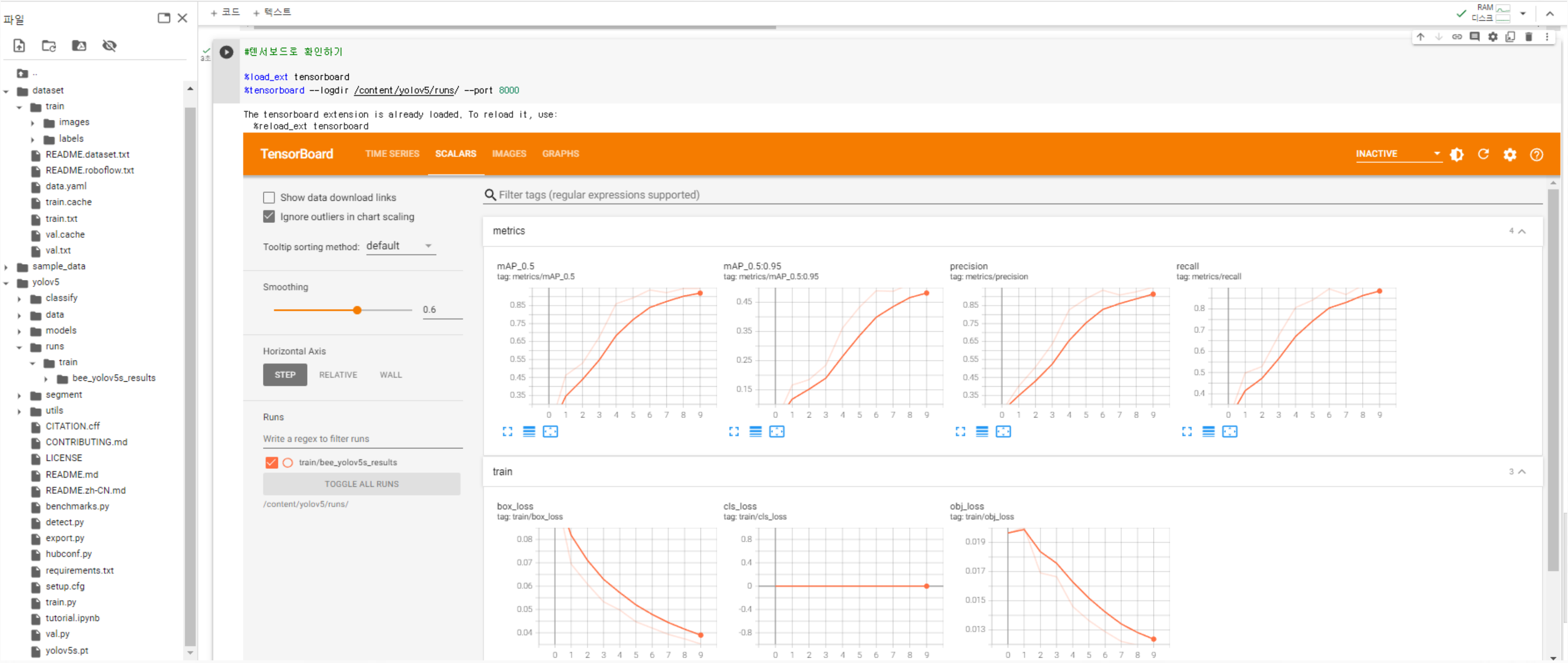
# 7. 트레이닝 완료하고 확인하기



```
                3/9      1.72G    0.05328    0.01664
              Class     Images  Instances
                all        230        265

              Epoch    GPU_mem   box_loss   obj_loss   cl
                4/9      1.72G    0.04979     0.0146
              Class     Images  Instances
                all        230        265

              Epoch    GPU_mem   box_loss   obj_loss   cl
                5/9      1.72G    0.04472    0.01364
              Class     Images  Instances
                all        230        265

              Epoch    GPU_mem   box_loss   obj_loss   cl
                6/9      1.72G    0.04205    0.01288
              Class     Images  Instances
                all        230        265

              Epoch    GPU_mem   box_loss   obj_loss   cl
                7/9      1.72G    0.03933    0.01221
              Class     Images  Instances
                all        230        265

              Epoch    GPU_mem   box_loss   obj_loss   cl
                8/9      1.72G    0.03741    0.01196
              Class     Images  Instances
                all        230        265

              Epoch    GPU_mem   box_loss   obj_loss   cl
                9/9      1.72G    0.03523     0.0117
              Class     Images  Instances
                all        230        265
```

결과값 확인

# 7. 트레이닝 완료하고 확인하기



```
#텐서보드로 확인하기

%load_ext tensorboard
%tensorboard --logdir /content/yolov5/runs/ --port 8000
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```

□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

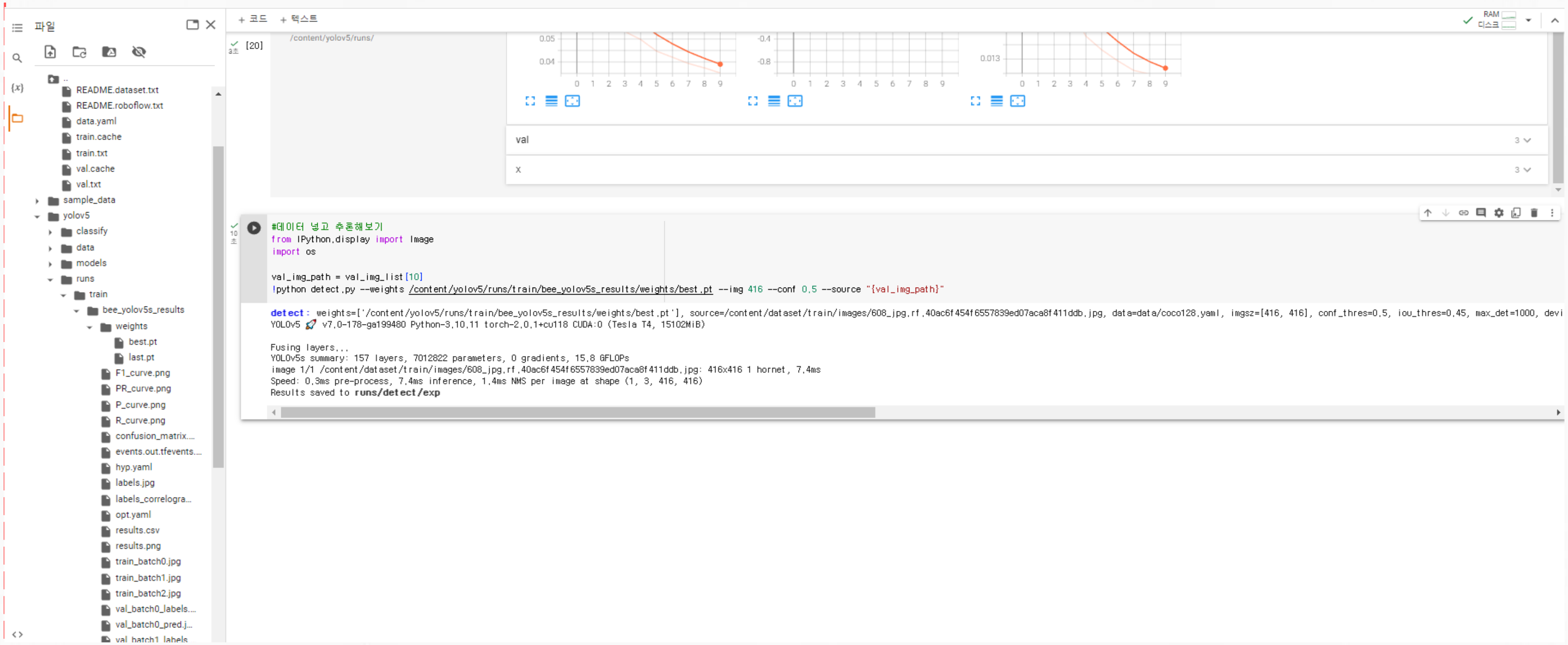
Horizontal Axis: STEP

Runs: ☑ train/bee_yolov5s_results

/content/yolov5/runs/

텐서플로우로 결과값 확인

# 8. 이미지 삽입 후 테스트

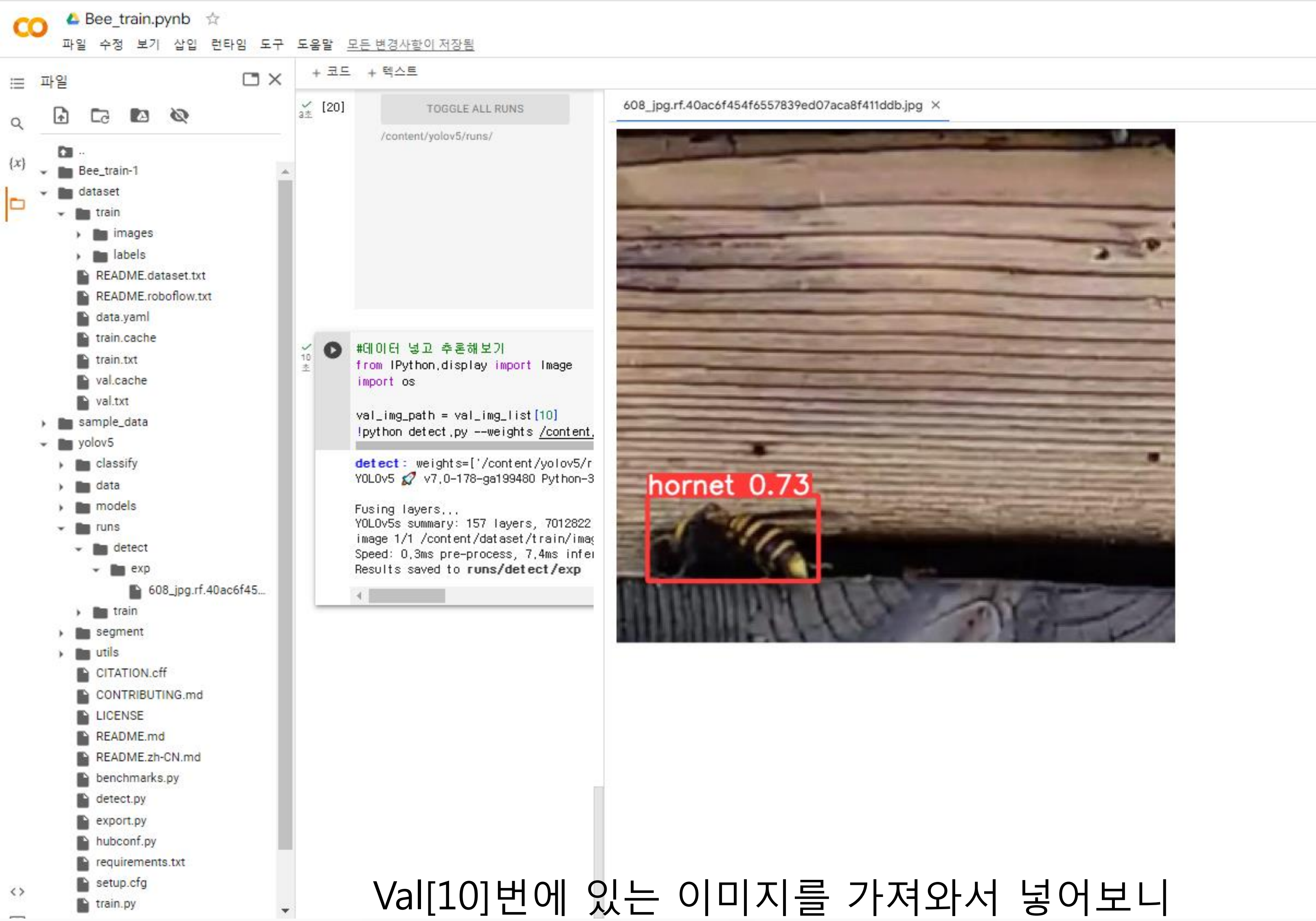


```
#데이터 넣고 추론해보기
from IPython.display import Image
import os

val_img_path = val_img_list[10]
!python detect.py --weights /content/yolov5/runs/train/bee_yolov5s_results/weights/best.pt --img 416 --conf 0.5 --source "{val_img_path}"
```

```
detect: weights=['/content/yolov5/runs/train/bee_yolov5s_results/weights/best.pt'], source=/content/dataset/train/images/608_jpg.rf.40ac6f454f6557839ed07aca8f411ddb.jpg, data=data/coco128.yaml, imgsz=[416, 416], conf_thres=0.5, iou_thres=0.45, max_det=1000, devi
YOLOv5 🚀 v7.0-178-ga199480 Python-3.10.11 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)

Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
image 1/1 /content/dataset/train/images/608_jpg.rf.40ac6f454f6557839ed07aca8f411ddb.jpg: 416x416 1 hornet, 7.4ms
Speed: 0.3ms pre-process, 7.4ms inference, 1.4ms NMS per image at shape (1, 3, 416, 416)
Results saved to runs/detect/exp
```

데이터 넣고 추론하기
runs/detect/exp로 저장됨

# 8. 이미지 삽입 후 테스트

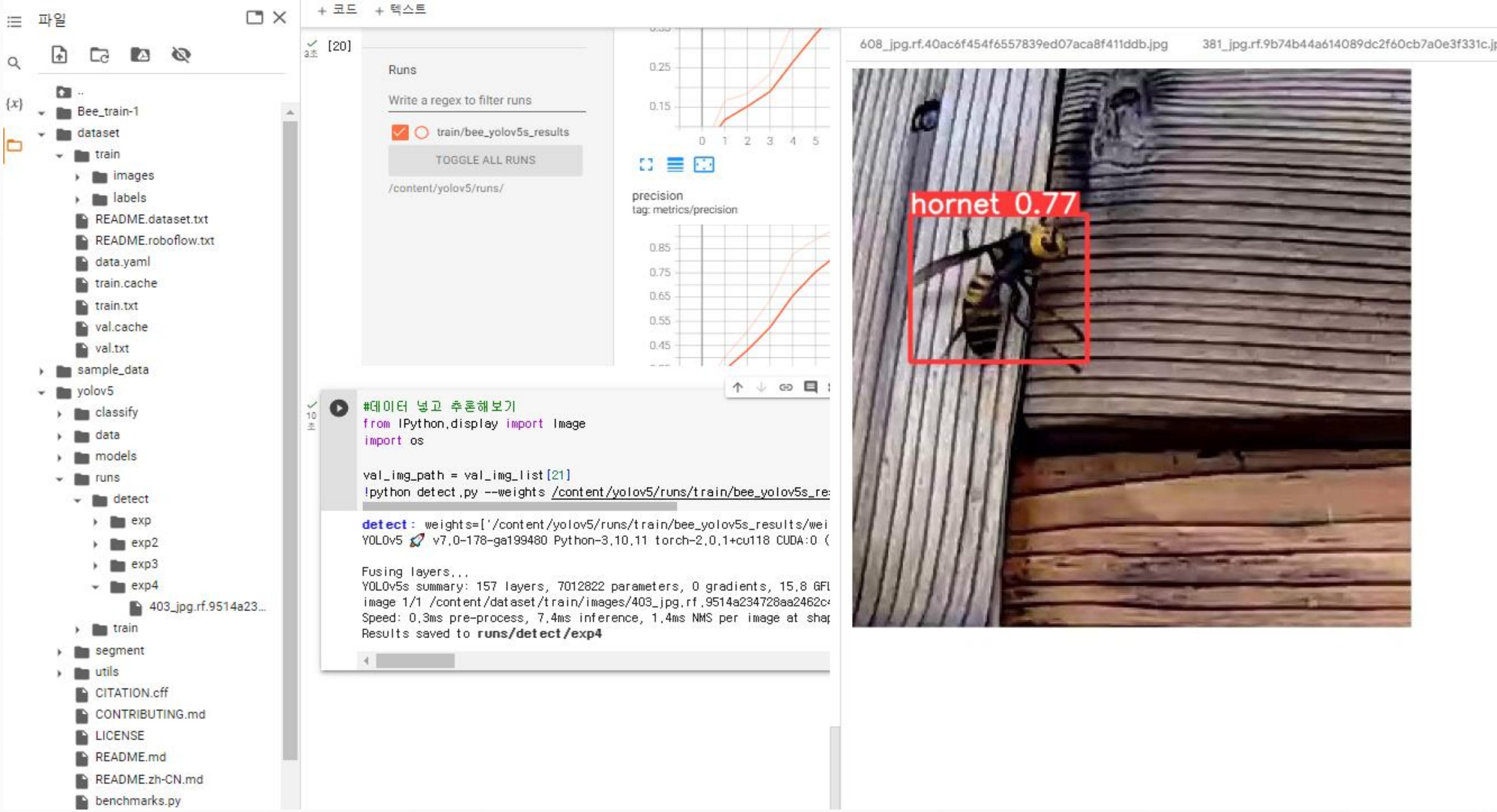


```
#데이터 넣고 추론해보기
from IPython.display import Image
import os

val_img_path = val_img_list[10]
!python detect.py --weights /content,
```

```
detect: weights=['/content/yolov5/r
YOLOv5 🚀 v7.0-178-ga199480 Python-3

Fusing layers...
YOLOv5s summary: 157 layers, 7012822
image 1/1 /content/dataset/train/imag
Speed: 0.3ms pre-process, 7.4ms infe
Results saved to runs/detect/exp
```

Val[10]번에 있는 이미지를 가져와서 넣어보니
잘 잡아내는 모습을 확인

# 8. 이미지 삽입 후 테스트



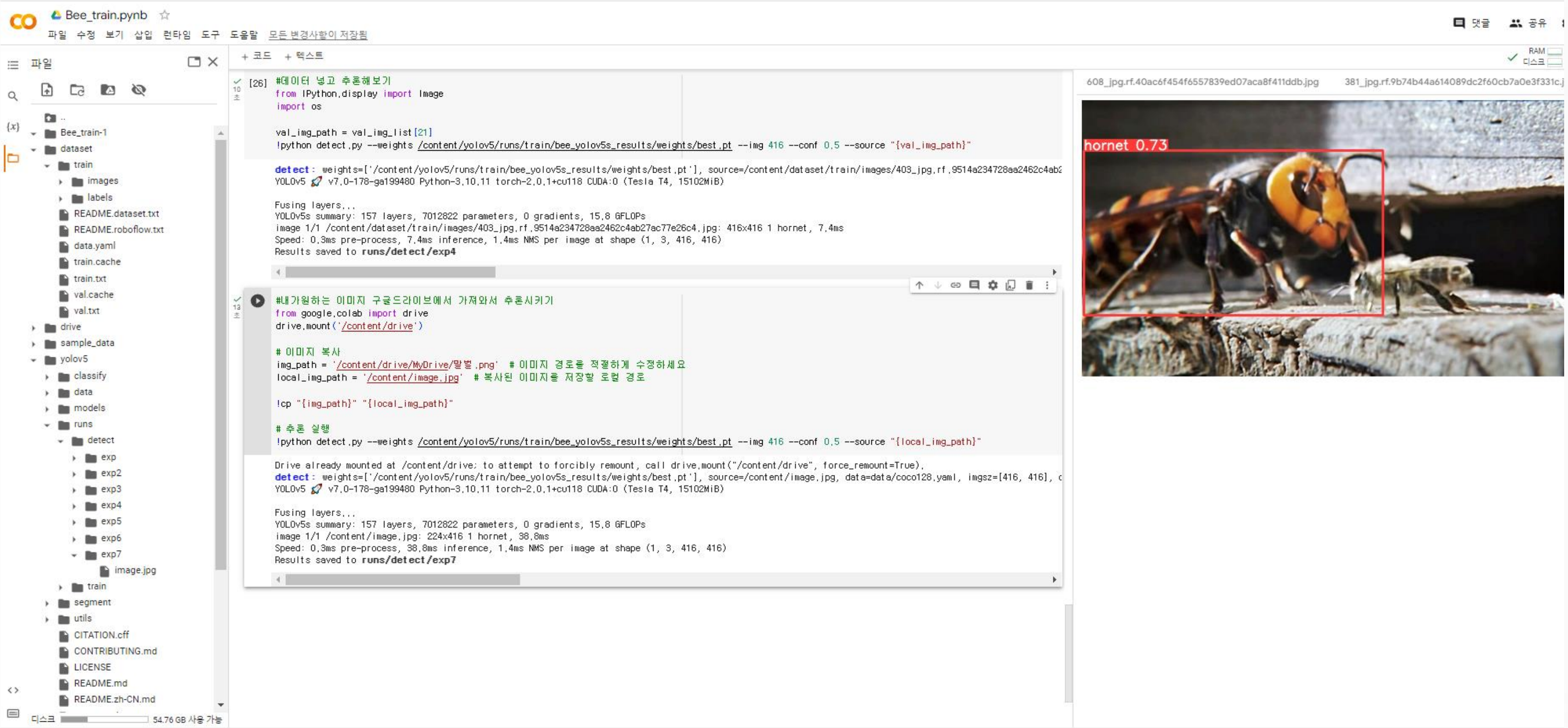
```
#데이터 넣고 추론해보기
from IPython.display import Image
import os

val_img_path = val_img_list[21]
!python detect.py --weights /content/yolov5/runs/train/bee_yolov5s_re
```

```
detect: weights=['/content/yolov5/runs/train/bee_yolov5s_results/wei
YOLOv5 🚀 v7.0-178-ga199480 Python-3.10.11 torch-2.0.1+cu118 CUDA:0 (

Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFL
image 1/1 /content/dataset/train/images/403_jpg.rf.9514a234728aa2462c
Speed: 0.3ms pre-process, 7.4ms inference, 1.4ms NMS per image at shap
Results saved to runs/detect/exp4
```

Val[21]번에 있는 이미지를 가져와서 넣어보니
잘 잡아내는 모습을 확인

# 9. 원하는 이미지 업로드 후 테스트

# 9. 원하는 이미지 업로드 후 테스트



```
[26] #데이터 넣고 추론해보기
from IPython.display import Image
import os

val_img_path = val_img_list[21]
!python detect.py --weights /content/yolov5/runs/train/bee_yolov5s_results/weights/best.pt --img 416 --conf 0.5 --source "{val_img_path}"
```

```
detect: weights=['/content/yolov5/runs/train/bee_yolov5s_results/weights/best.pt'], source=/content/dataset/train/images/403_jpg.rf.9514a234728aa2462c4ab2
YOLOv5 🚀 v7.0-178-ga199480 Python-3.10.11 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)

Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
image 1/1 /content/dataset/train/images/403_jpg.rf.9514a234728aa2462c4ab27ac77e26c4.jpg: 416x416 1 hornet, 7.4ms
Speed: 0.3ms pre-process, 7.4ms inference, 1.4ms NMS per image at shape (1, 3, 416, 416)
Results saved to runs/detect/exp4
```

```
#내가원하는 이미지 구글드라이브에서 가져와서 추론시키기
from google.colab import drive
drive.mount('/content/drive')

# 이미지 복사
img_path = '/content/drive/MyDrive/말벌.png'  # 이미지 경로를 적절하게 수정하세요
local_img_path = '/content/image.jpg'  # 복사된 이미지를 저장할 로컬 경로

!cp "{img_path}" "{local_img_path}"

# 추론 실행
!python detect.py --weights /content/yolov5/runs/train/bee_yolov5s_results/weights/best.pt --img 416 --conf 0.5 --source "{local_img_path}"
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
detect: weights=['/content/yolov5/runs/train/bee_yolov5s_results/weights/best.pt'], source=/content/image.jpg, data=data/coco128.yaml, imgsz=[416, 416], c
YOLOv5 🚀 v7.0-178-ga199480 Python-3.10.11 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)

Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
image 1/1 /content/image.jpg: 224x416 1 hornet, 38.8ms
Speed: 0.3ms pre-process, 38.8ms inference, 1.4ms NMS per image at shape (1, 3, 416, 416)
Results saved to runs/detect/exp7
```

구글드라이브에 업로드한 나의 이미지로 추론하기
정상적 인식 확인

# 감사합니다