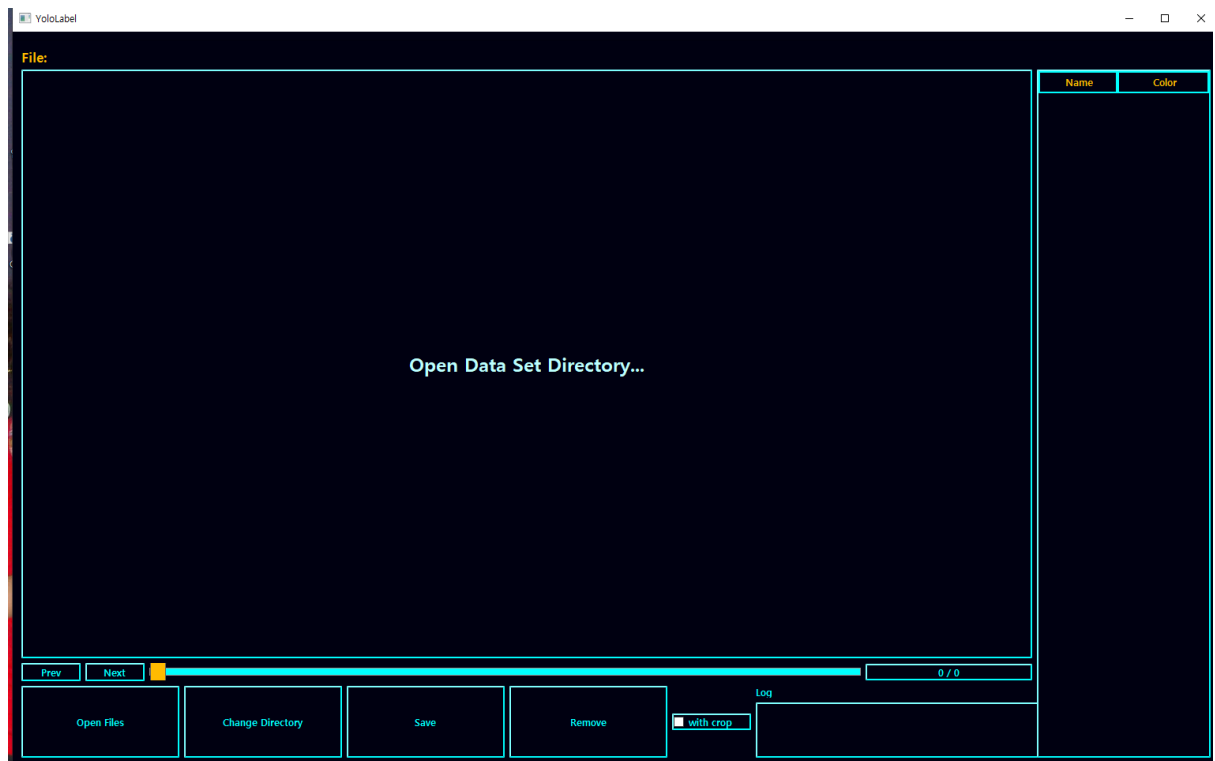
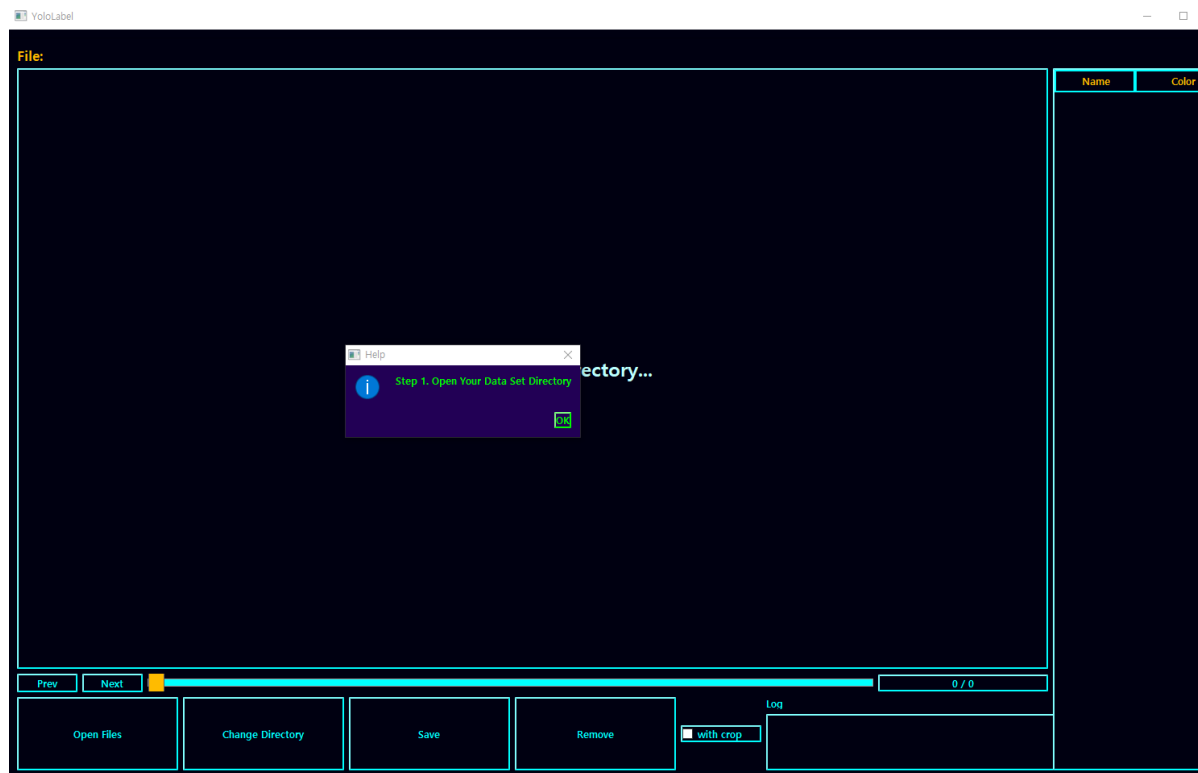


d:\LectureAll\Raspberrypi==\YoloLabel\images*.*					*
Name	Ext	Size	Date	Attr	
[..]		<DIR>	2022-11-23 14:56	----	
im001	PNG	366,369	2022-11-23 14:41	-a--	
im002	PNG	536,495	2022-11-23 14:42	-a--	
im003	PNG	352,048	2022-11-23 14:42	-a--	
im004	PNG	243,984	2022-11-23 14:42	-a--	
im005	PNG	582,123	2022-11-23 14:43	-a--	
im006	PNG	264,185	2022-11-23 14:43	-a--	
im007	PNG	277,817	2022-11-23 14:45	-a--	
im008	PNG	254,627	2022-11-23 14:45	-a--	
im009	PNG	364,502	2022-11-23 14:46	-a--	
im010	PNG	374,049	2022-11-23 14:46	-a--	
im011	PNG	580,891	2022-11-23 14:46	-a--	
im012	PNG	425,060	2022-11-23 14:47	-a--	
im013	PNG	439,938	2022-11-23 14:47	-a--	
im014	PNG	463,779	2022-11-23 14:47	-a--	
im015	PNG	340,061	2022-11-23 14:47	-a--	
im016	PNG	175,183	2022-11-23 14:48	-a--	
im017	PNG	260,538	2022-11-23 14:48	-a--	
im018	PNG	602,578	2022-11-23 14:48	-a--	
im019	PNG	428,477	2022-11-23 14:49	-a--	
im020	PNG	350,483	2022-11-23 14:49	-a--	
name	txt	8	2022-11-23 14:56	-a--	

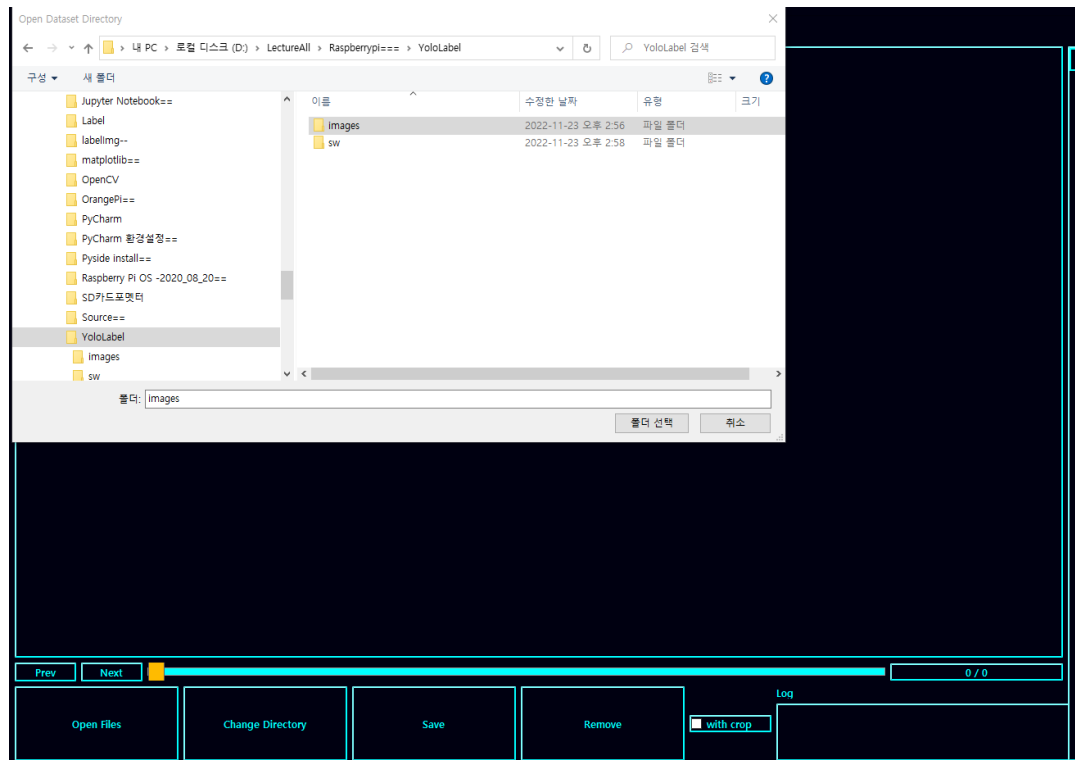
3) Labelling (예: YoloLabel은 적당한 폴더에 복사한 후에 YoloLabel.exe 실행)

. 하단에서 'Open Files' 클릭한 후 'Step 1. Open Your Data Set Directory'에서 'OK'를 클릭





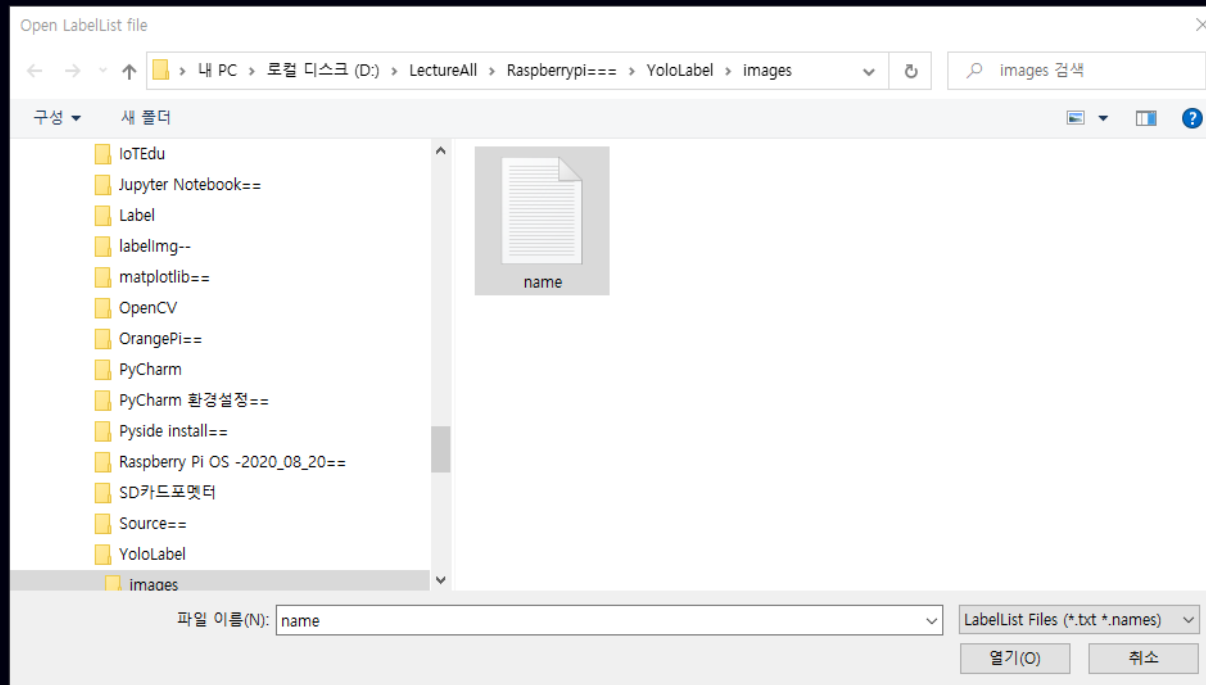
이미지가 저장된 '폴더 선택'



. 'Step 2. Open Your Label List File'에서 OK를 누르고 name.txt가 있는 폴더



File:



Prev Next 0 / 0

Open Files	Change Directory	Save	Remove	<input type="checkbox"/> with crop	Log
------------	------------------	------	--------	------------------------------------	-----

. 첫번째 사진부터 라벨링 작업 시작. 우측에서 해당되는 객체를 선택하고 라벨링(사각형)하고 마우스 좌측버튼 클릭. 라벨링 취소는 라벨선택한 후 마우스 우측 버튼 클릭. 라벨링후 하단에 'Save' 누르고, 하단에 'Next'를 클릭하여 다음 사진으로 진행해서 반복. 라벨링은 객체 전체가 포함되게 선택. 불필요한 사진 제거는 하단의 'Remove' 클릭

<Dataset 만들기(Roboflow 활용)>

- 1) Roboflow 사이트 로그인(<https://roboflow.com/>). 계정이 없으면 'Sign up'
 - . 상단 메뉴에서 'Projects' 를 선택하고, 우측에서 '+ Create New Project'를 클릭
 - . Project Name 등 설정한 후 (영어로 이름), 'Create Public Project' 클릭

 roboflow

Let's create your project.

cku >  New Public Project

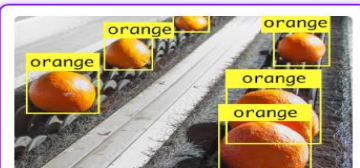
Project Name

Dog and Cats Classification

Annotation Group 

dogs-cats


Project Type



Object Detection

Identify objects and their positions with bounding boxes.

Best For

Counting 



Classification

Assign labels to the entire image.

Classification Type

☒ Multi-Label ☐ Single-Label

Best For


 Filtering  Content Moderation



Instance Segmentation

Detect multiple objects

Best For

 Measurements

Show More 

Cancel

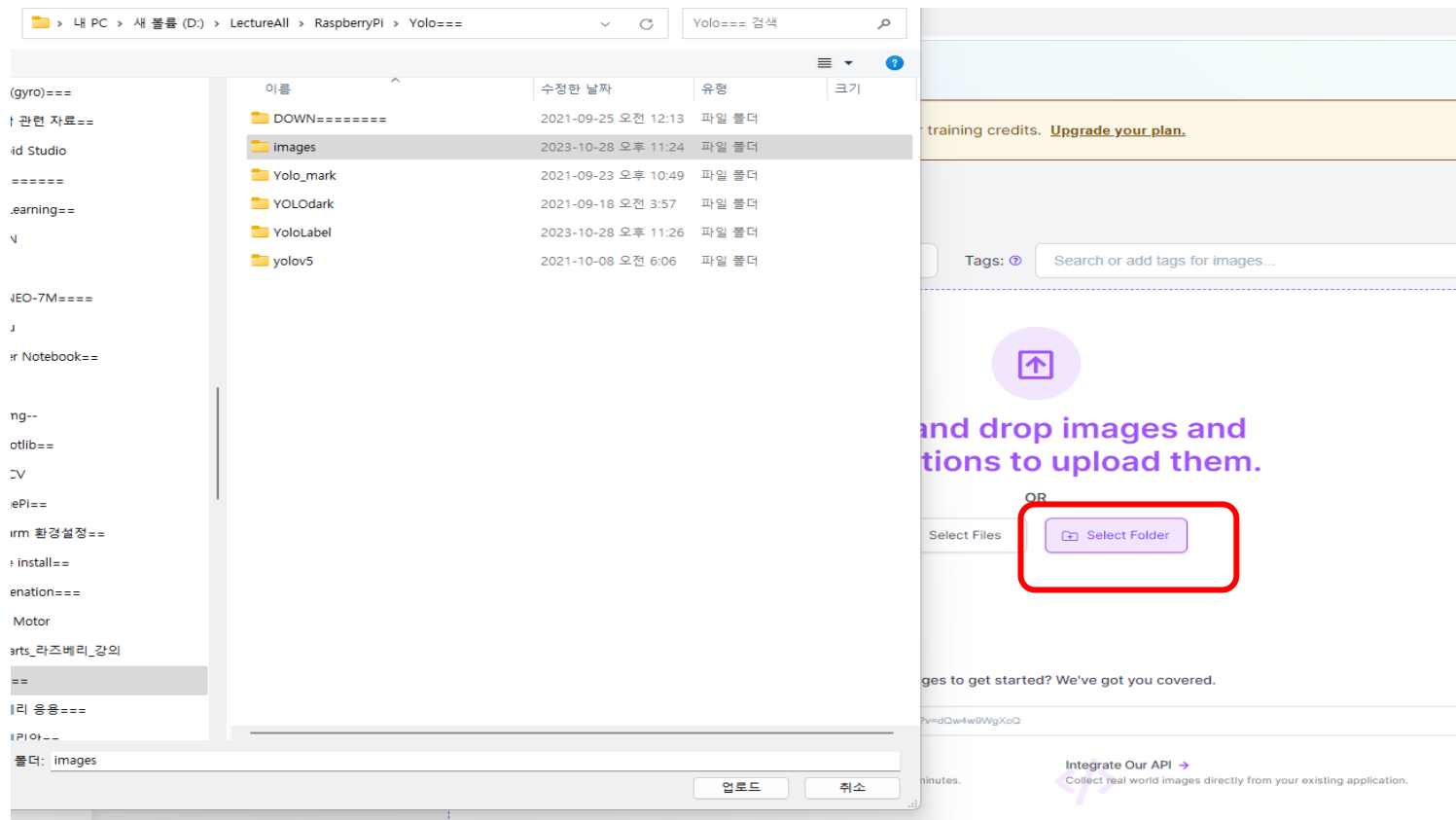
Create Public Project

, Project Name: Dog and Cat classification (이름은 영어로 임의로 지정)

. Annotation Group: dogs-cats (영어로 임의로 지정)

. 이상을 설정 후 우측 하단의 'Create Public Project'를 클릭

2) Upload화면 중앙에 있는 'Select Folder'를 클릭하여 image 폴더 지정한 후 '업로드'를 클릭하여 upload. 단 경로는 **영어**로 폴더명이 정해져야 함.



. 이때 다시 물어 보면 '업로드'를 클릭

Forum

This

파일 41개를 이 사이트에 업로드하시겠습니까?
'images'의 모든 파일이 업로드됩니다. 사이트를 신뢰할 수 있을 때만
실행하세요.

업로드취소

Upload [Want to change the classes on your annotated images?](#)

Batch Name: Tags:

↑

Drag and drop images and
annotations to upload them.

OR

Select Files

Select Folder

Need images to get started? We've got you covered.

Import YouTube Video:

→

Find a Universe Dataset →

Browse over 100k free datasets for images and build a model in minutes.

Integrate Our API →

Collect real world images directly from your existing application.

파일 41개를 이 사이트에 업로드하시겠습니까?

'images'의 모든 파일이 업로드됩니다. 사이트를 신뢰할 수 있을 때만
실행하세요.

업로드

취소

Upload

[? Want to modify or change classes on your uploaded images?](#)


Batch Name Uploaded on 11/23/22 at 3:28 pm

All Images **0** Annotated **0** Not Annotated **0**



Drag and drop
images and annotations

 Select Files

 Select Folder



Images
jpg, png, bmp



Annotations
in 26 formats >>



Video
mov, mp4, avi

This workspace reached its quota for training credits. [Upgrade your plan.](#)

Upload [Want to change the classes on your annotated images?](#)

Save and Continue

Batch Name: Folder: images

Tags: Search or add tags for images...

All Images 20

Annotated 20

Not Annotated 0

Drag and drop images and annotations.

Select Files

Select Folder

in .jpg, .png, .bmp, .webp in 26 formats in .mov, .mp4, .avi



im020.PNG



im019.PNG



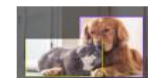
im018.PNG



im017.PNG



im016.PNG



im015.PNG



im014.PNG



im013.PNG



im012.PNG



im011.PNG



im010.PNG



im009.PNG



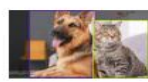
im008.PNG



im007.PNG



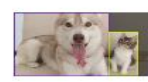
im006.PNG



im005.PNG



im004.PNG



im003.PNG



im002.PNG




im001.PNG

. 20개 image가 'Annotated'(라벨링)된 것을 보여줌. 혹시 'Not Annotated'된 것이 있으면 이 사이트에 다시 라벨링.

3) 'Save and Continue' 클릭




This workspace reached its quota for training credits. [Upgrade your plan.](#)


 Upload [? Want to change the classes on your annotated images?](#)


Batch Name: Tags: [?](#)


All Images **20** Annotated 20 Not Annotated 0


Drag and drop images and annotations.


 in .jpg, .png, .bmp  in 26 formats »  in .mov, .mp4, .avi



im020.PNG



im019.PNG



im018.PNG



im017.PNG



im016.PNG



im015.PNG



im014.PNG



im013.PNG



im012.PNG


im011.PNG


im010.PNG


im009.PNG


im008.PNG


im007.PNG

4) Split화면에서는 Train을 100%로 한 후 (슬라이드를 움직여서) 'Continue' 클릭.

How should we split these images? ×

Choose one [? What's Train, Valid, Test?](#)

Split Images Between Train/Valid/Test

Train 100% Valid 0% Test 0%

Not sure what this is? [Learn more on our blog.](#)

Cancel Continue

5) Dog and Cat classification Dataset에서 중앙에 있는 '+ Generate New Version'을 클릭 (안 나타나면 Skip)

Dog Cat classification Dataset

 Generate New Version

VERSIONS

To train a model, you must first generate a new version of your dataset.

Choose your dataset settings to get started.

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: 20

Classes: 2

Unannotated: 0



Train/Test Split

Training Set: 20 images

Validation Set: images

Testing Set: images



Preprocessing

 What can preprocessing do?

Decrease training time and increase performance by applying image transformations to all images in this dataset.

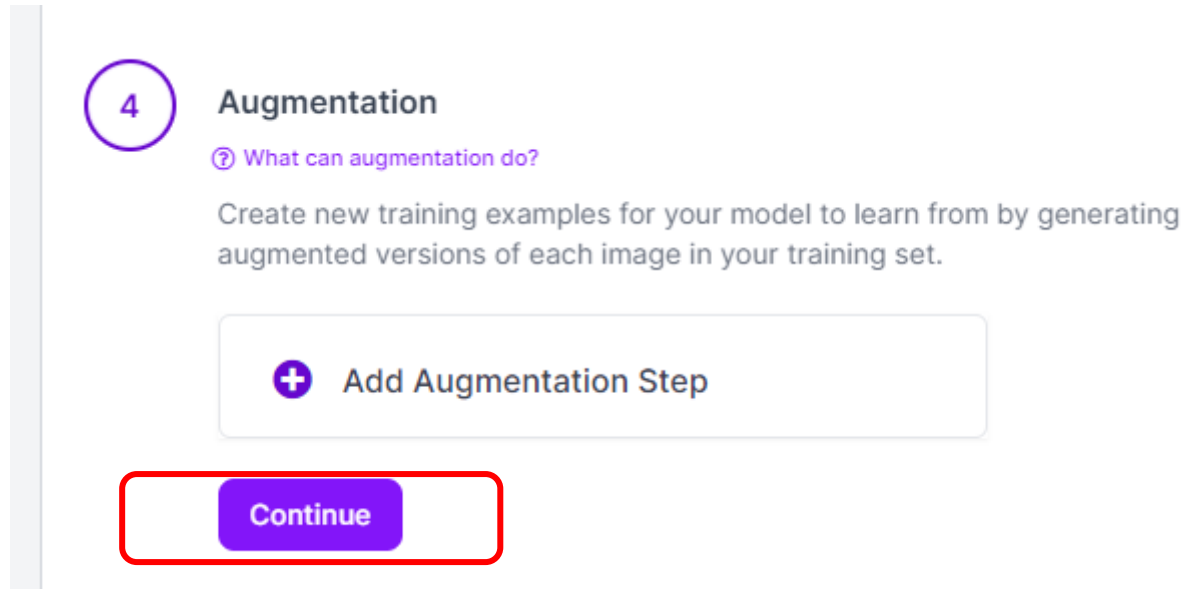
Auto-Orient

Edit

x

6) Preprocessing에서 'Continue' 클릭

7) Augmentation에서 'Continue' 클릭



8) Create에서 'Create'를 클릭

9) Dog and Cats classification Image Dataset에서 'Export Dataset' 클릭

Dog Cat classification Image Dataset

[+ Generate New Version](#)

VERSIONS

2023-10-29 12:21am
v1 Oct 29, 2023

v1 **2023-10-29 12:21am**

Generated on Oct 29, 2023

[Export Dataset](#)

This version doesn't have a model.

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

[Get More Credits](#)

[Custom Train and Upload](#)

Available Credits: 0

20 Total Images

[View All Images →](#)



Dataset Split

TRAIN SET
20 Images

100%

VALID SET
0 Images

%

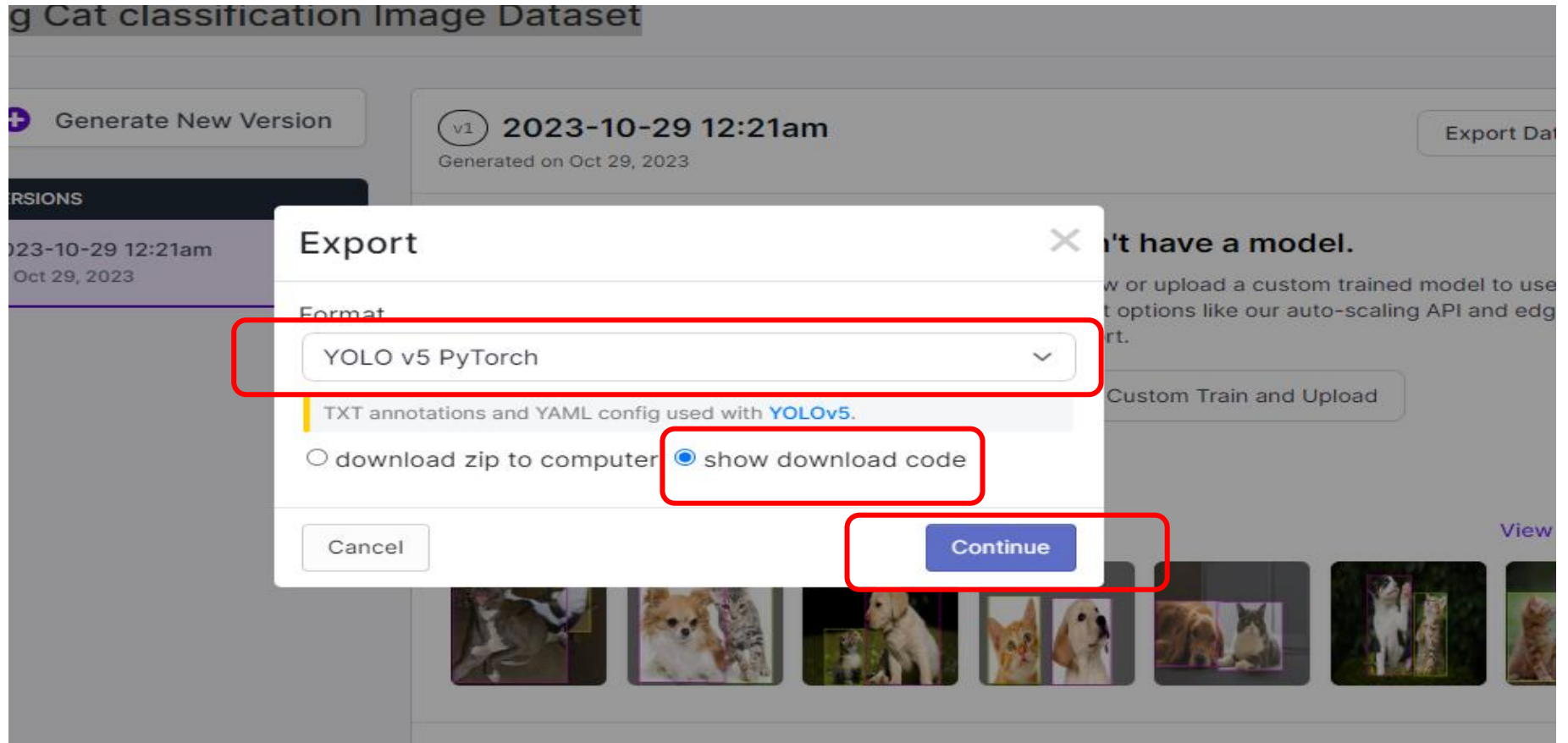
TEST SET
0 Images

%

Preprocessing


Auto-Orient: Applied

- 10) Format은 'YOLO v5 PyTorch', 'show download code'를 선택하고 하단에 있는 'Continue' 버튼 클릭



g and Cats Classification Image Dataset

 Create New Version

 **2024-04-11 5:21am**

Generated on Apr 11, 2024

VERSIONS

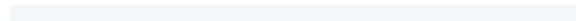
2024-04-11 5:21am

Apr 11, 2024

Export



Zippping Files...



Cancel

 Continue

Get More Credits

Available Credits: 0

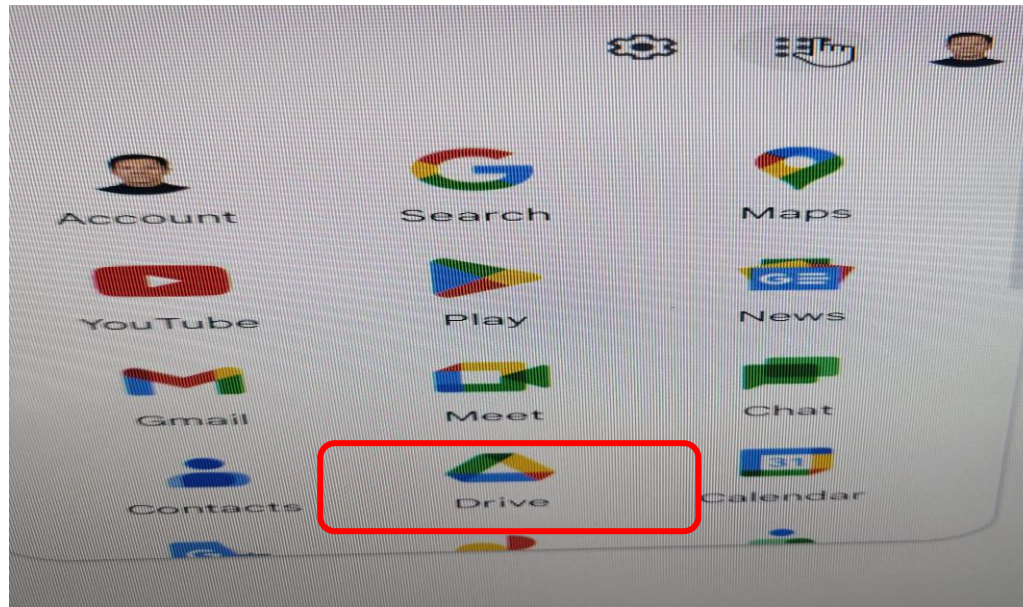
20 Total Images


```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="TBRLoVWy0oCer6eiqK0g")
project = rf.workspace("cku").project("dog-and-cats-classification")
version = project.version(1)
dataset = version.download("yolov5")
```

<Training (Colab)>

- 1) 구글 계정(없으면 가입)에서 사진 좌측의 점 9개를 클릭하면 보이는 Drive로 진입



2) My Drive/New Folder/More/Google Colaboratory 선택 (안 보이면 '+ Connect more apps'를 클릭해서 찾음)



+ New

Home

My Drive

› .ipynb_checkpoints

› Colab Notebooks

› detect_hands

› detect_hands1

› egohands

› images

› models

› notebooks

› ssd_mobilenet_v2_f...

› test

› Computers

Shared with me

Search in Drive



My Drive ▾

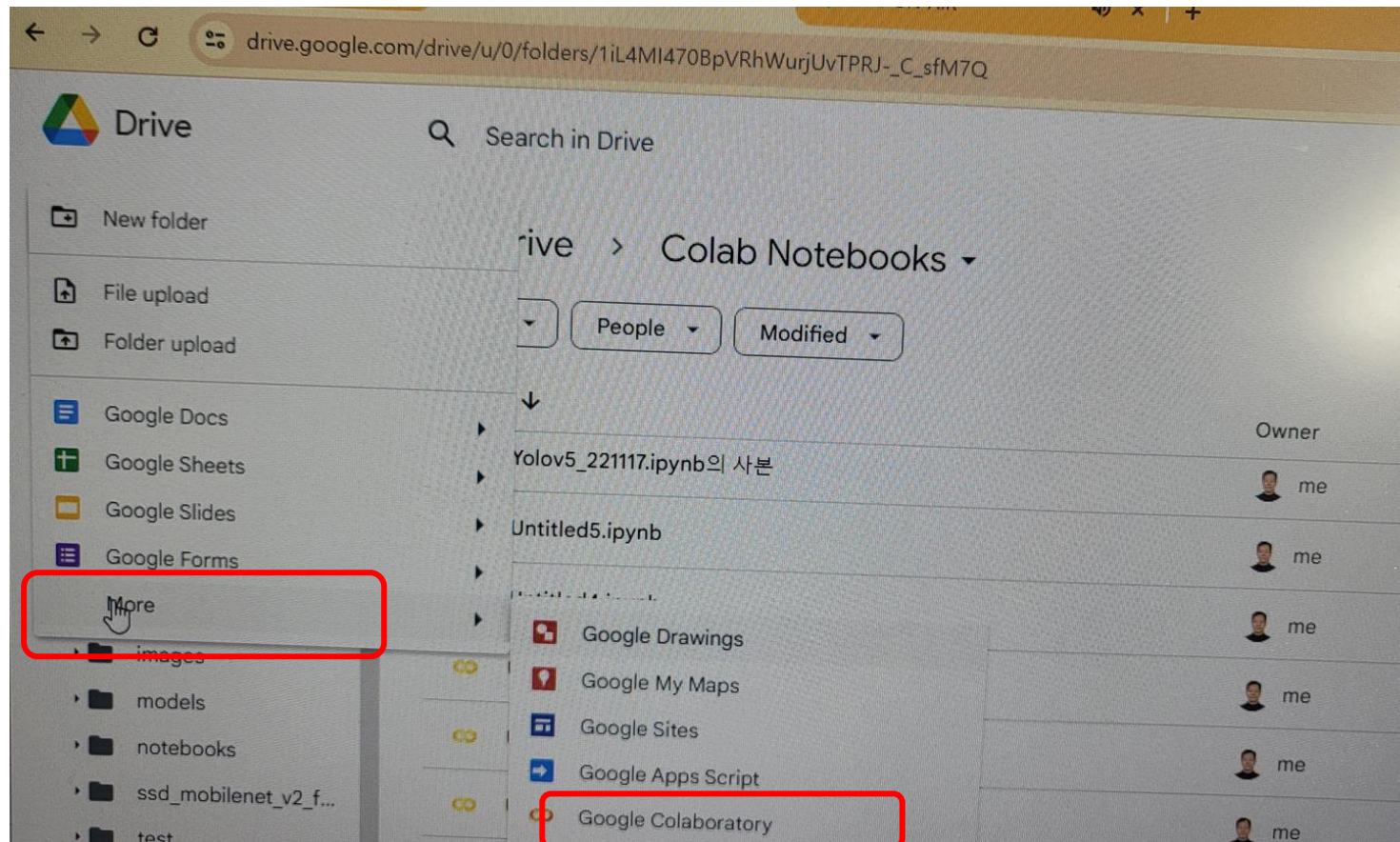
Type ▾

People ▾

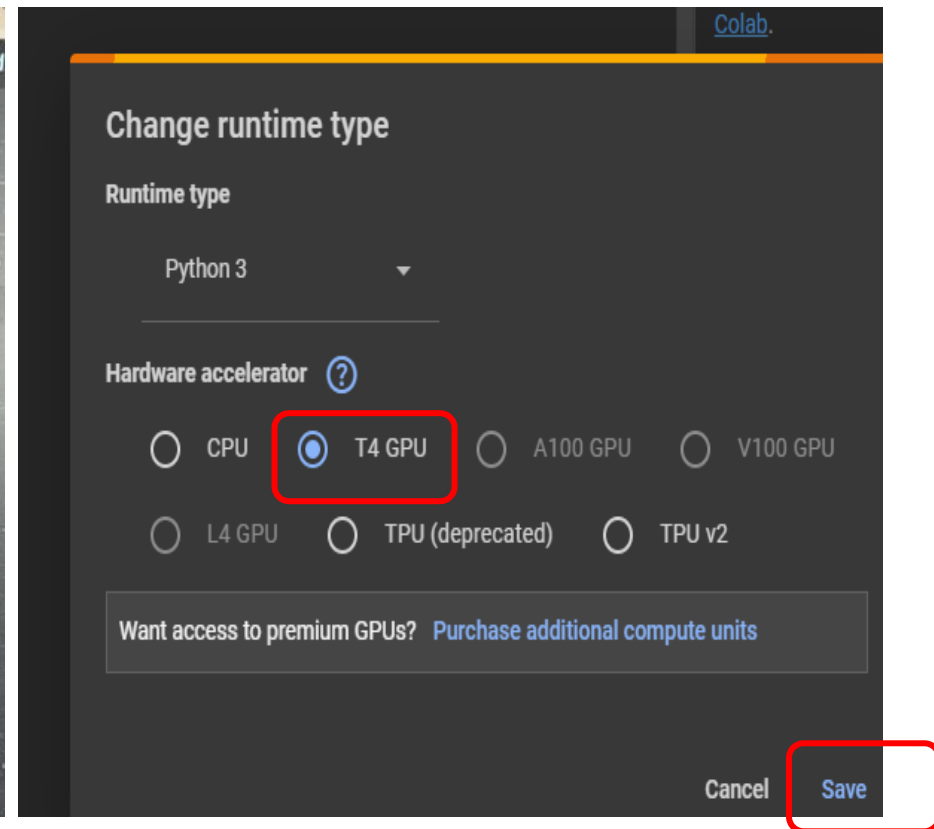
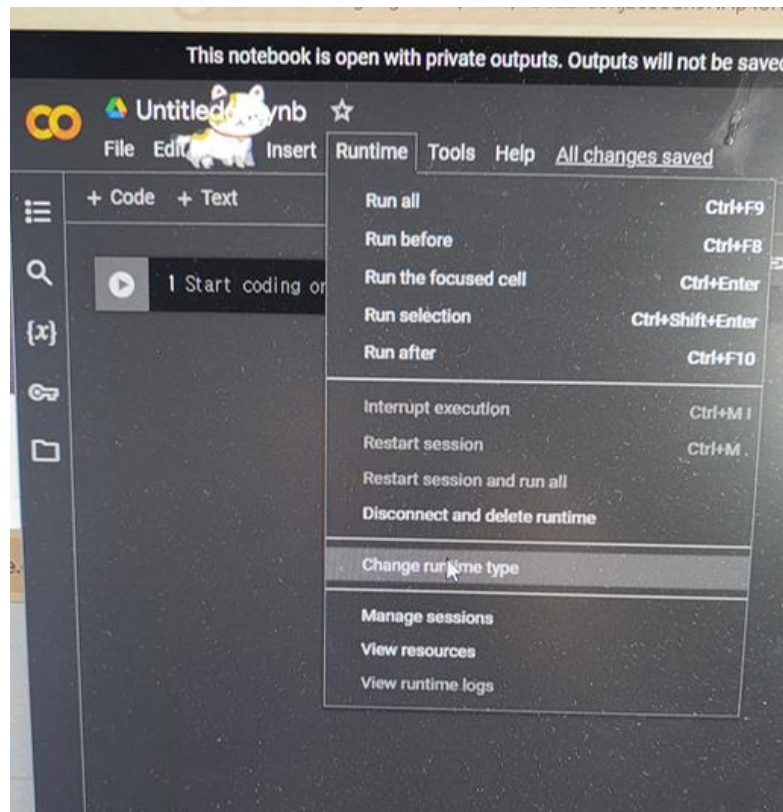
Modified ▾

Change your start page? Currently it's set to Home. You can change it anytime in Settings.

Name	Owner	Location
test	me	Ne
ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8	me	Ju
notebooks	me	O
models	me	Ne
images	me	Ne
egohands	me	Ne
detect_hands1	me	Ne



3) 아래의 Google Colab 상단 메뉴에서 'Runtime'(런타임)을 클릭하고 'Change runtime type'(런타임 유형변경)을 선택한 후, 'T4 GPU' 선택한 후 'Save'(저장) 클릭



3) 데이터셋 load (첫번째 셀에 붙여 넣기(^V) 하고, 셀좌측의 우측세모를 클릭해서 실행

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="TBRLoVWy0oCer6eiqK0g")
project = rf.workspace("cku").project("dog-and-cats-classification")
version = project.version(1)
dataset = version.download("yolov5")
```

✓ 18 초

```
1 !pip install roboflow
2
3 from roboflow import Roboflow
4 rf = Roboflow(api_key="TBRL0vWly0oCer6eiqK0g")
5 project = rf.workspace("cku").project("dog-cat-classification-q6hav")
6 dataset = project.version(1).download("yolov5")
7
```

Collecting roboflow
Downloading roboflow-1.1.7-py3-none-any.whl (58 kB)
58.8/58.8 kB 1.6 MB/s eta 0:00:00

Collecting certifi==2022.12.7 (from roboflow)
Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
155.3/155.3 kB 6.6 MB/s eta 0:00:00

Collecting chardet==4.0.0 (from roboflow)
Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
178.7/178.7 kB 21.9 MB/s eta 0:00:00

Collecting cycler==0.10.0 (from roboflow)
Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)

Collecting idna==2.10 (from roboflow)
Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
58.8/58.8 kB 7.6 MB/s eta 0:00:00

(중략)

```
Successfully uninstalled certifi-2023.7.22
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following errors:
lida 0.0.10 requires fastapi, which is not installed.
lida 0.0.10 requires kaleido, which is not installed.
lida 0.0.10 requires python-multipart, which is not installed.
lida 0.0.10 requires uvicorn, which is not installed.
Successfully installed certifi-2022.12.7 chardet-4.0.0 cycler-0.10.0 idna-2.10 opencv-python-headless-4.8.0.74 pyparsing-2.4.7 python-dotenv-0.20.0
WARNING: The following packages were previously imported in this runtime:
[certifi,cycler,pyparsing]
You must restart the runtime in order to use newly installed versions.

RESTART RUNTIME

loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Dog-Cat-classification--1 to yolov5pytorch:: 100%|██████████| 761/761 [00:00<00:00, 1132.88it/s]

Extracting Dataset Version Zip to Dog-Cat-classification--1 in yolov5pytorch:: 100%|██████████| 46/46 [00:00<00:00, 5647.24it/s]
```

5) YOLOv5 다운 및 학습 (상단에서 '+Code'를 클릭해서 새 cell 만들고 진행)

```
%cd /content
```

```
!git clone https://github.com/ultralytics/yolov5
```

```
3초 1 %cd /content
2 !git clone https://github.com/ultralytics/yolov5

/content
Cloning into 'yolov5'...
remote: Enumerating objects: 16026, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 16026 (delta 33), reused 41 (delta 25), pack-reused 15967
Receiving objects: 100% (16026/16026), 14.68 MiB | 17.40 MiB/s, done.
Resolving deltas: 100% (10999/10999), done.
```

%cd /content/yolov5/

!pip install -r requirements.txt

```
6초 1 %cd /content/yolov5/
2 !pip install -r requirements.txt

/content/yolov5
Collecting gitpython>=3.1.30 (from -r requirements.txt (line 5))
  Downloading GitPython-3.1.40-py3-none-any.whl (190 kB)
  190.6/190.6 kB 3.7 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)) (3.7.1)
Requirement already satisfied: numpy>=1.22.2 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 7)) (1.23.5)
Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 8)) (4.8.0.76)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 9)) (9.4.0)
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)) (6.0.1)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 12)) (2.31.0)
Requirement already satisfied: tqdm>=4.4.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 13)) (4.64.1)
```

(종료)

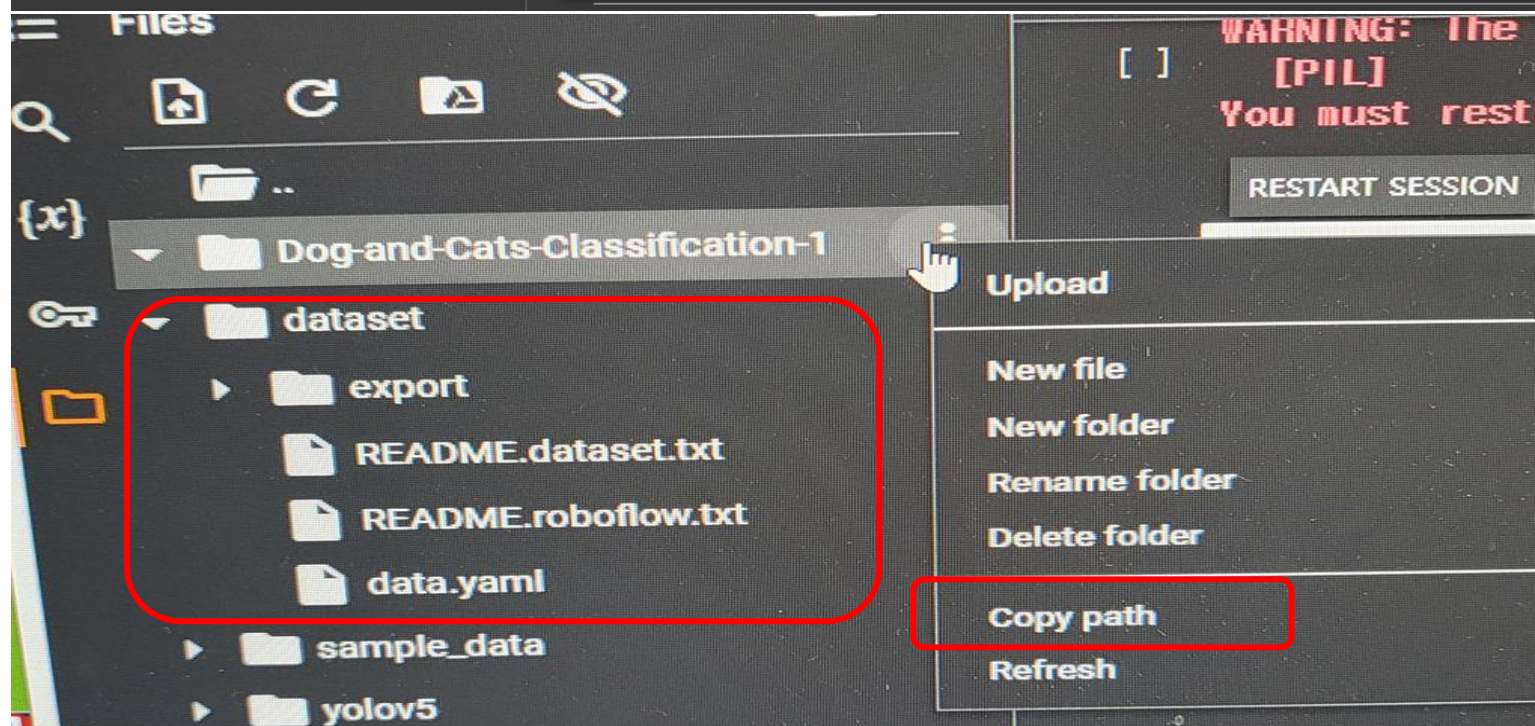
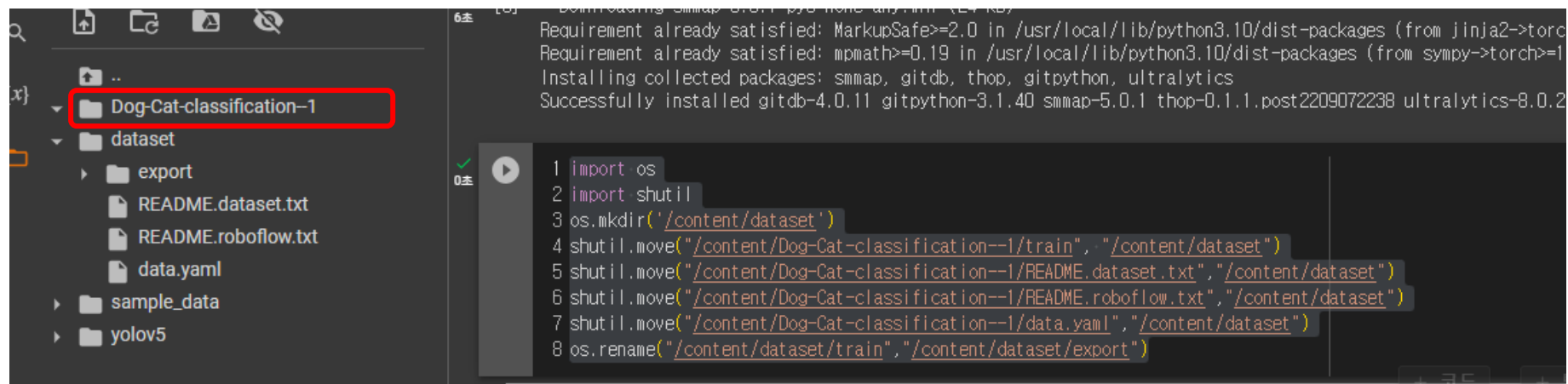
```
Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (2.1.0)
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.0.147->-r requirements.txt (line 18)) (9.0.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->-r requirements.txt (line 27)) (2023.3.post1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from cycler>=0.10->matplotlib>=3.3->-r requirements.txt (line 6)) (1.16.0)
Collecting smmap<5,>=3.0.1 (from gitdb<5,>=4.0.1->gitpython>=3.1.30->-r requirements.txt (line 5))
  Downloading smmap-5.0.1-py3-none-any.whl (24 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.8.0->-r requirements.txt (line 15)) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.8.0->-r requirements.txt (line 15)) (1.3.0)
Installing collected packages: smmap, gitdb, thop, gitpython, ultralytics
Successfully installed gitdb-4.0.11 gitpython-3.1.40 smmap-5.0.1 thop-0.1.1.post2209072238 ultralytics-8.0.202
```

. dataset 폴더 만들기 (/content/dataset. /content가 화면에 보이는 제일 높은 경로)

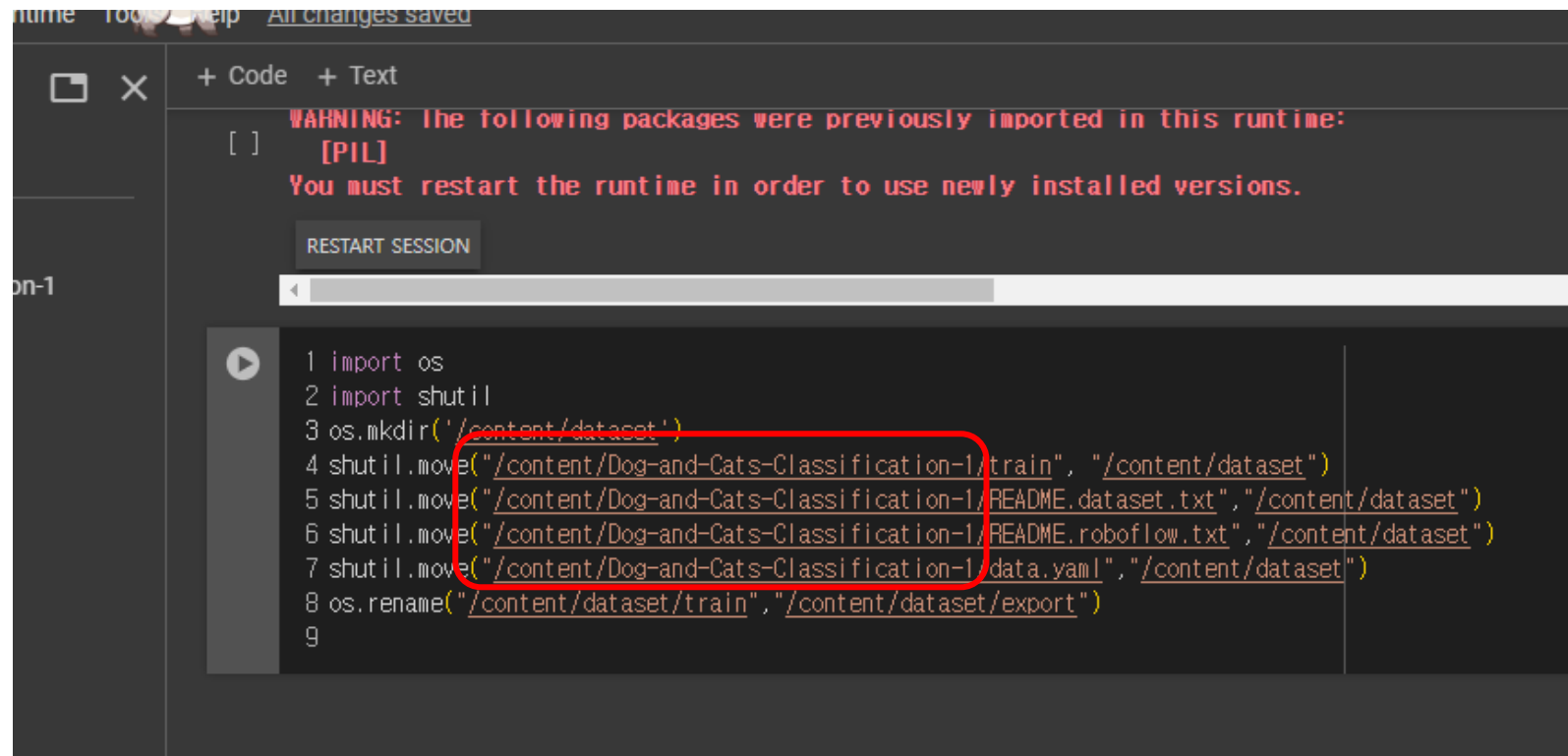
```
!pip install -r requirements.txt
```

위의 내용을 아래와 같이 고침. 단 경로는 'Dog-Cat-classification—1' 폴더 아래 4개 폴더 각각에 마우스 우측버튼을 눌러서 '경로복사'를 해 와서 위의 적색부분을 아래와 같이 수정

```
import os
import shutil
os.mkdir('/content/dataset')
shutil.move("/content/Dog-Cats-classification-1/train", "/content/dataset")
shutil.move("/content/Dog-Cats-classification-1/README.dataset.txt", "/content/dataset")
shutil.move("/content/Dog-Cats-classification-1/README.roboflow.txt", "/content/dataset")
shutil.move("/content/Dog-Cats-classification-1/data.yaml", "/content/dataset")
os.rename("/content/dataset/train", "/content/dataset/export")
```



. Data Set 이름(Dog-and-Cats-Classification-1)에서 우측 버튼을 눌러서 'Copy path' 한 후 이전 그림에서 path를 수정



```
WARNING: The following packages were previously imported in this runtime:
[PIL]
You must restart the runtime in order to use newly installed versions.
RESTART SESSION

1 import os
2 import shutil
3 os.mkdir('/content/dataset')
4 shutil.move("/content/Dog-and-Cats-Classification-1/train", "/content/dataset")
5 shutil.move("/content/Dog-and-Cats-Classification-1/README.dataset.txt", "/content/dataset")
6 shutil.move("/content/Dog-and-Cats-Classification-1/README.roboflow.txt", "/content/dataset")
7 shutil.move("/content/Dog-and-Cats-Classification-1/data.yaml", "/content/dataset")
8 os.rename("/content/dataset/train", "/content/dataset/export")
9
```

%cat /content/dataset/data.yaml

```
1 %cat /content/dataset/data.yaml

names:
  - cat
  - dog
nc: 2
roboflow:
  license: CC BY 4.0
  project: dog-cat-classification-q6hav
  url: https://universe.roboflow.com/cku/dog-cat-classification-q6hav/dataset/1
  version: 1
  workspace: cku
test: ../test/images
train: Dog-Cat-classification--1/train/images
val: Dog-Cat-classification--1/valid/images
```

%cd /

from glob import glob

img_list = glob('/content/dataset/export/images/*.jpg')

print(len(img_list))

```
1 %cd /
2 from glob import glob
3 img_list = glob('/content/dataset/export/images/*.jpg')
4 print(len(img_list))

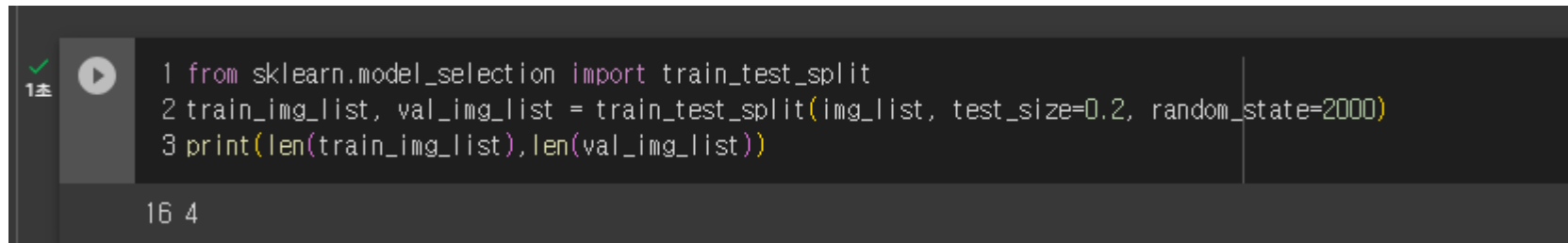
/
20
```



```
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)
```

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

A screenshot of a Jupyter Notebook cell. The code area contains three lines: `1 from sklearn.model_selection import train_test_split`, `2 train_img_list, val_img_list = train_test_split(img_list, test_size=0.2, random_state=2000)`, and `3 print(len(train_img_list),len(val_img_list))`. The output area shows the result `16 4`.

```
1 from sklearn.model_selection import train_test_split
2 train_img_list, val_img_list = train_test_split(img_list, test_size=0.2, random_state=2000)
3 print(len(train_img_list),len(val_img_list))

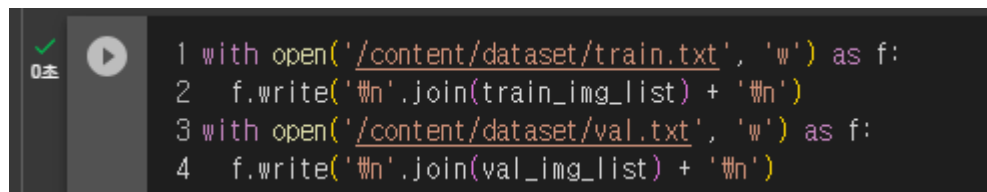
16 4
```

```
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')
```

A screenshot of a Jupyter Notebook cell. The code area contains four lines: `1 with open('/content/dataset/train.txt', 'w') as f:`, `2 f.write('\n'.join(train_img_list) + '\n')`, `3 with open('/content/dataset/val.txt', 'w') as f:`, and `4 f.write('\n'.join(val_img_list) + '\n')`.

```
1 with open('/content/dataset/train.txt', 'w') as f:
2     f.write('\n'.join(train_img_list) + '\n')
3 with open('/content/dataset/val.txt', 'w') as f:
4     f.write('\n'.join(val_img_list) + '\n')
```

```
import yaml
```

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

```
    data = yaml.full_load(f)
```

print(data)

```
1 import yaml
2 with open('/content/dataset/data.yaml', 'r') as f:
3     data = yaml.full_load(f)
4 print(data)

{'names': ['cat', 'dog'], 'nc': 2, 'roboflow': {'license': 'CC BY 4.0', 'project': 'dog-cat-classification-q6hav', 'url': 'https://universe.roboflow.com/cku/dog-cat-classification-q6hav/dataset/1', 'version': '1.0.0'}}
```

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)

```
1 data['train']='/content/dataset/train.txt'
2 data['val']='/content/dataset/val.txt'
3 with open('/content/dataset/data.yaml', 'w') as f:
4     yaml.dump(data, f)
```

print(data)

```
1 print(data)

{'names': ['cat', 'dog'], 'nc': 2, 'roboflow': {'license': 'CC BY 4.0', 'project': 'dog-cat-classification-q6hav', 'url': 'https://universe.roboflow.com/cku/dog-cat-classification-q6hav/dataset/1', 'version': '1.0.0'}}
```

%cd /content/yolov5/

!python train.py --img 416 --batch 16 --epochs 50 --data /content/dataset/data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name dog_cats_yolov5s_results

```
1 %cd /content/yolov5/
2 !python train.py --img 416 --batch 16 --epochs 50 --data /content/dataset/data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name dog_cats_yolov5s_results

... /content/yolov5
2023-10-29 11:31:46.759181: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been
2023-10-29 11:31:46.759243: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been
2023-10-29 11:31:46.759298: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been
train: weights=yolov5s.pt, cfg=./models/yolov5s.yaml, data=/content/dataset/data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=50, batch_size=16, imgsz=416, rect=False, resume=False, nosave=False,
github: up to date with https://github.com/ultralytics/yolov5
YOLOv5 v7.0-230-g53efd07 Python=3.10.12 torch=2.1.0+cu118 CUDA=0 (Tesla T4, 15102MiB)
```

(중략)

```
Epoch      GPU_mem  box_loss  obj_loss  cls_loss  Instances  Size
48/49      1.52G   0.08132   0.03729   0.02833     75      416: 100% 1/1 [00:00<00:00, 7.86it/s]
          Class    Images  Instances      P          R    mAP50  mAP50-95: 100% 1/1 [00:00<00:00, 12.46it/s]
          all         4          8   0.00667         1    0.24    0.0833

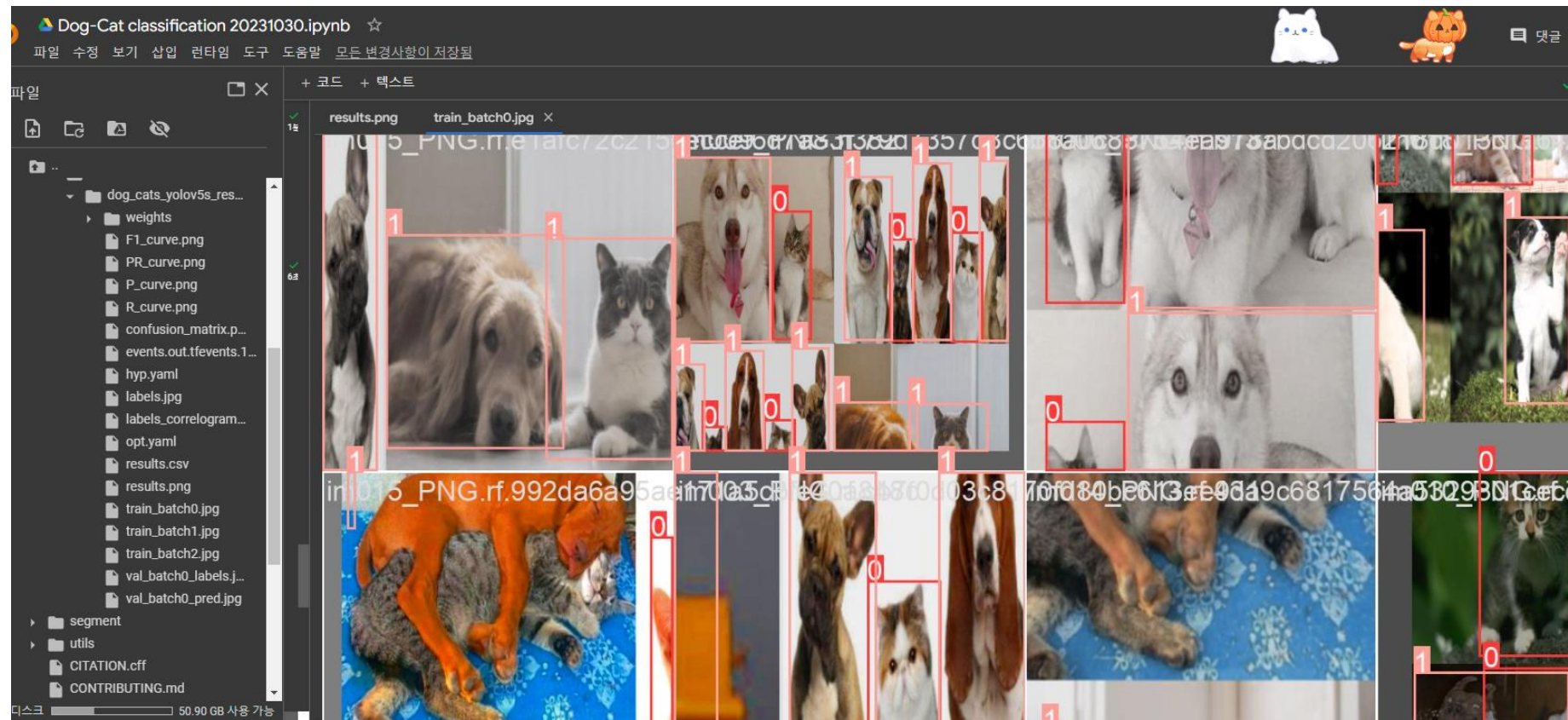
Epoch      GPU_mem  box_loss  obj_loss  cls_loss  Instances  Size
49/49      1.52G   0.07321   0.03047   0.02612     60      416: 100% 1/1 [00:00<00:00, 10.20it/s]
          Class    Images  Instances      P          R    mAP50  mAP50-95: 100% 1/1 [00:00<00:00, 7.45it/s]
          all         4          8   0.00667         1    0.24    0.0833

50 epochs completed in 0.018 hours.
Optimizer stripped from runs/train/dog_cats_yolov5s_results/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/dog_cats_yolov5s_results/weights/best.pt, 14.3MB

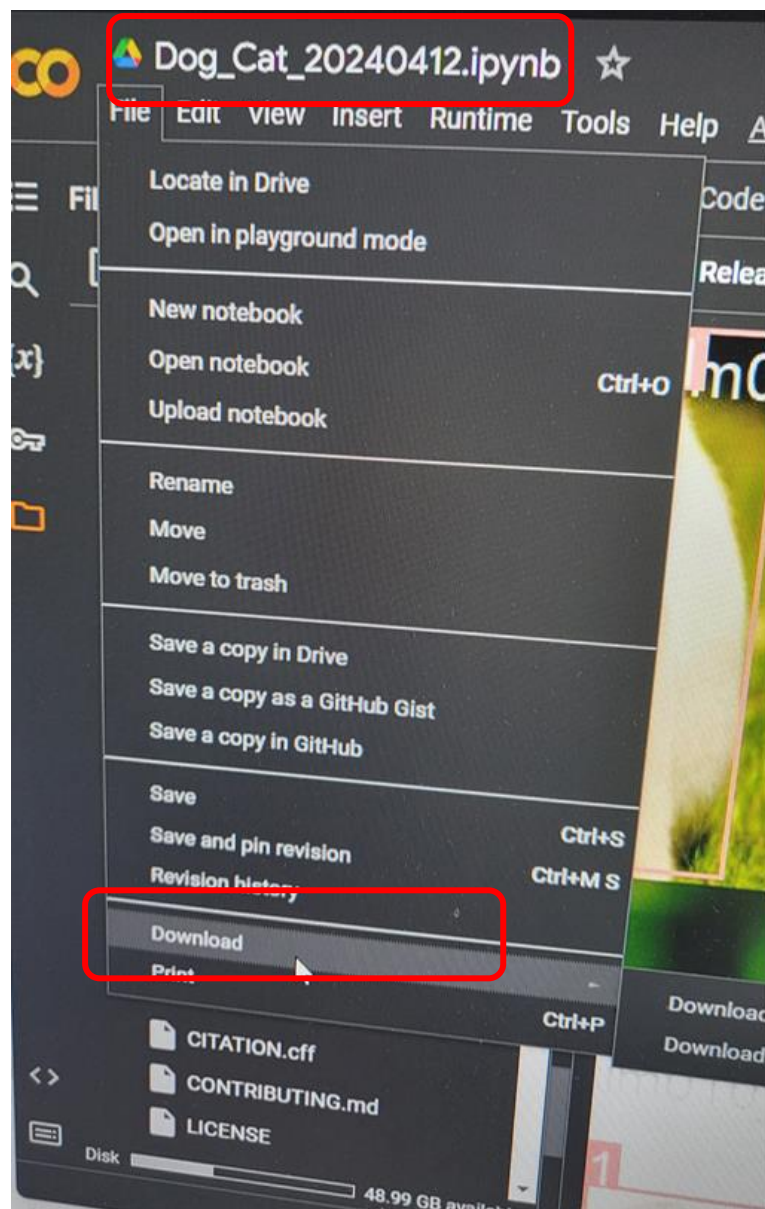
Validating runs/train/dog_cats_yolov5s_results/weights/best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
          Class    Images  Instances      P          R    mAP50  mAP50-95: 100% 1/1 [00:00<00:00, 13.33it/s]
          all         4          8   0.00667         1    0.239    0.0827
          cat         4          4   0.00659         1    0.176    0.077
          dog         4          4   0.00675         1    0.303    0.0884

Results saved to runs/train/dog_cats_yolov5s_results
```

. Training 결과 보기(runs/train/dog_cats_yolov5s_results)



. 필요하면 Download할 수 있음.



* 참고: <https://lynnshin.tistory.com/48>, <https://github.com/ultralytics/yolov5>

<https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb>

```
!python detect.py --weights /content/yolov5/runs/train/gun_yolov5s_results/weights/best.pt --img 416--source /content/dataset/test/pistol1.png
```

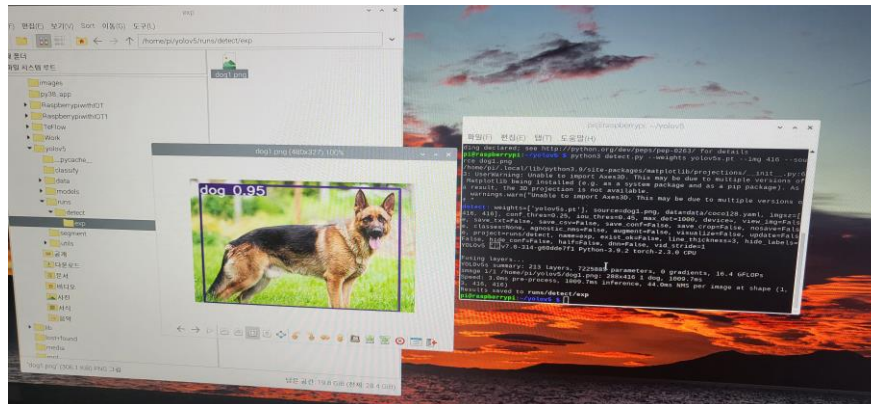
* pistol1.png를 dataset/test 폴더에 만들어서 넣은 후 detection. --source 0에서 cv2는 지원하지 않는다는 에러메시지

라즈베리파이 상에서 detect

\$ git clone <https://github.com/ultralytics/yolov5>

. 'yolov5s.pt', 'dog1.png' 를 라즈베리파이의 'yolov5' 폴더에 옮겨 놓은 후

\$ python3 detect.py --weights yolov5s.pt --img 416 --source dog1.png




```
$ python3 detect.py --weights yolov5s.pt --img 416 --source 0
```

(webcam 인식이 안될 때 \$ sudo modprobe bcm2835-v4l2)

