데이터 학습 (google colab)

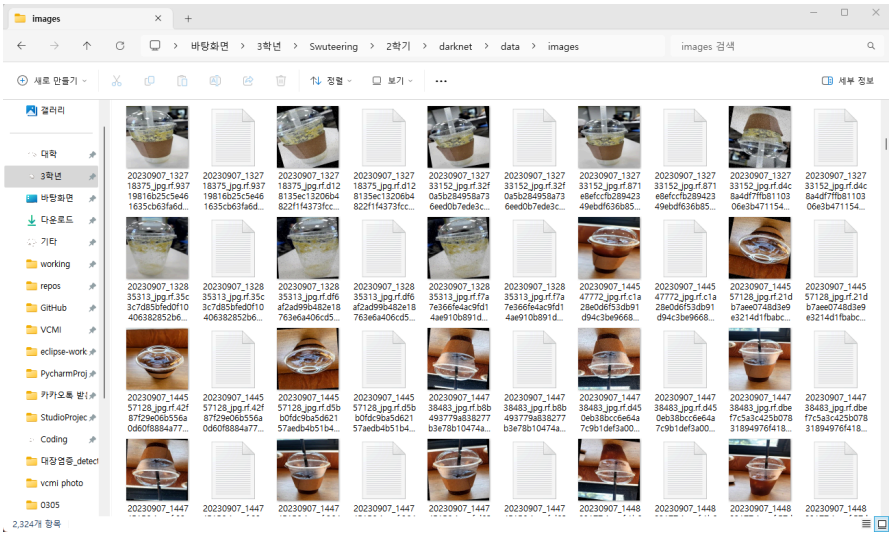
상태 완료

- yolo 깃허브 클론해오기

> GitHub - AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of
> YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet) - AlexeyAB/darknet
>
> https://github.com/AlexeyAB/darknet

git clone https://github.com/AlexeyAB/darknet.git

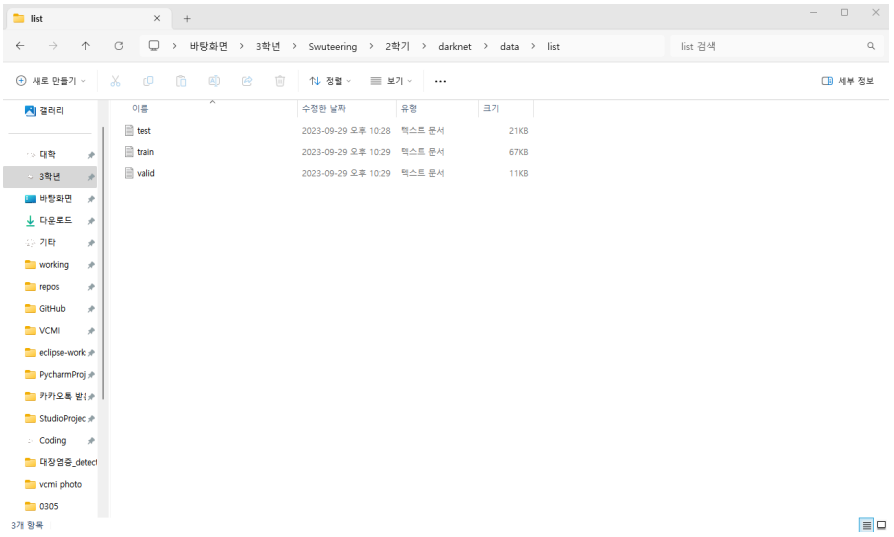
- darknet 폴더 수정
  - darknet/data 내에 images 폴더 생성
    - roboflow에서 다운로드 받은 모든 이미지와 annotation 파일들을 images 폴더 안에 업로드



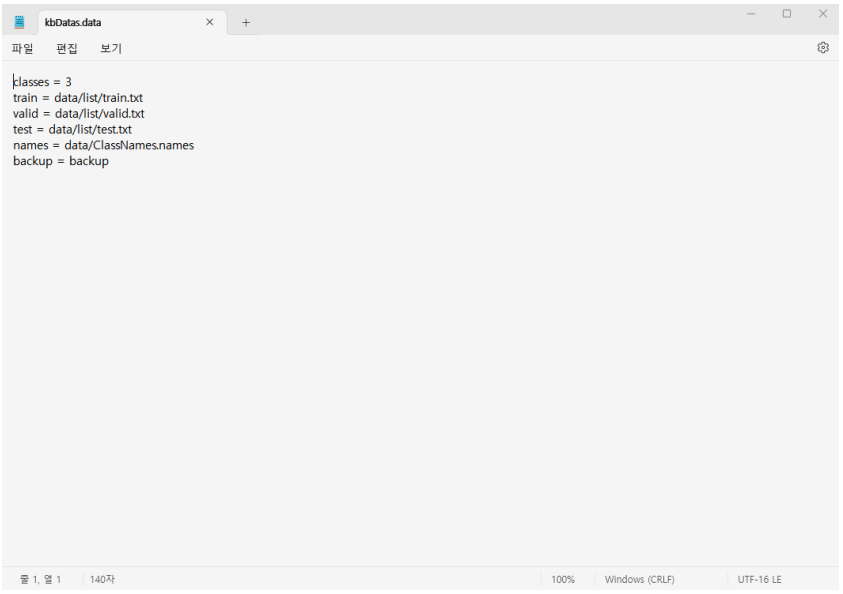
  - darknet/data 내에 list 폴더 생성
    - test에 쓸 데이터 경로, train에 쓸 데이터 경로, valid에 쓸 데이터 경로 지정
    - train 70, validation 10, test 20 으로 비율 설정

```
import os
import math

########################################################################################
# 당신의 데이터 셋 비율을 조정합니다.
# 세 변수의 합이 100이 되도록 해주세요.
train_set_ratio = 70
validation_set_ratio = 10
test_set_ratio = 20
########################################################################################

# file_path는 조정해주세요. 현 상태는 data/SplitDatAsTXT.py가 있고, data/images 안에 이미지들이 있습니다.
file_path = 'images\\'
```

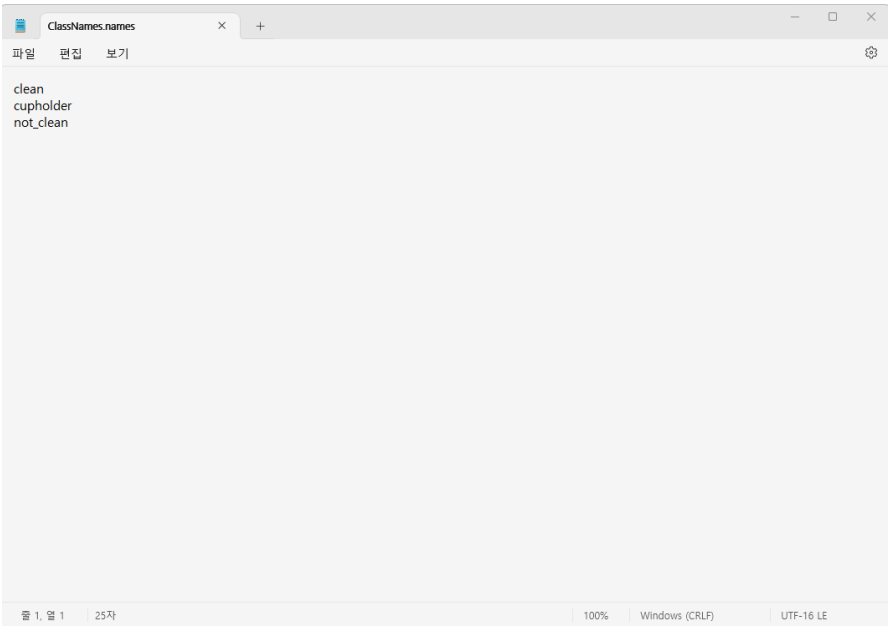
```
# 역슬래쉬가 아닌 /로 구분하기 위해 변경해줍니다.
file_path_as_originSlash = file_path[:-1] + "/"

file_names = os.listdir(file_path)
file_len = len(file_names)
i=0
while i<file_len:
    temp_fileName = file_names[i]
    if temp_fileName[-4:] == ".txt":
        file_names.pop(i)
        file_len = len(file_names)
        #print(str(i) + "   " + str(file_len) + "    " + temp_fileName)
    else: i += 1

# 1. Test Set 목록
print("****************** 1. Train Set 목록(비율: " + str(train_set_ratio) + "%) ******************")
for i in range(0, math.ceil(file_len * (train_set_ratio / 100))):
    print('data/' + file_path_as_originSlash + file_names[i])
final_index = math.ceil(file_len * (train_set_ratio / 100))

# 2. Validation Set 목록
print("****************** 2. Validation Set 목록(비율: " + str(validation_set_ratio) + "%) ******************"
for i in range(final_index, final_index+ math.ceil(file_len * (validation_set_ratio / 100))):
    print('data/' + file_path_as_originSlash + file_names[i])
final_index = final_index + math.ceil(file_len * (validation_set_ratio / 100))

# 3. Test Set 목록
print("****************** 3. Test Set 목록(비율: " + str(test_set_ratio) + "%) ******************")
for i in range(final_index, file_len):
    print('data/' + file_path_as_originSlash + file_names[i])
```



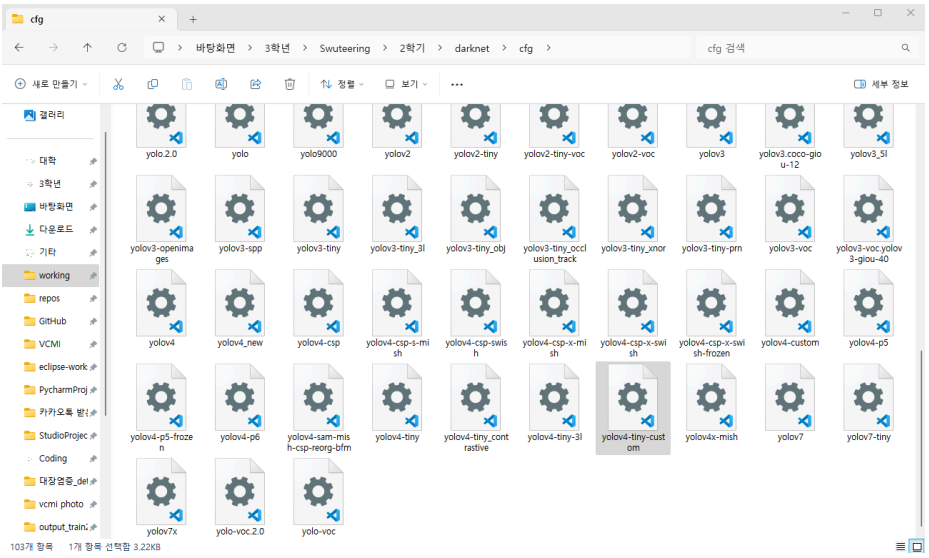
- .data 파일 만들기
  - 모델에 쓰일 자료들의 위치 지정



- .names 파일 만들기
  - 클래스 이름 지정해주기



- cfg 파일 수정
  - /darknet/cfg/yolov4-tiny-custom.cfg 파일을 찾는다



- 해당 파일을 다음과 같이 수정해준다
  1. subdivisions변수 수정
     - **7번째 줄**에 있는 **subdivisions를 subdivisions = 16**으로 수정
     - batch size인 64를 얼마나 나누어 학습을 할 것인지에 대한 설정 값을 조정
     - 기본 사이즈는 8, out of memory 방지를 위해 16으로 설정
     - 만약 Colab에서 문제가 생기면 32로 변경
  2. max_batches 변수 수정
     - **20번째** 줄에 있는 **max_batches**를 트레이닝 시킬 class의 수대로 수정
     - 클래스 갯수를 n이라고 한다면, n*2000
     - 우리는 3개의 class를 구분하기 때문에 **max_batches = 6000**
  3. steps 변수 수정
     - **22번째** 줄에 있는 **steps**변수를 수정
     - max_batches의 80%와 90%
     - 4800, 5400 으로 수정
  4. classes 변수 수정
     - **220번째 줄과 269번째 줄에 있는 classes변수** 수정
     - **classes = 트레이닝 시킬 class 개수 = 3**
  5. filters 변수 수정
     - **212번째 줄과 263번째 줄의 filters 변수** 수정
     - **filters = (classes+coords+1)*<number of mask>**
     - **classes = n, coords = 4, <number of mask> = 3**
     - **filters=(n+5)*3 ⇒ 24로 수정**
- darknet 폴더 내 모든 파일들을 kb_Detection_Darknet 의 이름으로 압축
- colab에서 yolo v4 tiny 학습
  - 경로 설정
    - 내 드라이브 > YOLOv4-tiny 폴더 생성
    - 내 드라이브 > YOLOv4-tiny > backup 폴더 생성
    - 내 드라이브 > Colab Notebooks > kb_Detection_Darknet 압축 파일 업로드
    - 내 드라이브 > Colab Notebooks > train 코드를 위한 colab 파일 생성

- yolov4-tiny.ipynb
  - 런타임 유형을 T4로 변경한 후 실행

```
from google.colab import drive
drive.mount('/content/drive')
```

```
!mkdir /content/darknet

!unzip "/content/drive/MyDrive/Colab Notebooks/kb_Detection_Darknet.zip" -d "/content/darknet"
```

```
%cd /content/darknet

!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
!sed -i 's/LIBSO=0/LIBSO=1' Makefile

!make
!chmod +x ./darknet
```

```
!rm -rf /content/darknet/backup
!ln -s /content/drive/'MyDrive'/YOLOv4/backup /content/darknet

!sudo apt install dos2unix

!dos2unix ./data/list/train.txt
!dos2unix ./data/list/valid.txt
!dos2unix ./data/list/test.txt
!dos2unix ./data/ClassNames.names
!dos2unix ./data/kbDatas.data
!dos2unix ./cfg/yolov4-custom.cfg
```

```
def imShow(path):
  import cv2
  import matplotlib.pyplot as plt
  %matplotlib inline

  image = cv2.imread(path)
  height, width = image.shape[:2]
  resized_image = cv2.resize(image, (3*width, 3*height), interpolation = cv2.INTER_CUBIC)

  fig = plt.gcf()
  fig.set_size_inches(18, 10)
  plt.axis("off")
  plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
  plt.show()

def upload():
  from google.colab import files
  uploaded = files.upload()
  for name, data in uploaded.items():
    with open(name, 'wb') as f:
      f.write(data)
      print('saved file', name)
```

```
def download(path):
  from google.colab import files
  files.download(path)
```

```
%cd /content/darknet

!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4.conv.137
```

```
%cd /content/darknet/

# yolo 학습
!time ./darknet detector train data/kbDatas.data cfg/yolov4-custom.cfg yolov4.conv.137 -dont_show -map
```

```
# train loss 이미지 출력 및 valid data에 대한 mAP 함께 출력
imShow('chart.png')
```

```
# test 데이터에 대한 prediction 출력
!time ./darknet detector test data/kbDatas.data cfg/yolov4-custom.cfg backup/yolov4-custom_final.weights -dont
download('result_v4.txt')
```

```
# 이미지 예측
!time ./darknet detector test data/kbDatas.data cfg/yolov4-custom.cfg backup/yolov4-custom_final.weights /cont
imShow('predictions.jpg')
download('predictions.jpg')
```