Google Colab을 이용한 YOLOV4 훈련

1. Colab 이용
   1. Colab사이트 접속

* 브라우저에서 코랩 사이트를 접속한다.
* 링크: <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ko>
  1. 내 노트 생성
* 메뉴 표시줄 파일을 클릭하고 새 노트 클릭
* 사용할 수 있는 내 Colab 노트북이 열립니다.
* 메뉴 표시줄 수정을 클릭
* **노트 설정을 누르고 하드웨어 가속기를 GPU로 변경**

1. Google drive 사전 세팅

2-1) Google drive에 yolov4 폴더를 만든다.

2-2) 만든 yolov4폴더로 이동 후 training폴더를 생성한다.

1. Darknet 세팅

3-1) Colab 노트북의 생성한 노트로 이동

3-2) Google drive를 마운트한다.

* 코드

%cd ..

from google.colab import drive

drive.mount('/content/gdrive')

3-3) 심볼릭 링크를 만들어 경로를 설정한다.

* 코드

!ln -s /content/gdrive/My\ Drive/ /mydrive

%cd /mydrive/yolov4

3-4) git에서 darknet 폴더를 복제한다.

* 코드

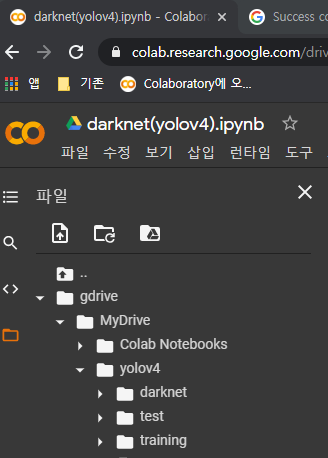
!git clone <https://github.com/AlexeyAB/darknet>

3-5) darknet폴더 이동

%cd darknet

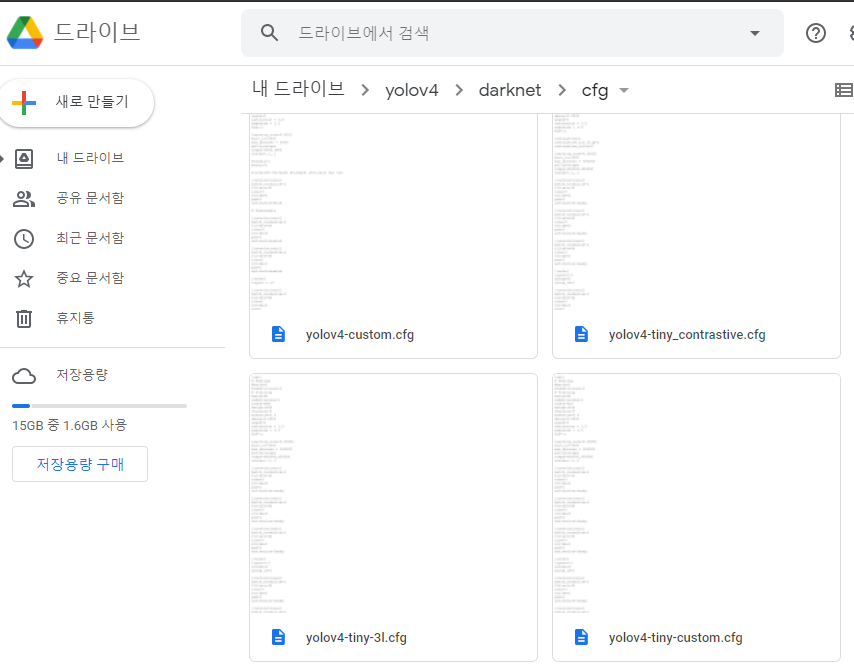
3-6) 현재까지 세팅 확인

* 사전에 만든 yolov4폴더와 training폴더를 확인 할 수 있으며, 또한 darknet폴더도 복제된 것을 확인 할 수 있다.



3-7) 훈련을 위한 파일 생성

* 사전 준비(YOLO\_Label.docx)을 통해 준비된 라벨링 데이터를 obj폴더에 사진(jpg)과 라벨 데이터(txt)를 넣는다.
* obj폴더는 압축 시킨다.
* cfg파일 수정
  + darknet 폴더 내 cfg폴더의 yolov4-custom.cfg를 다운



* + 다운 받은 cfg를 열고 Colab사양에 맞게 변경한다.
    - **batch = 64**
    - **subdivision = 16**
    - **width = 416 height=416**
    - **max\_batches = (classes \* 2000)**
    - **steps = (max\_batches의 80%값), (max\_batches의 90%값)**
    - **[yolo]부분과 바로 위 [convolutional]을 변경**
      * **[yolo]의 filter 수정**
        + **filter = (classes + 5) \* 3**
        + **해당 라인 = 961, 1049, 1137라인이다.**
      * **[convolutional]의 classes 수정**
        + **classes = 훈련시킬 대상의 개수**
        + **해당 라인 = 968, 1049, 1137라인이다.**
* obj.data와 obj.names를 수정
  + obj.data 수정

classes = 1

train = data/train.txt

valid = data/test.txt

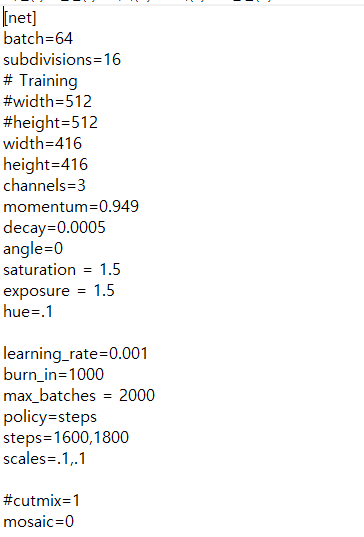
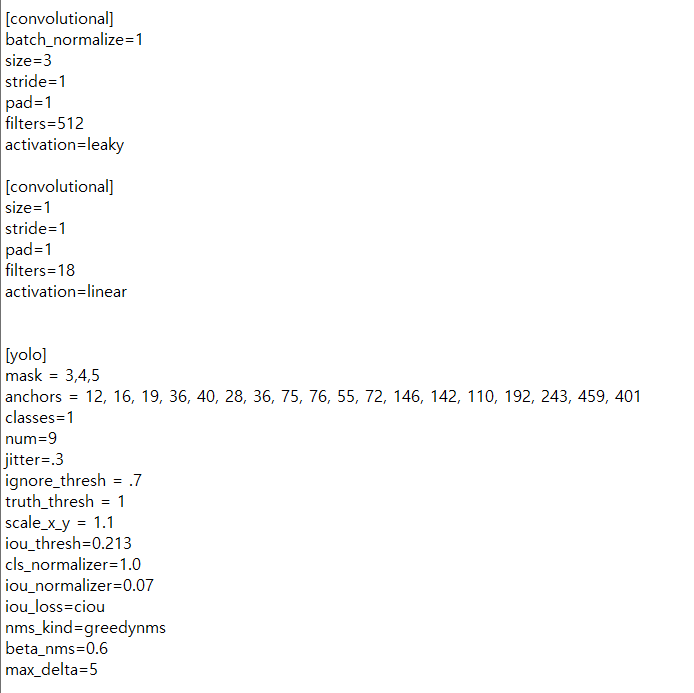
names = data/obj.names

backup = /mydrive/yolov4/training/

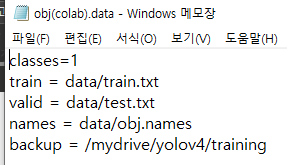
* + obj.names

cat

* + 수정 확인
    - yolov4-custom.cfg 수정

* + - obj.data



* process.py 스크립트 파일 생성

import glob, os

# Current directory

current\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))

print(current\_dir)

current\_dir = 'data/obj'

# Percentage of images to be used for the test set

percentage\_test = 10;

# Create and/or truncate train.txt and test.txt

file\_train = open('data/train.txt', 'w')

file\_test = open('data/test.txt', 'w')

# Populate train.txt and test.txt

counter = 1

index\_test = round(100 / percentage\_test)

for pathAndFilename in glob.iglob(os.path.join(current\_dir, "\*.jpg")):

title, ext = os.path.splitext(os.path.basename(pathAndFilename))

if counter == index\_test:

counter = 1

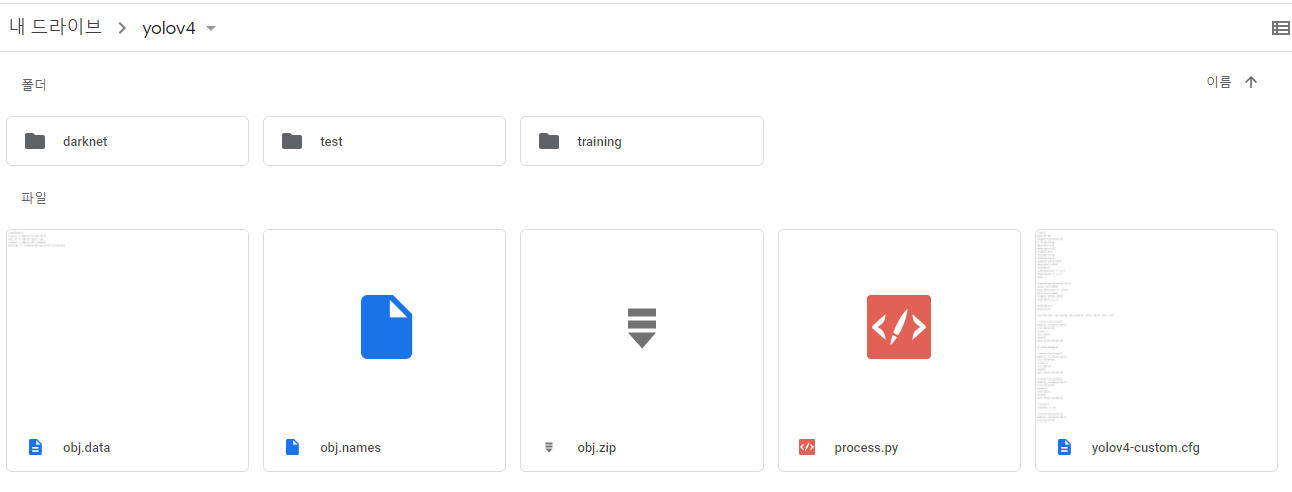
file\_test.write("data/obj" + "/" + title + '.jpg' + "\n")

else:

file\_train.write("data/obj" + "/" + title + '.jpg' + "\n")

counter = counter + 1

* 준비된 파일을 Google drive 속 yolov4폴더에 업로드 한다.
  + 파일 확인
    - 준비한 5개 파일이 yolov4폴더에 업로드 된 것을 확인 할 수 있다.



3-8) 메이크 파일을 변경하여 opencv 및 GPU 활성화

!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

3-9) 파일 수정 후 메이크

!make

3-10) 파일 압축 해제 및 세팅

!unzip /mydrive/yolov4/obj.zip -d data/obj

!cp /mydrive/yolov4/yolov4-custom.cfg cfg

!cp /mydrive/yolov4/obj.names data

!cp /mydrive/yolov4/obj.data data

!cp /mydrive/yolov4/process.py .

!python process.py

!ls data/

3-11) 사전 훈련된 YOLOV4 가중치 다운로드

!wget https://github.com/AlexeyAB/darknet/releases/download/darknet\_yolo\_v3\_optimal/yolov4.conv.137

3-12) 훈련하기

!./darknet detector train data/obj.data cfg/yolov4-custom.cfg yolov4.conv.137 -dont\_show -map

3-13) Colab 연결 유지 (세션 최대 12시간 유지)

* 코렙 사용시 학습 도중 세션이 만료되면 정보가 사라질 수 있어 세션 종료 방지 코드 입력
* Ctrl + Shift + i를 누르고 Console로 이동 다음 코드를 붙여넣고 Enter입력
* function ClickConnect(){
* console.log("Working");
* document
* .querySelector('#top-toolbar > colab-connect-button')
* .shadowRoot.querySelector('#connect')
* .click()
* }
* setInterval(ClickConnect,60000)

1. 결과확인

4-1) 결과 그래프 확인을 위한 imShow함수 추가

* 코드

(\*주의! plt.axis(“off”)부분은 붙여넣기 후 재작성 필요)

# define helper functions

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()

* 그래프 확인 코드

imShow('chart.png')

4-2) MAP(평균 정밀도) 확인

* 코드
  + Map가 높을수록 최상의 결과를 제공한다.
  + XXXX는 해당 반복 번호를 기입(ex) yolov4-custom\_1000.weights면 1000기입)

!./darknet detector map data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom\_xxxx.weights -points 0

* 코드 실행 오류시
  + 파일 권한 변경 실행

!chmod +x ./darknet

4-3) 학습 사진을 제외한 다른 사진을 이용해 결과 확인

* 확인을 위해 cfg수정 필요
  + 코드

%cd cfg

!sed -i 's/batch=64/batch=1/' yolov4-custom.cfg

!sed -i 's/subdivisions=16/subdivisions=1/' yolov4-custom.cfg

%cd ..

* 결과 확인
  + 확인을 위한 데이터 준비
    - 준비한 사진을 Google drive속 yolov4폴더에 test폴더 생성 후 사진 업로드
    - 코드

!./darknet detector test data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom\_best.weights /mydrive/yolov4/test/cat5.jpg -thresh 0.3

imShow('predictions.jpg')

1. 재시작 방법

5-1) 순서

* 3-2)부분 Google drive 마운트 재실행
* 3-3)부분 심볼릭 링크 재설정
* 결과 확인을 했을 때 또는 cfg파일이 변경됬을 때
  + 3-5)부분 darknet 폴더 이동
* 3-8)부분 메이크 파일 변경 재실행
* 3-9)부분 메이크 실행
* 3-10)부분 압축 해제 및 세팅 재실행
* 마지막 last weights를 이용해서 학습 재시작
  + 코드
* !./darknet detector train data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom\_last.weights -dont\_show -map
* 3-13) Colab 연결 유지 코드 재실행

\*) 출처

- 사용자 지정 YOLOV4 개체 탐지기 교육(Google Colab사용)

<https://ichi.pro/ko/sayongja-jijeong-yolov4-gaeche-tamjigi-gyoyug-google-colab-sayong-6710443722856>