Cough & Sneeze Detection Using Yolov3

Introduction:

Activity recognition aims to recognize the actions and goals of one or more persons from a series of observations on the persons' actions and the environmental conditions using artificial intelligence. Here we have used Activity Detection for Cough and Sneeze detection.

Use Cases:

- Health Sector.
- In Shopping Areas for prevention during pandemic.
- Hospitals.
- Can be used in Schools or any other area.

Dataset Collection:

For dataset we've used videos from biisc dataset of Activity Detection (BII Sneeze-Cough Human Action Video Dataset). Because we focus only on two actions so we only extract the cough and sneeze videos from dataset and extract frames out of them.

Dataset: https://web.bii.a-star.edu.sg/~chengli/FluRecognition.htm

Code:

```
#For sneeze and cough both

import os

import cv2

pathOut = r"/Users/Aveen Faheem/Desktop/sneezeimg/"

count = 0

counter = 1

listing = os.listdir(r'/Users/Aveen Faheem/Desktop/sneeze')

for vid in listing:

    vid = r"/Users/Aveen Faheem/Desktop/sneeze/"+vid

    cap = cv2.VideoCapture(vid)
```

```
count = 0
counter += 1
success = True
while success:
    success,image = cap.read()
    print('Read a new frame:',success)
    if count%30 == 0 :
        cv2.imwrite(pathOut + 'video%d'%counter + 'sneeze%d.jpg'%count ,image)
    count+=1
```

For Re-naming the files

```
import os
os.getcwd()
collection = "/Users/Aveen Faheem/Desktop/sneezeimg"
for i, filename in enumerate(os.listdir(collection)):
    os.rename("/Users/Aveen Faheem/Desktop/sneezeimg/" + filename, "/Users/Aveen Faheem/Desktop/sneezeimg/"+ "sneeze" + str(i) + ".jpg")
```

Annotations:

For annotations of these images I've used LabelImg. You can use it by following these steps.

1. Pip install labelImg
2. Go to command prompt and write labelmg
3. It will pop-up and you can annotate your images.
4. Map the desired area, label it and save in .txt format.
5. Make sure you are keeping the classes values right.

Methodology:

Custom Activity Detection on yolov3 of Cough and Sneeze:

For training yolov3 model on collab, I've followed these steps.

1. Download Darknet
   https://github.com/pjreddie/darknet

Download darknet and make changes in Make file
GPU = 1
CUDNN=1
OPENCV=1

Replace new make file with old one in colab.

2. Download Pre-trained weights for yolov3
   https://pjreddie.com/media/files/darknet53.conv.74

   Keep these weights in darknet folder on colab.

3. Changes in cfg file
   In directory **darknet\cfg**, creating a copy of ***"yolov3.cfg"*** in the same folder and rename it.
   Make these changes according to your number of classes.
   # Line 8 & 9:
   width = 416, height = 416
   # Line 20
   max_batches = 6000
   # Line 22
   steps = 5400
   #Line 603, 689, 776:
   filters = 18
   #Line 610, 696, 783:
   classes = 1

   Formula for filters = (filters = (classes + 5)*3)
   filters=(2+5)*3 = 21 **classes = 2.**

   After making changes add this cfg file in cfg folder.

4. Detector.c file

   From directory **darknet\examples** folder, open file **"*detector*.c" . At line 138, modify this line as below:**

   if(i%1000==0 || (i < 1000 && i%100 == 0)){

5. Split Train and Test data
   Code:

```
import glob, os
## Current directory
current_dir = '/content/drive/My Drive/backup/'# Path to the current directory
#print(current_dir)

# 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('train.txt', 'w')
file_test = open('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(current_dir + title + '.jpg' + "\n")
    else:
        file_train.write(current_dir + title + '.jpg' + "\n")
        counter = counter + 1
```

After dividing train and test data, change their paths according to colab path to access data from drive. Keep all these files along with data on drive as well.

6. Data and names file
   Change names file according to given annotations 0 or 1 to which activity.
   0 cough
   1 sneeze so make names file as
   Cough
   Sneeze
   For data file add these things
   classes= 1 #number of objects, in our case is 1
   train = data/train.txt
   test = data/test.txt
   names = data/yolo.names
   backup = backup

7. Setting Colab

- After downloading darknet to colab.
- Replace makefile
- Add cfg file in cfg folder
- Add(train.txt, test.txt, names file, data file) in data folder
- Change detector.c in examples folder
- Add darknet53.conv74 to darknet folder.

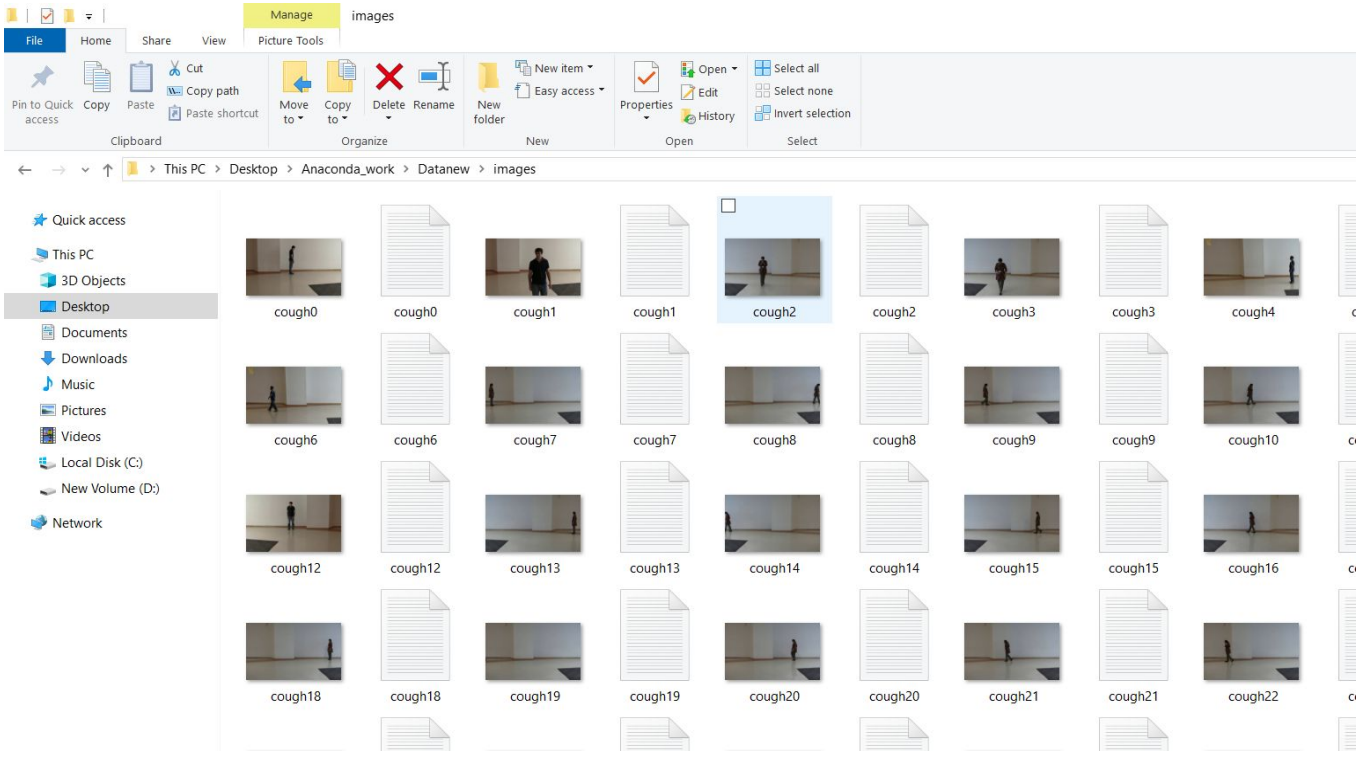
For more guidance follow these links

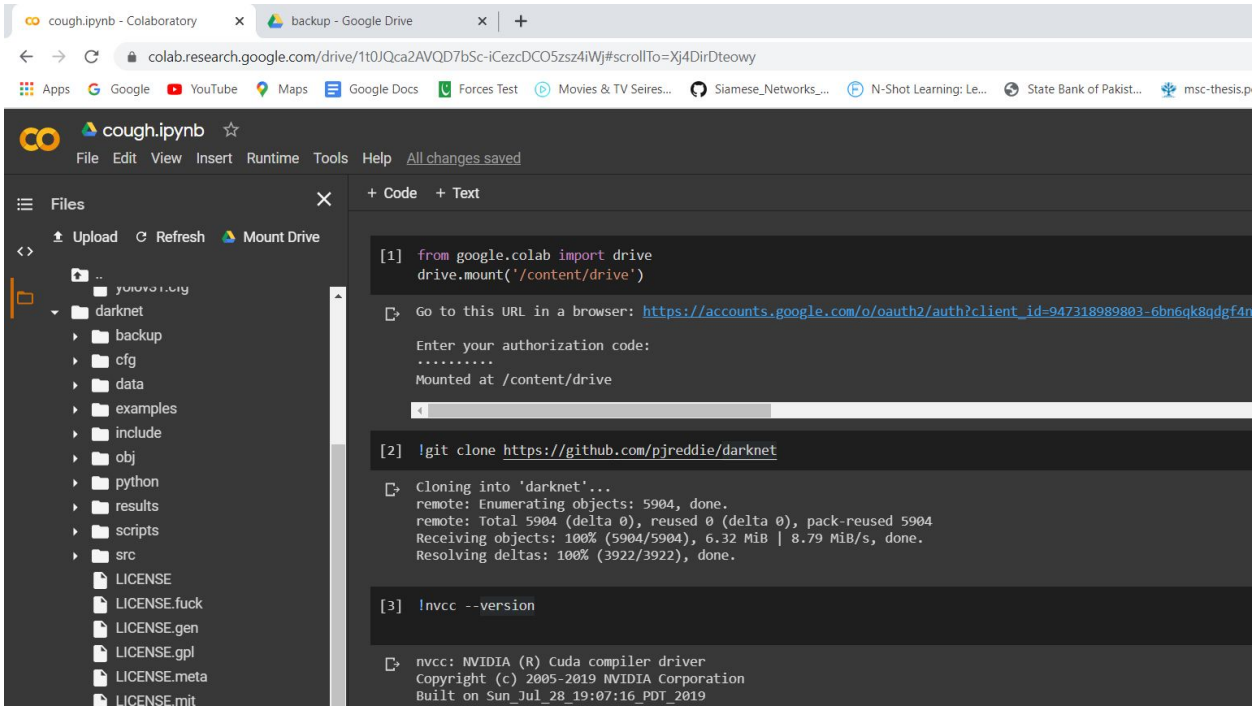
https://medium.com/@quangnhatnguyenle/how-to-train-yolov3-on-google-colab-to-detect-custom-objects-e-g-gun-detection-d3a1ee43eda1

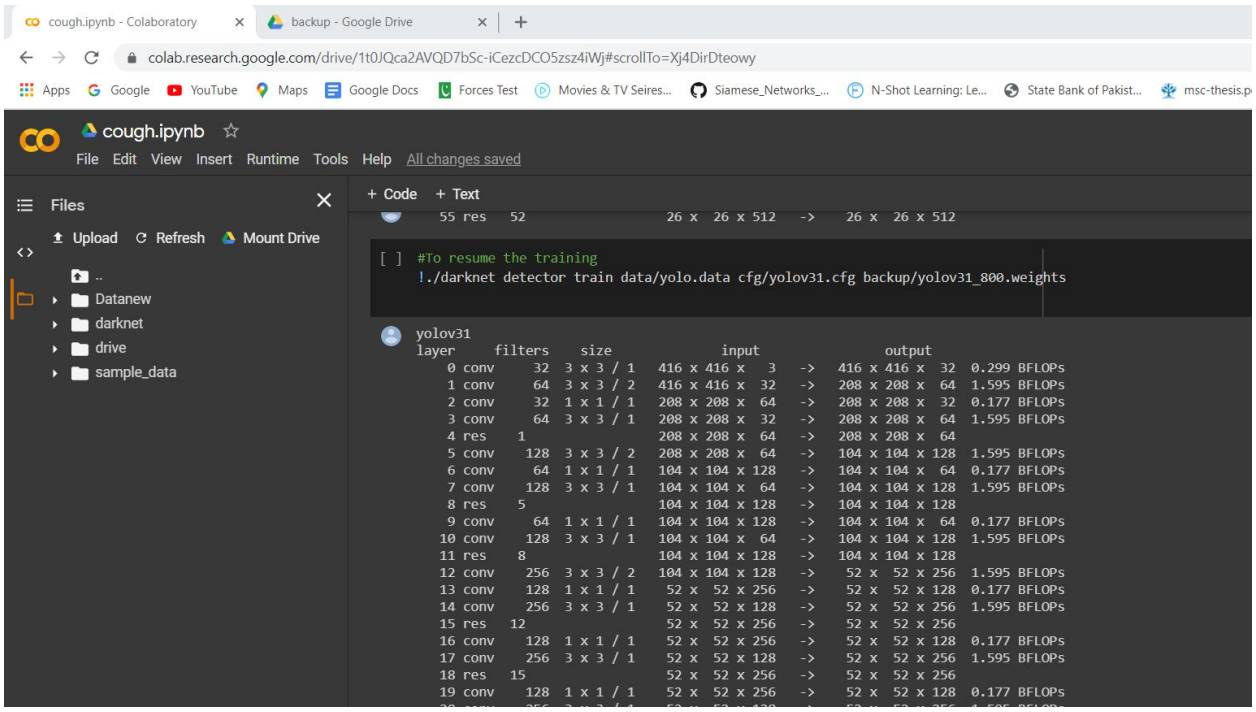
https://colab.research.google.com/drive/13-9pAz9nxUYm-0LlNV9tVtS57g8mHAOb#scrollTo=fBQgzSpY7Vkh

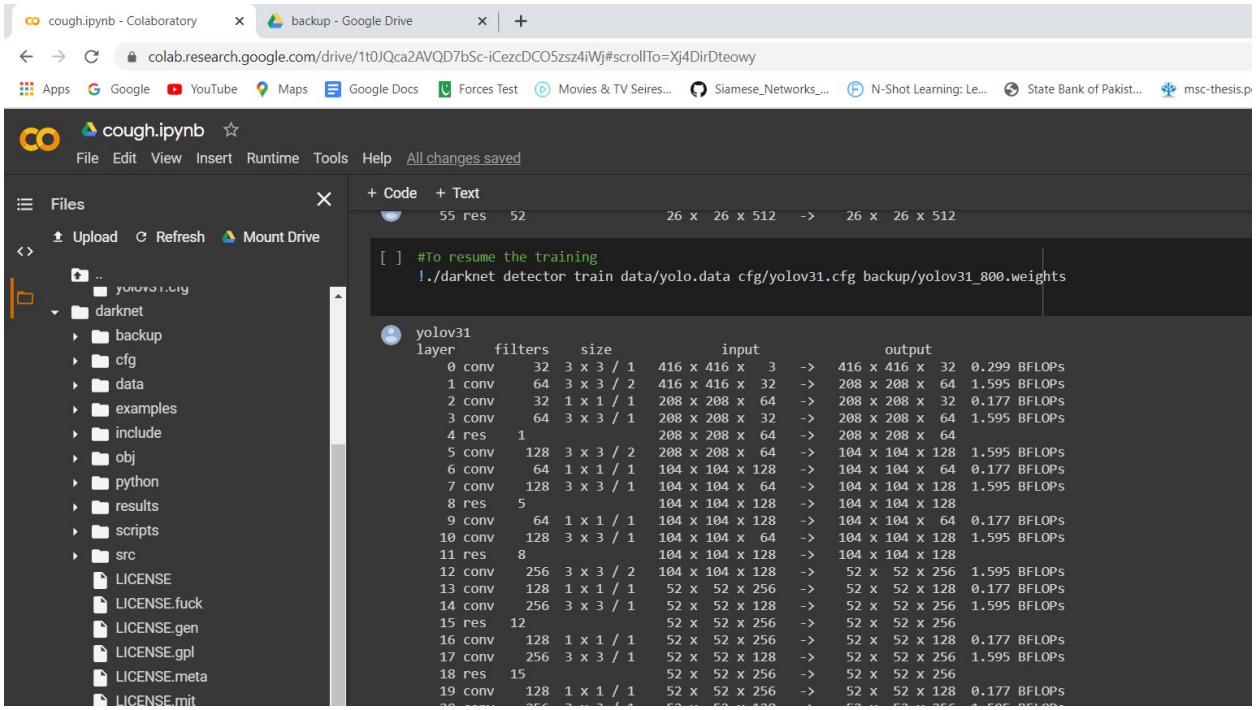
https://github.com/pjreddie/darknet/issues/174

Training:

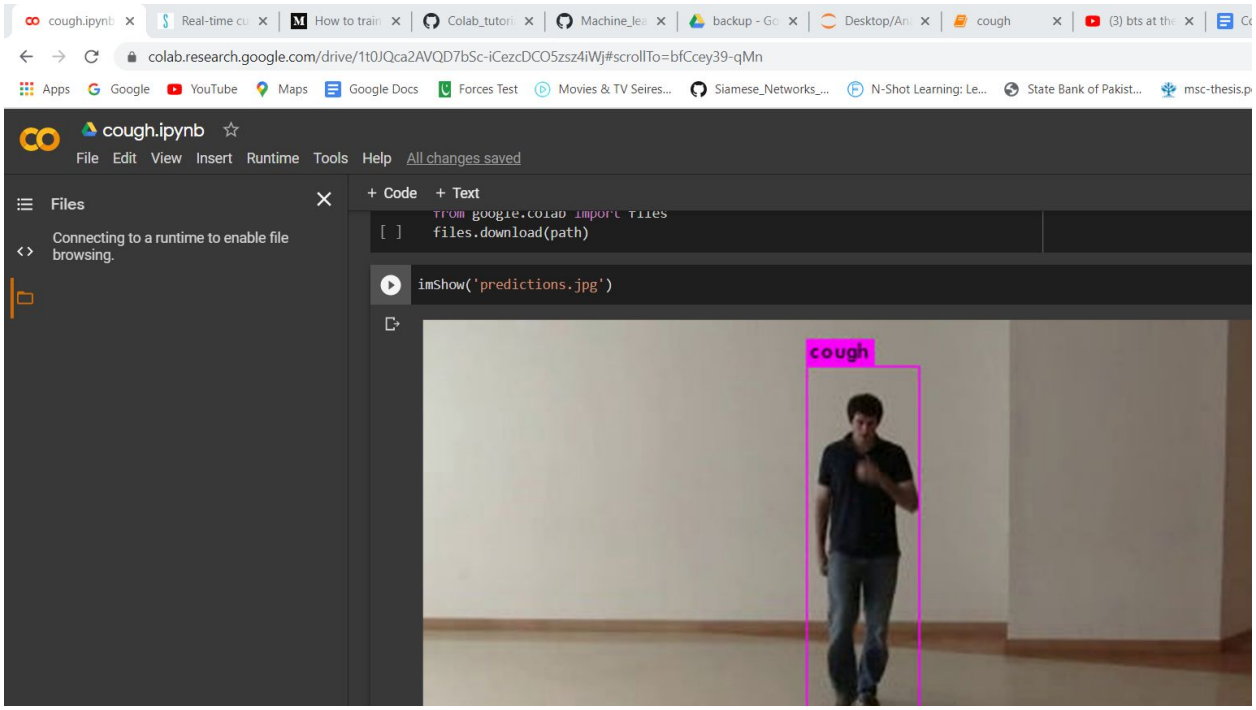


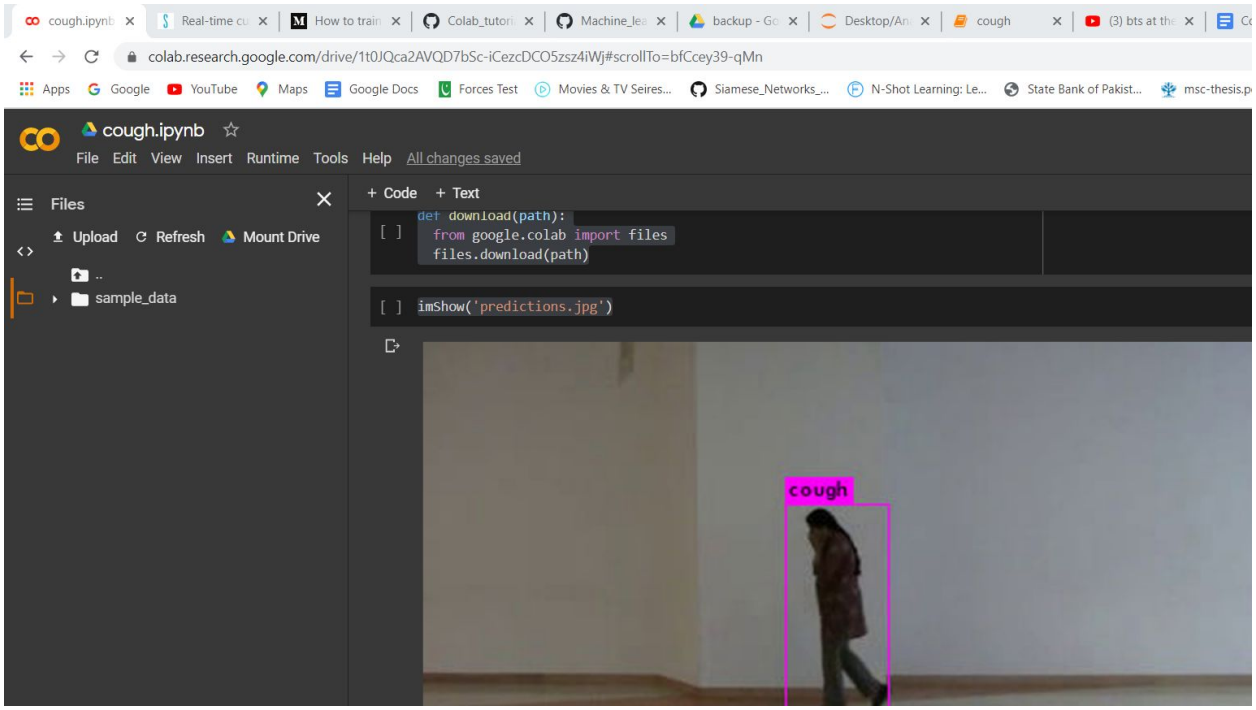

cough.ipynb - Colaboratory
backup - Google Drive
colab.research.google.com/drive/1t0JQca2AVQD7bSc-iCezcDCO5zsz4iWj#scrollTo=Xj4DirDteowy
Apps Google YouTube Maps Google Docs Forces Test Movies & TV Seires... Siamese_Networks_... N-Shot Learning: Le... State Bank of Pakist... msc-thesis.p
cough.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
Upload Refresh Mount Drive
darknet
backup
cfg
data
examples
include
obj
python
results
scripts
src
LICENSE
LICENSE.fuck
LICENSE.gen
LICENSE.gpl
LICENSE.meta
LICENSE.mit
+ Code + Text
[1] from google.colab import drive
drive.mount('/content/drive')
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n
Enter your authorization code:
.....
Mounted at /content/drive
[2] !git clone https://github.com/pjreddie/darknet
Cloning into 'darknet'...
remote: Enumerating objects: 5904, done.
remote: Total 5904 (delta 0), reused 0 (delta 0), pack-reused 5904
Receiving objects: 100% (5904/5904), 6.32 MiB | 8.79 MiB/s, done.
Resolving deltas: 100% (3922/3922), done.
[3] !nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:07:16_PDT_2019

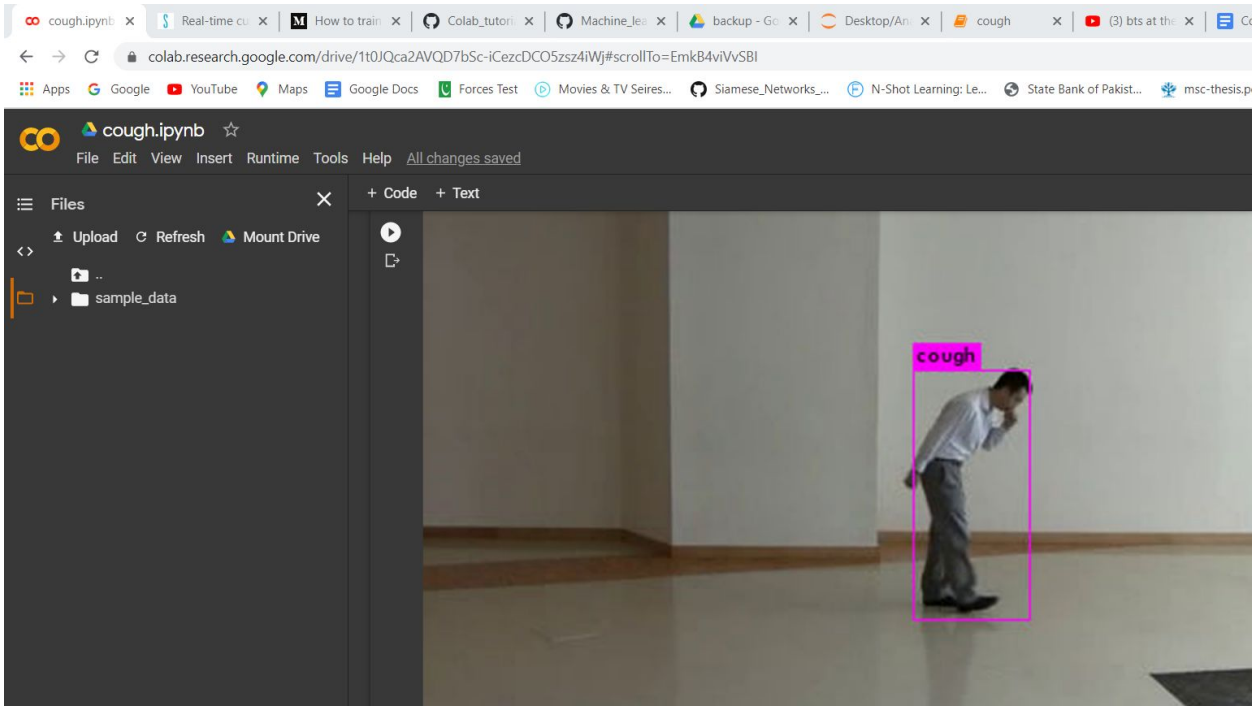

cough.ipynb - Colaboratory
backup - Google Drive
colab.research.google.com/drive/1t0JQca2AVQD7bSc-iCezcDCO5zsz4iWj#scrollTo=Xj4DirDteowy
Apps Google YouTube Maps Google Docs Forces Test Movies & TV Seires... Siamese_Networks_... N-Shot Learning: Le... State Bank of Pakist... msc-thesis.p
cough.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
Upload Refresh Mount Drive
Datanew
darknet
drive
sample_data
+ Code + Text
55 res 52 26 x 26 x 512 -> 26 x 26 x 512
[ ] #To resume the training
!./darknet detector train data/yolo.data cfg/yolov31.cfg backup/yolov31_800.weights
yolov31
layer filters size input output
0 conv 32 3 x 3 / 1 416 x 416 x 3 -> 416 x 416 x 32 0.299 BFLOPs
1 conv 64 3 x 3 / 2 416 x 416 x 32 -> 208 x 208 x 64 1.595 BFLOPs
2 conv 32 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 32 0.177 BFLOPs
3 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64 1.595 BFLOPs
4 res 1 208 x 208 x 64 -> 208 x 208 x 64
5 conv 128 3 x 3 / 2 208 x 208 x 64 -> 104 x 104 x 128 1.595 BFLOPs
6 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BFLOPs
7 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BFLOPs
8 res 5 104 x 104 x 128 -> 104 x 104 x 128
9 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BFLOPs
10 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BFLOPs
11 res 8 104 x 104 x 128 -> 104 x 104 x 128
12 conv 256 3 x 3 / 2 104 x 104 x 128 -> 52 x 52 x 256 1.595 BFLOPs
13 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BFLOPs
14 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BFLOPs
15 res 12 52 x 52 x 256 -> 52 x 52 x 256
16 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BFLOPs
17 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BFLOPs
18 res 15 52 x 52 x 256 -> 52 x 52 x 256
19 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BFLOPs

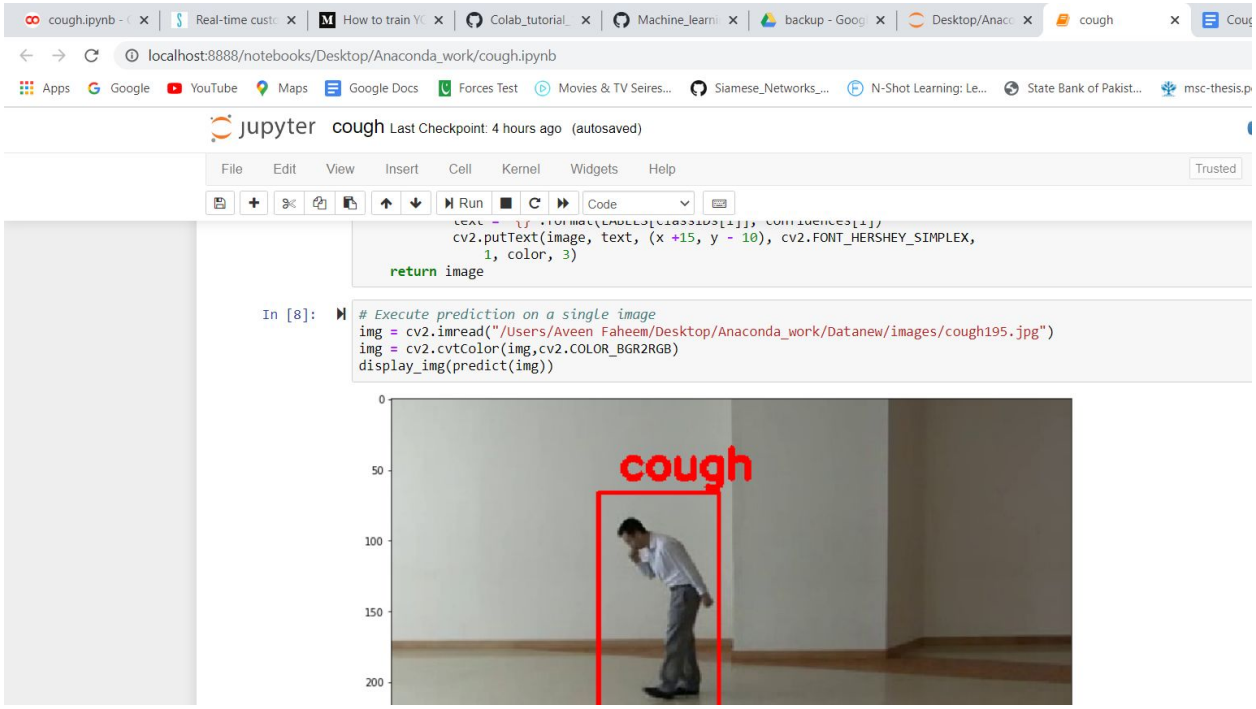

cough.ipynb - Colaboratory
backup - Google Drive
colab.research.google.com/drive/1t0JQca2AVQD7bSc-iCezcDCO5zsz4iWj#scrollTo=Xj4DirDteowy
cough.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
Upload Refresh Mount Drive
darknet
backup
cfg
data
examples
include
obj
python
results
scripts
src
LICENSE
LICENSE.fuck
LICENSE.gen
LICENSE.gpl
LICENSE.meta
LICENSE.mit
+ Code + Text
55 res 52 26 x 26 x 512 -> 26 x 26 x 512
[ ] #To resume the training
!./darknet detector train data/yolo.data cfg/yolov31.cfg backup/yolov31_800.weights
yolov31
layer filters size input output
0 conv 32 3 x 3 / 1 416 x 416 x 3 -> 416 x 416 x 32 0.299 BFLOPs
1 conv 64 3 x 3 / 2 416 x 416 x 32 -> 208 x 208 x 64 1.595 BFLOPs
2 conv 32 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 32 0.177 BFLOPs
3 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64 1.595 BFLOPs
4 res 1 208 x 208 x 64 -> 208 x 208 x 64
5 conv 128 3 x 3 / 2 208 x 208 x 64 -> 104 x 104 x 128 1.595 BFLOPs
6 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BFLOPs
7 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BFLOPs
8 res 5 104 x 104 x 128 -> 104 x 104 x 128
9 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BFLOPs
10 conv 128 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 128 1.595 BFLOPs
11 res 8 104 x 104 x 128 -> 104 x 104 x 128
12 conv 256 3 x 3 / 2 104 x 104 x 128 -> 52 x 52 x 256 1.595 BFLOPs
13 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BFLOPs
14 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BFLOPs
15 res 12 52 x 52 x 256 -> 52 x 52 x 256
16 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BFLOPs
17 conv 256 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BFLOPs
18 res 15 52 x 52 x 256 -> 52 x 52 x 256
19 conv 128 1 x 1 / 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BFLOPs

Testing:


cough.ipynb
colab.research.google.com/drive/1t0JQca2AVQD7bSc-iCezcDCO5zsz4iWj#scrollTo=bfCcey39-qMn
cough.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
Connecting to a runtime to enable file browsing.
+ Code + Text
files.download(path)
imShow('predictions.jpg')
cough


cough.ipynb
colab.research.google.com/drive/1t0JQca2AVQD7bSc-iCezcDCO5zsz4iWj#scrollTo=bfCcey39-qMn
cough.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
Upload Refresh Mount Drive
sample_data
+ Code + Text
def download(path):
from google.colab import files
files.download(path)
imShow('predictions.jpg')
cough





Results:

On 3000 weights results are right and they are detecting the right part of images. In videos; on dataset videos it is giving right results but to new videos it is that much accurate still testing more to see results.