Yolov4 for Mask Detection

Aim: To detect if person wearing mask or not using Yolov4 Object detection model.

Datasets: The dataset we are using for Mask detection contain 2 classes: a) person with mask and b) person without mask. These dataset contains of around 1510 images, as well as their bounding boxes in the YOLO format labelled text files.

Link of Dataset: https://www.kaggle.com/datasets/techzizou/labeled-mask-dataset-yolo-darknet

About Yolov4 Model:

YOLOv4 was a real-time object detection model published in April 2020 that achieved state-of-the-art performance on the COCO dataset. It works by breaking the object detection task into two pieces, regression to identify object positioning via bounding boxes and classification to determine the object's class.

YOLO an acronym for 'You only look once', is an object detection algorithm that divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself.

Link of Yolov4 Github: https://github.com/AlexeyAB/darknet

Steps to train model:

1) Create yolov4 and training folders in your drive.

Create a folder named ***yolov4*** in your drive.

Next, create another folder named ***training*** inside the ***yolov4*** folder. This is where we will save our trained weights (This path is mentioned in the ***obj.data*** file which we will upload later)

2) Mount drive, link your folder and navigate to /mydrive/yolov4 folder.

```
#mount drive
%cd ..
from google.colab import drive
drive.mount('/content/gdrive')

# this creates a symbolic link so that now the path /content/gdrive/My\ Drive/ is equal to /mydrive
!ln -s /content/gdrive/My\ Drive/ /mydrive

# list the contents of /mydrive
!ls /mydrive

#Navigate to /mydrive/yolov4
%cd /mydrive/yolov4
```

3) Clone darknet git repository.

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

4) Create & upload the following files which we need for training a custom detector.

a. Labeled Custom Dataset

b. Custom cfg file

c. obj.data and obj.names files

d. process.py file (to create train.txt and test.txt files for training)

5) Make changes in the makefile to enable OPENCV and GPU.

```
%cd 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
```

6) Run make command to build Darknet.

```
!make
```

7) Copy all the files from the ***yolov4*** folder to the darknet directory.

```
# Copy the yolov4-custom.cfg file so that it is now in /darknet/cfg/ folder
!cp /mydrive/yolov4/yolov4-custom.cfg cfg

# Copy the obj.names and obj.data files from your drive so that they are now in /darknet/data/ folder
!cp /mydrive/yolov4/obj.names data
!cp /mydrive/yolov4/obj.data  data

# Copy the process.py file to the current darknet directory

!cp /mydrive/yolov4/process.py .
```

8) Run the **process.py** python script to create the **train.txt** & **test.txt** files inside the data folder

```
!python process.py

# list the contents of data folder to check if the train.txt and test.txt files have be en created
!ls data/
```

9) Download the pre-trained ***yolov4*** weights.

```
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
```

10) Traing of custom detector

```
!./darknet detector train data/obj.data cfg/yolov4-custom.cfg yolov4.conv.137 -dont_show -map
```

11) Check mAP (mean, Average, Precision)

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

12) Testing of custom detector

a) Make change to custom config file
   change line batch to batch=1

change line subdivisions to subdivisions=1

```
%cd cfg
!sed -i 's/batch=64/batch=1/' yolov4-custom.cfg
!sed -i 's/subdivisions=16/subdivisions=1/' yolov4-custom.cfg
%cd ..
```

b) Run detector on image

```
!./darknet detector test data/obj.data cfg/yolov4-
custom.cfg /mydrive/yolov4/training/yolov4-
custom_best.weights /content/gdrive/MyDrive/Mask_Testing/test11.png -
thresh 0.3
imShow('predictions.jpg')
```