REAL TIME FACE MASK DETECTION GUIDE

George Nikitakis

Project Objective

The goal of this project is to create a real time object detection model to differentiate people that are wearing a mask from those who aren't wearing one. I will not be building this model from scratch, but will use the pretrained tensorflow model, SSD MobileNet V2 320x320, and use transfer learning to customize this model for face mask detection.

STAGE 1: Creating Our Dataset

Step 1: Download Our Dependencies and Github Project File

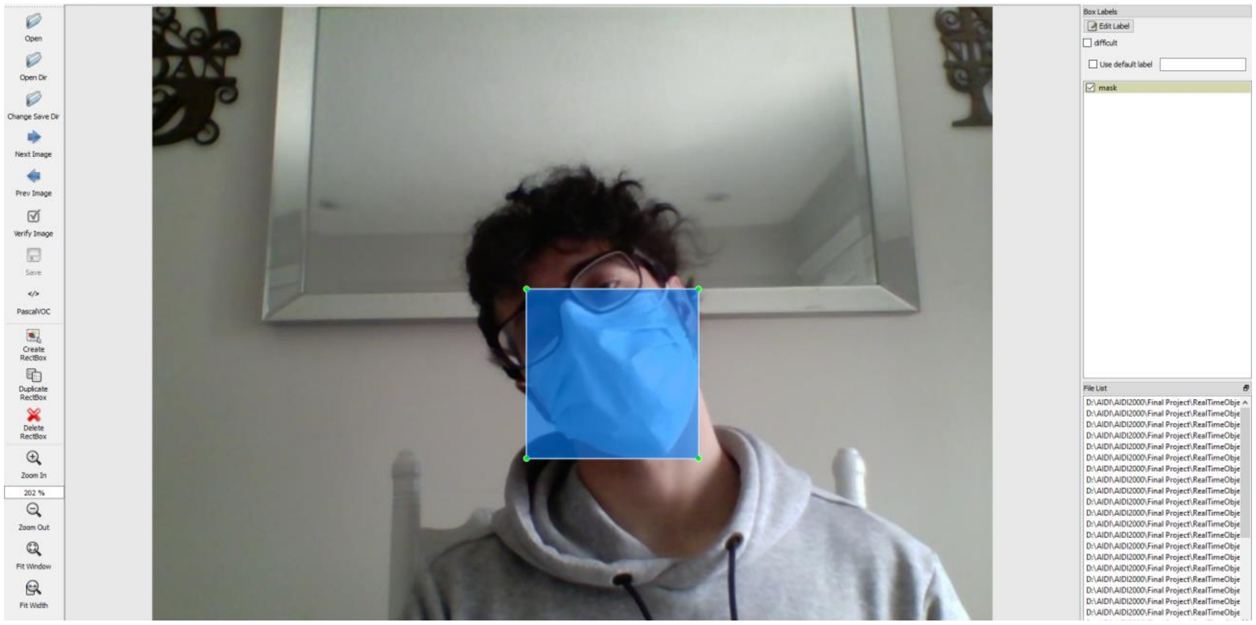
Download and Install Tensorflow 2.8.0 onto local machine.
Download and unzip the project file from the github link provided.

Step 2: Create Images Dataset

Run the "RealTimeDetection_DataCreation.ipynb" file in the main folder.
Utilizing your webcam, it will default to taking 40 images (20/label) with a 10sec timer interval between each picture. For the first 20 pictures please take with your mask on and the last 20 pictures are taken with your mask completely off. For best results, change head tilt and/or rotation between each picture to get multiple angles. Images will be saved in "imgs_path" in their respective folders.

Step 3: Annotating Images using LabelImg

Download and run the LabelImg application. Click on the "Open Dir" file on the side panel and open the folder titled "face_mask". This should load the 20 images with the face mask that was recently taken in Step 2. Make sure the "</>" icon reads PascalVOC underneath it as we will be using a pretrained model in Tensorflow and not YOLO. We will now annotate these images. Press "W" on your keyboard or click on the "Create RectBox" icon on the side panel. Draw a rectangular box around the face mask area of the image, making sure that the face mask is completely within the box boundaries. For best results the mask should span the entire width and height of the box. Once satisfied, press ctrl/cmd s to save the annotated image along with the .xml file into the current folder. After saving, click “Next Image” icon on the side to navigate to the next image. Repeat the process for all the images in the face mask folder. Once finished, complete the same process for the images in the “no_mask” folder.



Step 4: Splitting Training and Testing Dataset

After we have labeled and annotated all 40 images (20 face mask + 20 no mask), create two separate folders titled “train” and “test” within the “images” folder. Select 16 images and their corresponding .xml files from both the face_mask and no_mask folders and move them into the train folder. Finally, take the remaining images and their corresponding .xml files and place them in the test folder. Delete the empty “face_mask” and “no_mask” folders.

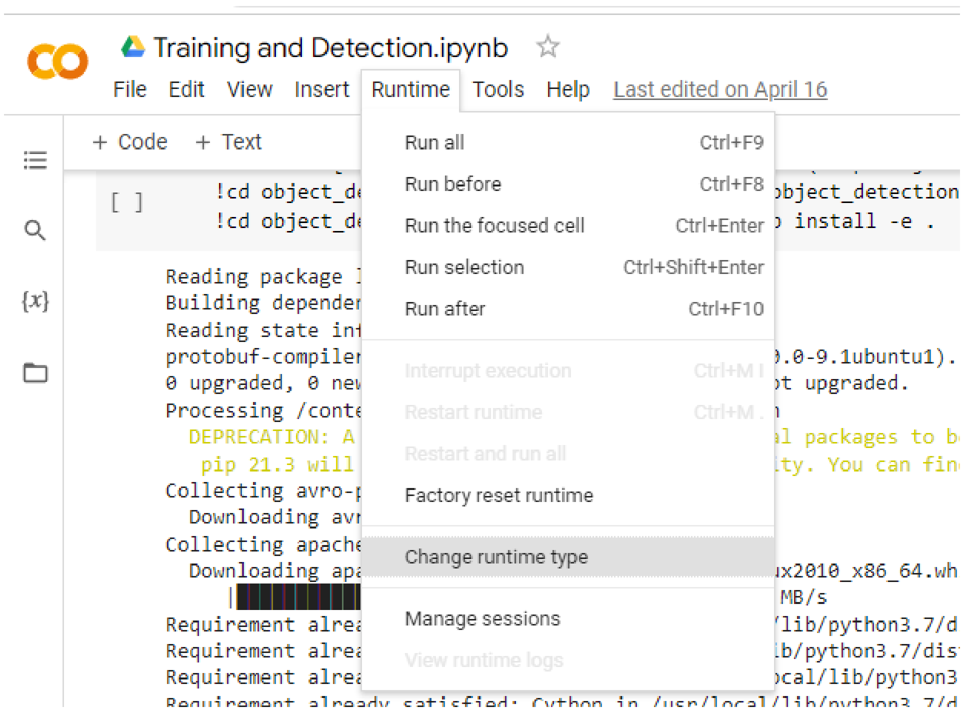
Step 5: Zip the Train and Test Folders

Run the “Compress for Colab Training” section of the code to create a zip file titled “archive.tar.gz”. Stage 1 is now complete.

STAGE 2: Training on Colab

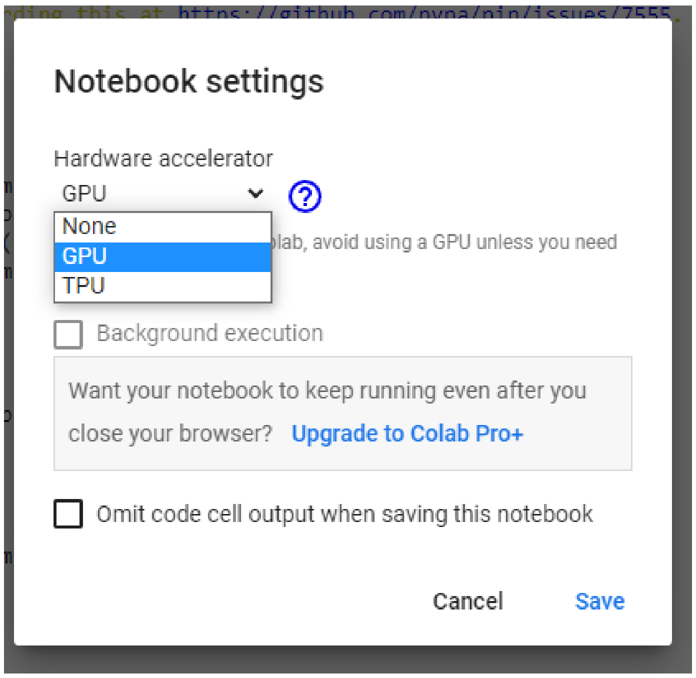
Step 1: Colab Setup

Open the “Training and Detection.ipynb” file on Google Colab. Click on the “Runtime” tab at the top of the screen. Click on “Change runtime type”. Under “Hardware Accelerator” select “GPU” and click save.



```
[ ]     !cd object_de                          object_detection
        !cd object_de                          install -e .

    Reading package l
    Building depender
    Reading state inf
    protobuf-compiler                          .0-9.1ubuntu1).
    0 upgraded, 0 new                          ot upgraded.
    Processing /conte
      DEPRECATION: A                           al packages to b
       pip 21.3 will                           ity. You can fin
    Collecting avro-p
      Downloading avr
    Collecting apache
      Downloading apa                          ux2010_x86_64.wh
                                               MB/s
    Requirement alrea                          /lib/python3.7/d
    Requirement alrea                          ib/python3.7/dis
    Requirement alrea                          ocal/lib/python3
    Requirement already satisfied: Cython in /usr/local/lib/python3.7/d
```



Hardware accelerator
GPU

…lab, avoid using a GPU unless you need

Background execution

Want your notebook to keep running even after you close your browser? Upgrade to Colab Pro+

Omit code cell output when saving this notebook

Step 2: Download and Install TF Pretrained Models

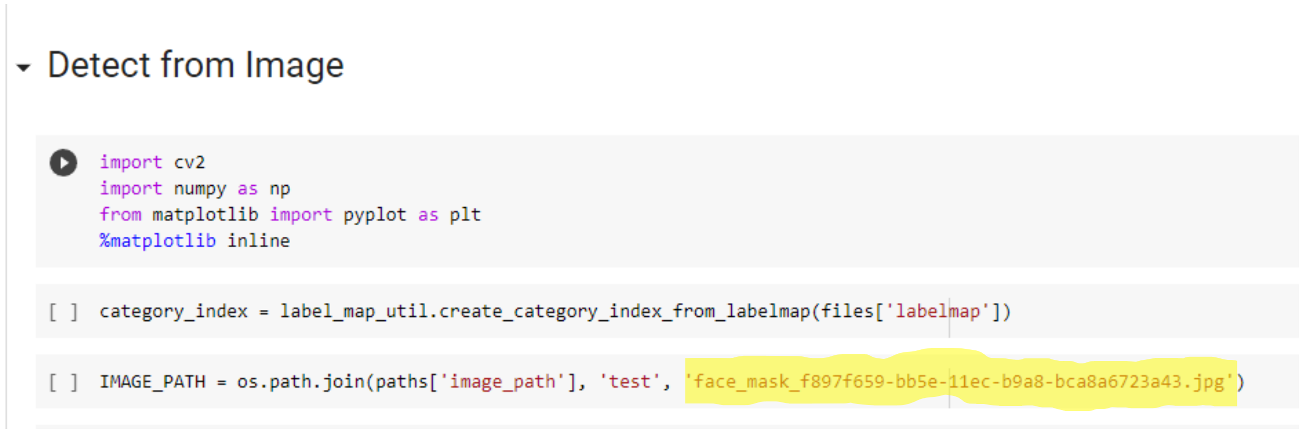
Run the “Setup” and “Download and Install TF Pretrained Models” sections of the code. New folders have now been created and stored temporarily in your cloud vm environment.

Step 3: Import and extract archive.tar.gz file

Navigate to the “images” folder in the folders created in step 2. (location = object_detection/workspace/images). Upload the archive zip file created in stage 1, step 5. Run sections “Create Label Map” and “Create TF Records”.

Step 4: Train and Test Model

Now we are ready to train and test our model. Run sections “Copy Model Config to Training Folder” to but not including “Detect from Image”. This part will take a while to train, but once all sections have finished running you will end up with a working model. To test this model on an image, copy and paste the image file name and paste it in place of the last string argument in the “IMAGE_PATH” cell block. Remember to keep the image format (.jpg/.png/etc…)



Detect from Image

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

```
[ ] category_index = label_map_util.create_category_index_from_labelmap(files['labelmap'])
```

```
[ ] IMAGE_PATH = os.path.join(paths['image_path'], 'test', 'face_mask_f897f659-bb5e-11ec-b9a8-bca8a6723a43.jpg')
```

Step 5: Real Time Detection

Access the saved training checkpoint files from the temporary files on the side. All the checkpoint files will be found in the following file path location (object_detection/workspace/models/my_ssd_modnet). Download all files with the name “ckpt” as well as the file named “checkpoint” onto your local machine.

Open “Real_Time_Detection.ipynb” file in your local machine (I used VS Code where I have installed all requirements listed in the “requirement.txt” file). Run code sections up until section “Load Train Model Checkpoint”. Before running the final sections save the downloaded files into the folder “my_ssd_mobnet” found in the following file path: Tensorflow/workspace/models. After all the files are placed in this folder run the final two sections of code.

Step 6: Enjoy!