***Object detection Assignments***

1. Hand-Sign detection:

* In hand sign detection we have to detect the sign images. So our problem is basically object detection. Our motive is detect hand signs which are ['thumbsup', 'thumbsdown', 'livelong']

Start with "Image Collection.ipynb" file:

In this file our main goal is to capture images from webcam or get the data from what ever you get clear images. So to get clear images I used mobile camera and then reduced its size to up to

400KB. If we use large file this cause resource error and the RAM got exhausted. So better is try to use small sizes of Image. If we want to use the webcam for that the code is written here

'''for label in labels:

    cap = cv2.VideoCapture(0)

    print('Collecting images for {}'.format(label))

    time.sleep(5)

    for imgnum in range(number\_imgs):

        print('Collecting image {}'.format(imgnum))

        ret, frame = cap.read()

        imgname = os.path.join(IMAGES\_PATH,label,label+'.'+'{}.jpg'.format(str(uuid.uuid1())))

        cv2.imwrite(imgname, frame)

        cv2.imshow('frame', frame)

        time.sleep(2)

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

cap.release()

cv2.destroyAllWindows()'''

And then we have to label the part of image we have to detect and for that we have to GIT clone the repository.

URL for github is "https://github.com/tzutalin"

And then bifurcates the image as well as labels file in the train and test folder.

Now in second file"Training and Detection.ipynb":

Our main code is here. First of all we need GTX nvidia graphic card(GPU) or TPU for the training model as normal CPU does not able to run the code properly and due to this we must need graphic cards.

Now for training our model we need the CUDA and CuDNN which are compatible to each other as well as Tensorflow. So checking for compatibility "https://www.tensorflow.org/install/source\_windows" one

can the given URL as any version of Tensorflow does not matches with CUDA it gives error while training.

We have to clone "https://github.com/tensorflow/models" this URL for importing image detection module in your environment.

Later we have to install the protocolbuffers using this link "https://github.com/protocolbuffers/protobuf/releases/download/v3.15.6/protoc-3.15.6-win64.zip"

The below code snippet is to check if your installation of object detection library is fine, if not then we have to install all libraries as it is needed.

If builders not found in protobuf library then you have to first uninstall protobuf. Later install protobuf and again install protobuf version==3.20. If you not this method you got error of compatibility.

"VERIFICATION\_SCRIPT = os.path.join(paths['APIMODEL\_PATH'], 'research', 'object\_detection', 'builders', 'model\_builder\_tf2\_test.py')

!python {VERIFICATION\_SCRIPT}"

And then Downloaded pre-trained model from tensorflow zoo. The URL is "https://github.com/tensorflow/models/blob/master/research/object\_detection/g3doc/tf1\_detection\_zoo.md"

I used "ssd\_mobilenet\_v2\_fpnlite\_320x320\_coco17\_tpu-8.tar.gz" for our data. It depends on accuracy as well as the speed how fast our model trains.

And then follow the code to generate the TF records as well as to train model.

Reference video link: "https://www.youtube.com/watch?v=yqkISICHH-U&t=5130s"

1. License plate detection:

* Link to problem statement: <https://www.kaggle.com/datasets/andrewmvd/car-plate-detection>
* Trained\_model.zip should be unzipped in “/content/Tensorflow/workspace/models/my\_ssd\_mobnet”. It contains the already trained model using ssd\_mobnet using 10000 train steps.