AGEING SIGN BATCH 9

User Manual

ERROR GUIDE ON PAGENO. 9,10

REFER FOR TRAIN AND TEST DATASET

https://drive.google.com/drive/folders/18TRMxGRYNf2qGe2Aeg_yYh76HmcisrSf?usp=sharing

OPEN COMMAND PROMPT AND USE CD FUNCTION AND SPECIFY PATH WHERE THE PROJECT IS TO BE DONE

Step 1. Creating a new virtual environment

```
python -m venv verzeo
```

Step 2 . Activate your virtual environment

```
.\verzeo\Scripts\activate
```

```
(verzeo) C:\
```

Something like this interface will be seen

After this create a folder name Tensorflow where the virtual environment is created

Be sensitive in case keep the T capital as in the code it is used

Create a folder structure like this

Make two folders inside images train and test and paste the images with xml

```
Tensorflow    - ---  workspace ---- images --- train
                                                  --  test
```

INSTALLING ALL DEPENDENCIES –

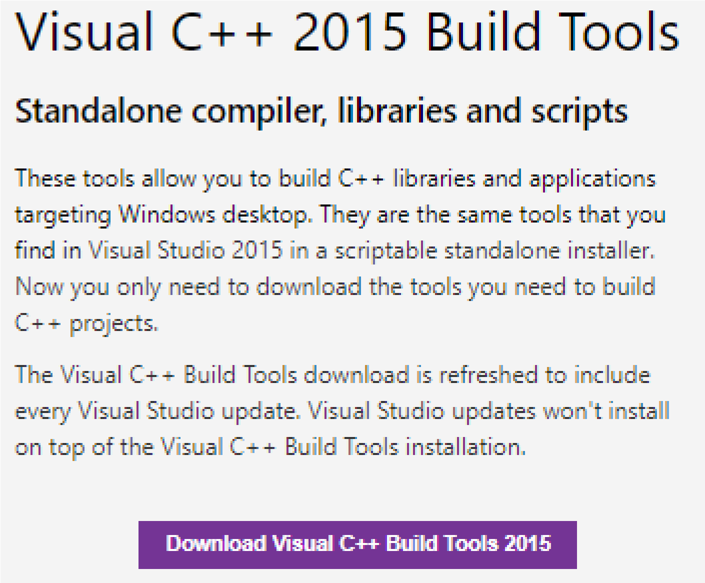
Everything must be done in verzeo environment

```
python -m pip install --upgrade pip
pip install ipykernel
python -m ipykernel install --user --name=verzeo
```

here name = verzeo defines you kernel name

```
pip install tensorflow=2.5.0
pip install tensorflow-gpu
```

INSTALL VISUAL STUDIO BUILD TOOL



Visual C++ 2015 Build Tools

Standalone compiler, libraries and scripts

These tools allow you to build C++ libraries and applications targeting Windows desktop. They are the same tools that you find in Visual Studio 2015 in a scriptable standalone installer. Now you only need to download the tools you need to build C++ projects.

The Visual C++ Build Tools download is refreshed to include every Visual Studio update. Visual Studio updates won't install on top of the Visual C++ Build Tools installation.

If this step does not work, manually install the community edition of Visual Studio.

https://visualstudio.microsoft.com/downloads/

Don't forget to double-check the download and installation of C++ libraries.

After then, all of the installations will be completed.

Using the virtual environment to access the Jupyter notebook

And check the kernel name is same as created if not then change it

Executing the step 1 of the code :

After running all codes in step 1 of the ipynb file, this will be the folder structure created after running code -

Tensorflow -- models

protoc

scripts

workspace - --- annotations

-- images

-- models

--- pre-trained-models

Executing the step 2 of the code

After running all codes in step 2 of the ipynb file, this will be the folder structure.

```
TensorFlow/
    └─ models/
        ├─ community/
        ├─ official/
        ├─ orbit/
        ├─ research/
        └── ...
```

THIS STEP WILL DOWNLOAD PRETRAINED MODELS FROM TENSORFLOW ZOO AND WILL SETUP EVERYTHING NEEDED

```
VERIFICATION_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'builders', 'model_builder_tf2_test.py') # Verify Installation !python {VERIFICATION_SCRIPT}
```

Your last line of output must be this

OK (skipped=1)

If you encounter any difficulties while executing the verification script, go back and review the guide from the beginning.

If the previous step was performed successfully, continue to the next step. It means you've completed the setup of all the components required for object detection.

Executing the step 3 of the code

pbtxt file will be created in the annotations folder after running label map code (step 3 code).

Executing the step 4 of the code

The fourth

This code will create **tf records.**

To see if they have been made or not, simply go to

Tensorflow\workspace\annotations

If you see train.record and test.record files in this location, you're ready to move on to the next step.

Executing the 5th and 6th step

The pipline.config file will be copied to

Tensorflow\workspace\models\ssd mobilenet v2 fpnlite 640x640

The code, on the other hand, will setup the required settings.

7th Step - Training the model

This stage will take a long time to train the model, so don't get in the way.

```
print(command)
```

following this command output –

Similar outcome will be displayed.

```
python Tensorflow\models\research\object_detection\model_main_tf2.py --model_dir=Tensorflow\workspace\models\my_ssd_mobnet  --pipeline_config_path=Tensorflow\workspace\models\my_ssd_mobnet\pipeline.config --num_train_steps=3000
```

Simply open a command prompt, enable the virtual environment, paste the command, and the training will begin.

EITHER RUN THE BELOW !{COMMAND} OR JUST GO TO COMMAND PROMPT ACITIVATE THE VIRTUAL ENV. AND PASTE THE COPIED OUTPUT AND RUN

NOTE: THIS WILL START TRAINING WHILE IT MAY TAKE MORE TIME TO COMPLETE IT DEPENDING ON YOUR PC SPECS

Refer to the error guide if you encounter any problems.

8th step

While the following step is required for assessing the model, it is an optional step that you can skip right to the next stage. OR follow the code comments for step 8

9th Step -

Before running this code

navigate to this directory

Tensorflow\TFODCourse\Tensorflow\workspace\models\ssd_mobilenet_v2_fpnlite_640x640

When you enter, you'll notice

Files such as ckpt-4.index and others , just check the highest checkpoint and edit the no. in this line

```
ckpt.restore(os.path.join(paths['CHECKPOINT_PATH'], 'ckpt-4')).expect_partial()
```

If we have two folders named ckpt-4.index and ckpt-5.index, we will simply select the one with the greatest number and update it in code as follows.

```
ckpt.restore(os.path.join(paths['CHECKPOINT_PATH'], 'ckpt-5')).expect_partial()
```

After that, follow the steps –

10 th Step -

Finally, we've arrived at the final step: images and video detection.

```
IMAGE_PATH = os.path.join(paths['IMAGE_PATH'], 'test', image.jpg')
```

Replace image.jpg with whichever picture you wish to refer to from the test.

You will receive a result if you run all of the programmes in each stage.

Next step -

When you run this code, it will open your webcam and try to identify symptoms of ageing.

NEXT PAGE IS FOR ERROR GUIDE

ERROR GUIDE

Error: No module named ‘xxxxxx’
Solution: Install that module

```
!pip install xxxxxx
```

Example:
Error: No module named typeguard
Solution: pip install typeguard # note the name of the module will not always equal the package name

Error: AttributeError: module 'sip' has no attribute 'setapi'
Solution: Downgrade matplotlib to version 3.2 by running the following command

```
!pip install matplotlib==3.2
```

Error: ValueError: numpy.ndarray size changed, may indicate binary incompatibility. Expected 88 from C header, got 80 from PyObject
Solution: Reinstall pycocotools

```
Pip uninstall pycocotools -y
```

```
Pip install pycocotools
```

Error: ValueError: 'images' must have either 3 or 4 dimensions.
Solution: Restart your jupyter notebook as the Webcam is unavailable. If using images, this normally means your image name and path is incorrect.

Error:error: (-2:Unspecified error) The function is not implemented. Rebuild the library with Windows, GTK+ 2.x or Cocoa support. If you are on Ubuntu or Debian, install libgtk2.0-dev and pkg-config, then re-run

cmake or configure script in function 'cvDestroyAllWindows'
Solution: Reinstall opencv and uninstall opencv-headless

```
pip uninstall opencv-python-headless -y
```

```
pip install opencv-python --upgrade
```

Error: When running GenerateTFRecords script you receive an error like the following: File "Tensorflow\scripts\generate_tfrecord.py", line 132, in create_tf_example classes.append(class_text_to_int(row['class'])) File "Tensorflow\scripts\generate_tfrecord.py", line 101, in class_text_to_int return label_map_dict[row_label] KeyError: 'ThumbsDown' # YOUR LABEL HERE
Solution: This is likely because you mismatches between your annotations and your labelmap. Ensure that the label names from your annotations match the label map exactly, note it is case sensitive.

Error: When running training script from the command line, you get a No module error. e.g. ModuleNotFoundError: No module named 'cv2'
Solution: Remember you need to activate your environment at the command line in order to leverage all the packages you have installed in it.

ERROR – GIT CLONING ERROR GO MANUALLY IN CODE CHECK THE URL OR GITHUB AND JUST DOWNLOAD THE ZIP AND EXTRACT WITH RESPECT TO THE FOLDER STRUCTURE AND AFTER ANALYSING CODE .