Tutorial by Nooh Ayub

(Faster RCNN for Face Mask Detection Using Pretrained Model)

Objectives:

- Understand Pre trained model
- train this pretrained model for your custom dataset
- Evaluating the model on face mask test dataset

Here's a step by step breakdown:

Step 1 — Create a directory named "Faster_RCNN_pytorch" and inside this directory create a virtual environment:

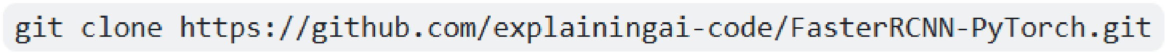
(a) create a conda environment with python 3.10.15:

```
conda create --name faster_rcnn_pytorch python=3.10.15
```

(b) now activate this environment:

```
conda activate faster_rcnn_pytorch
```

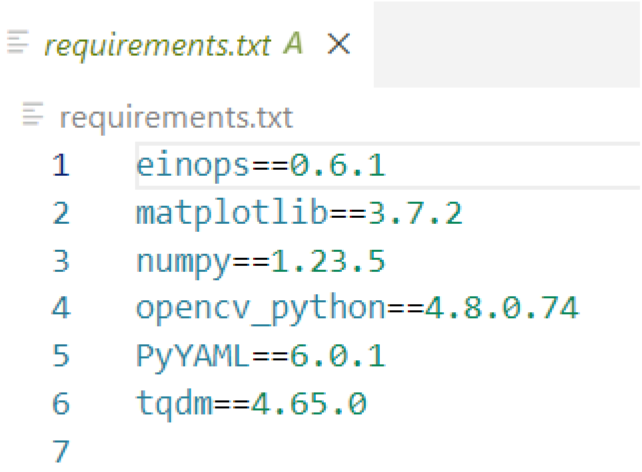
Step 2 — Now clone the following github repository and go inside that cloned folder using the subsequent command:



```
git clone https://github.com/explainingai-code/FasterRCNN-PyTorch.git
```

```
cd FasterRCNN-PyTorch
```

Step 3 — Next make some changes to the requirements.txt file before installing it, it should like like this:



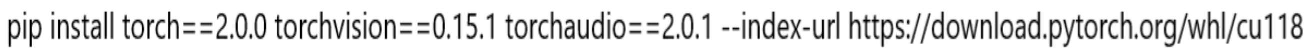
```
einops==0.6.1
matplotlib==3.7.2
numpy==1.23.5
opencv_python==4.8.0.74
PyYAML==6.0.1
tqdm==4.65.0

```

Step 4— now install all the dependencies in the requirements.txt file:

```
pip install -r requirements.txt
```

Step 5— Next install pytorch, torchvision and torchaudio with cuda support from the following command:

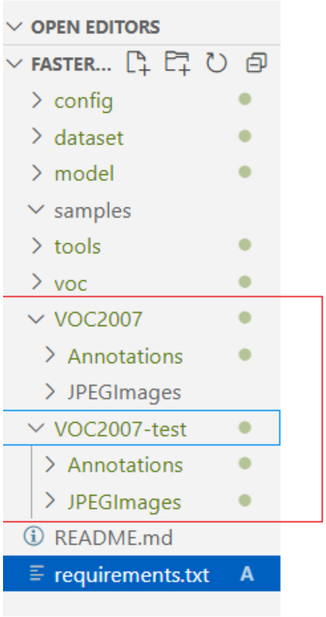
(a) We have used cuda 11.8 support for pytorch



```
pip install torch==2.0.0 torchvision==0.15.1 torchaudio==2.0.1 --index-url https://download.pytorch.org/whl/cu118
```

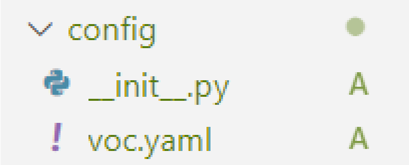
Step 6—Next make sure your dataset follows the following tree structure within the directory:

(a) We have used PASCAL Voc (2007) dataset format:



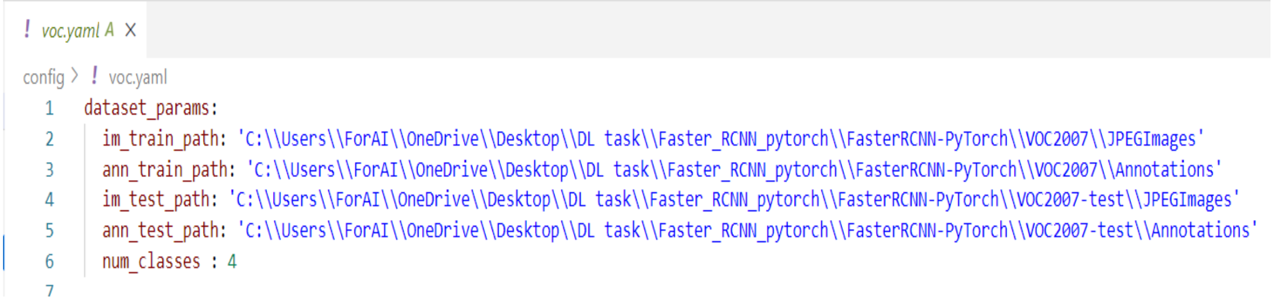
```
OPEN EDITORS
FASTER...
  config
  dataset
  model
  samples
  tools
  voc
  VOC2007
    Annotations
    JPEGImages
  VOC2007-test
    Annotations
    JPEGImages
  README.md
  requirements.txt  A
```

Step 7— Now edit the voc.yaml file inside the config folder and specifiy the correct paths for train, test datasets:



```
config
  __init__.py   A
  voc.yaml      A
```

(a) Please make sure your paths are pointing to the correct directories otherwise you will run into errors:



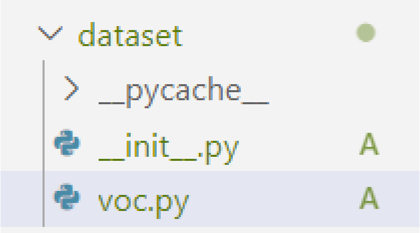
voc.yaml A ×

config > voc.yaml

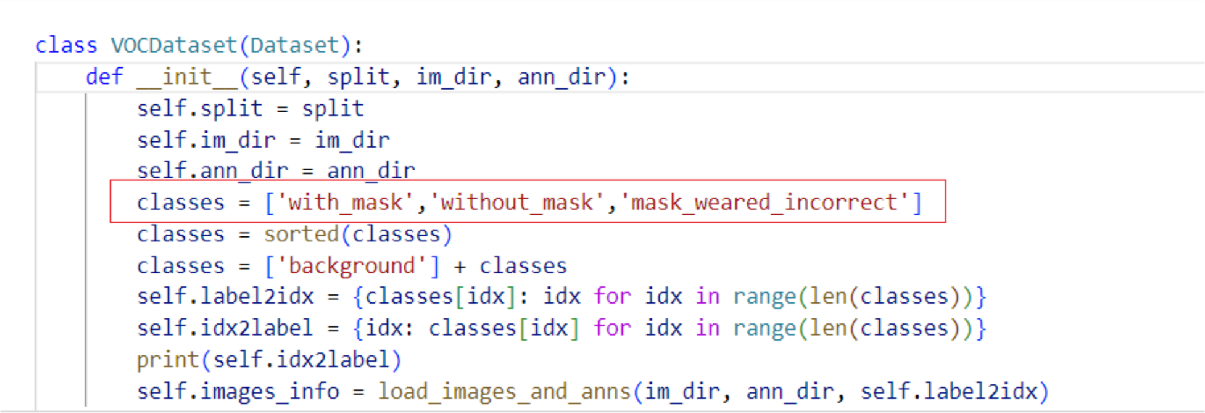
```yaml
dataset_params:
  im_train_path: 'C:\\Users\\ForAI\\OneDrive\\Desktop\\DL task\\Faster_RCNN_pytorch\\FasterRCNN-PyTorch\\VOC2007\\JPEGImages'
  ann_train_path: 'C:\\Users\\ForAI\\OneDrive\\Desktop\\DL task\\Faster_RCNN_pytorch\\FasterRCNN-PyTorch\\VOC2007\\Annotations'
  im_test_path: 'C:\\Users\\ForAI\\OneDrive\\Desktop\\DL task\\Faster_RCNN_pytorch\\FasterRCNN-PyTorch\\VOC2007-test\\JPEGImages'
  ann_test_path: 'C:\\Users\\ForAI\\OneDrive\\Desktop\\DL task\\Faster_RCNN_pytorch\\FasterRCNN-PyTorch\\VOC2007-test\\Annotations'
  num_classes : 4

```

(b) Also edit the number of classes as 4 since we have 3 classes for face mask dataset i.e (‘with_mask’, ‘without_mask’, ‘mask_weared_incorrect’) + the background class also.

Step 8— Edit the class labels in the voc.py inside the dataset folder:

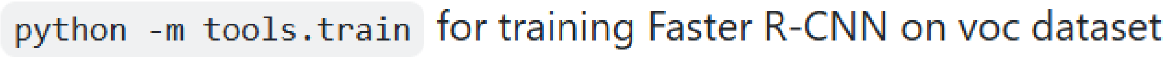


```
dataset
  __pycache__
  __init__.py    A
  voc.py         A
```



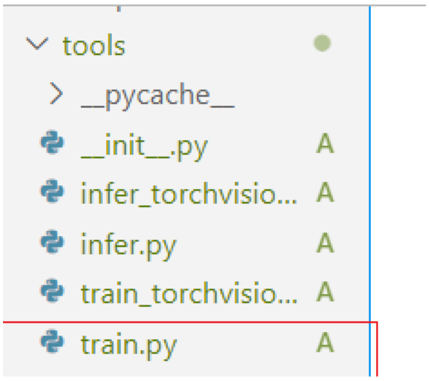
```python
class VOCDataset(Dataset):
    def __init__(self, split, im_dir, ann_dir):
        self.split = split
        self.im_dir = im_dir
        self.ann_dir = ann_dir
        classes = ['with_mask','without_mask','mask_weared_incorrect']
        classes = sorted(classes)
        classes = ['background'] + classes
        self.label2idx = {classes[idx]: idx for idx in range(len(classes))}
        self.idx2label = {idx: classes[idx] for idx in range(len(classes))}
        print(self.idx2label)
        self.images_info = load_images_and_anns(im_dir, ann_dir, self.label2idx)
```

Step 9— Next we will begin training by running the script train.py in the tools directory using the following command:



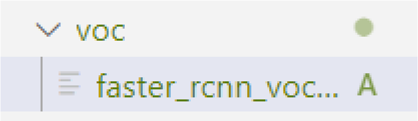
`python -m tools.train` for training Faster R-CNN on voc dataset

(a) Tools directory structure:



```
tools
  __pycache__
  __init__.py            A
  infer_torchvisio...    A
  infer.py               A
  train_torchvisio...    A
  train.py               A
```

....After the training has complete the model weights are saved in the following folder:



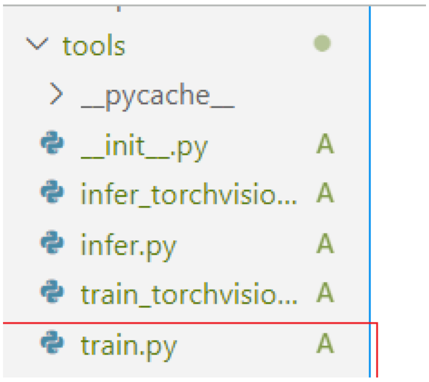
voc ●
faster_rcnn_voc... A

Step 10— Now we have to evaluate our trained model using inference with the help of the following commands:

`python -m tools.infer --evaluate False --infer_samples True` for generating inference predictions

`python -m tools.infer --evaluate True --infer_samples False` for evaluating on test dataset

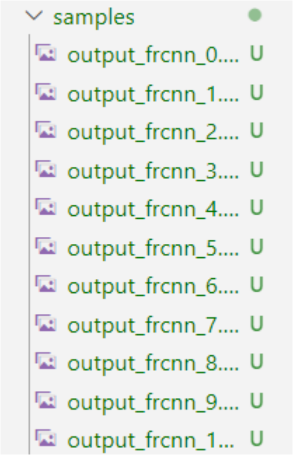
(a) Tools directory:



tools ●
__pycache__
__init__.py A
infer_torchvisio... A
infer.py A
train_torchvisio... A
train.py A

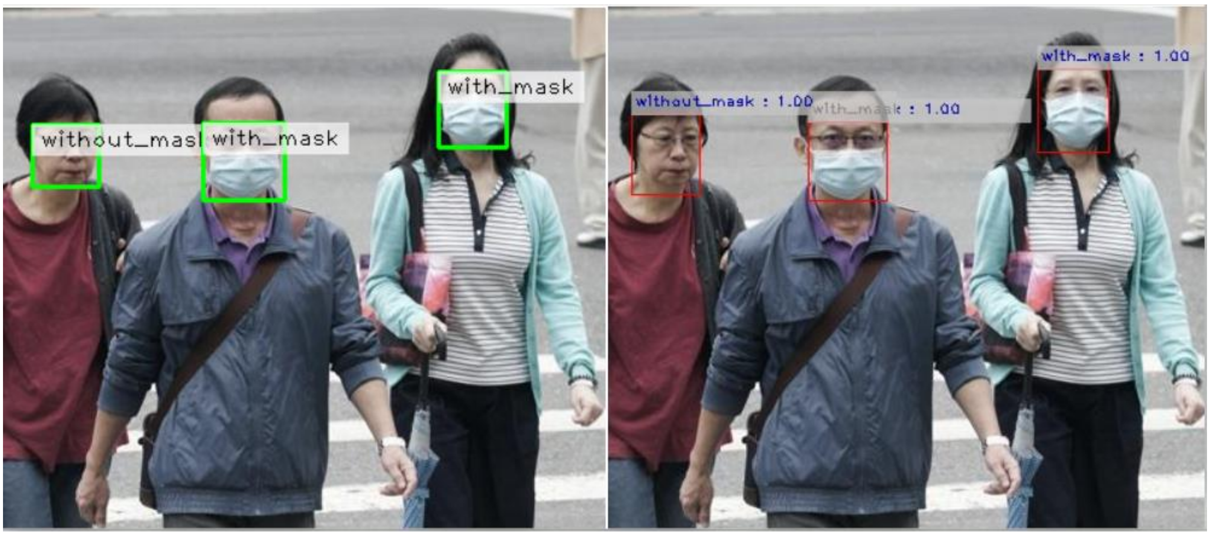
Step 11— the prediction results are saved in the samples directory where green bounding boxes are the ground truths and the red bounding boxes indicates the predictions:

(a) Directory for test predictions:

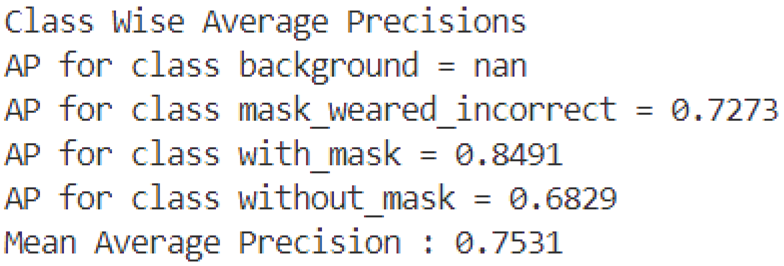


samples ●
output_frcnn_0.... U
output_frcnn_1.... U
output_frcnn_2.... U
output_frcnn_3.... U
output_frcnn_4.... U
output_frcnn_5.... U
output_frcnn_6.... U
output_frcnn_7.... U
output_frcnn_8.... U
output_frcnn_9.... U
output_frcnn_1... U

(b) Sample prediction



Step 12— mAP score after training on 30 epochs:



```
Class Wise Average Precisions
AP for class background = nan
AP for class mask_weared_incorrect = 0.7273
AP for class with_mask = 0.8491
AP for class without_mask = 0.6829
Mean Average Precision : 0.7531
```