Lab 8 – PyTorch and Deep Learning 2

Prior to commencing with the tasks, ensure that you:

1. Install the Raspberry Pi Camera Module **prior** to plugging in the Pi.
2. Synchronise the time on the Pi, either by connecting to a different network, i.e. a Hotspot, or by using: `sudo date -s 'YYYY-MM-DD hh:mm:ss`
3. Clone the Lab8 Repository from GitHub Classroom **into the ee347 folder**, using this link. Open the ee347 folder in VSCode and work from there. **Do not move the .venv folder.**
4. Commit and Push to GitHub after each task. Each task should be completed in the individual taskX.py scripts provided. Ensure model/data files are included in your .gitignore or you may have difficulties pushing to GitHub.

Tasks

1. Taking inspiration from the camera app developed in lab 6, capture 60 images of each teammate's face and save in the data folders provided. Use haarcascades to crop to the faces and resize the face crops to 64x64. Save 50 of the images to the train folder and 10 to the test folder. One person should be saved in the 0 folder, while the other in the 1 folder. Your data folder should look as follows:

```
data
├── test
│   ├── 0
│   └── 1
└── train
    ├── 0
    └── 1
```

2. Run the train script in task2.py.
3. Edit the train script to add a random horizontal flip transform to the train loop.
4. Edit the train script to add a random rotation transform (maximum 10 degrees) to the train loop.
5. Train using CrossEntropyLoss and NLLLoss. Tabulate the results and explain them to the TAs.
6. Adjust the learning rate and the batch size of the train script to improve training performance. Explain your choice to the TAs.
7. Adjust the training script to save the confusion matrix after each epoch. Overwrite the image to simulate a 'live' confusion matrix. Use sklearn to create the confusion matrix.
8. Using the best combination of the elements you've adjusted above, train until you are satisfied with the performance and save the model.
9. Deploy to an application that captures an image, crops to the face and classifies which teammate is present.
10. Adjust the application to run on a live video stream. Draw bounding boxes and show the detected names on the bounding boxes.