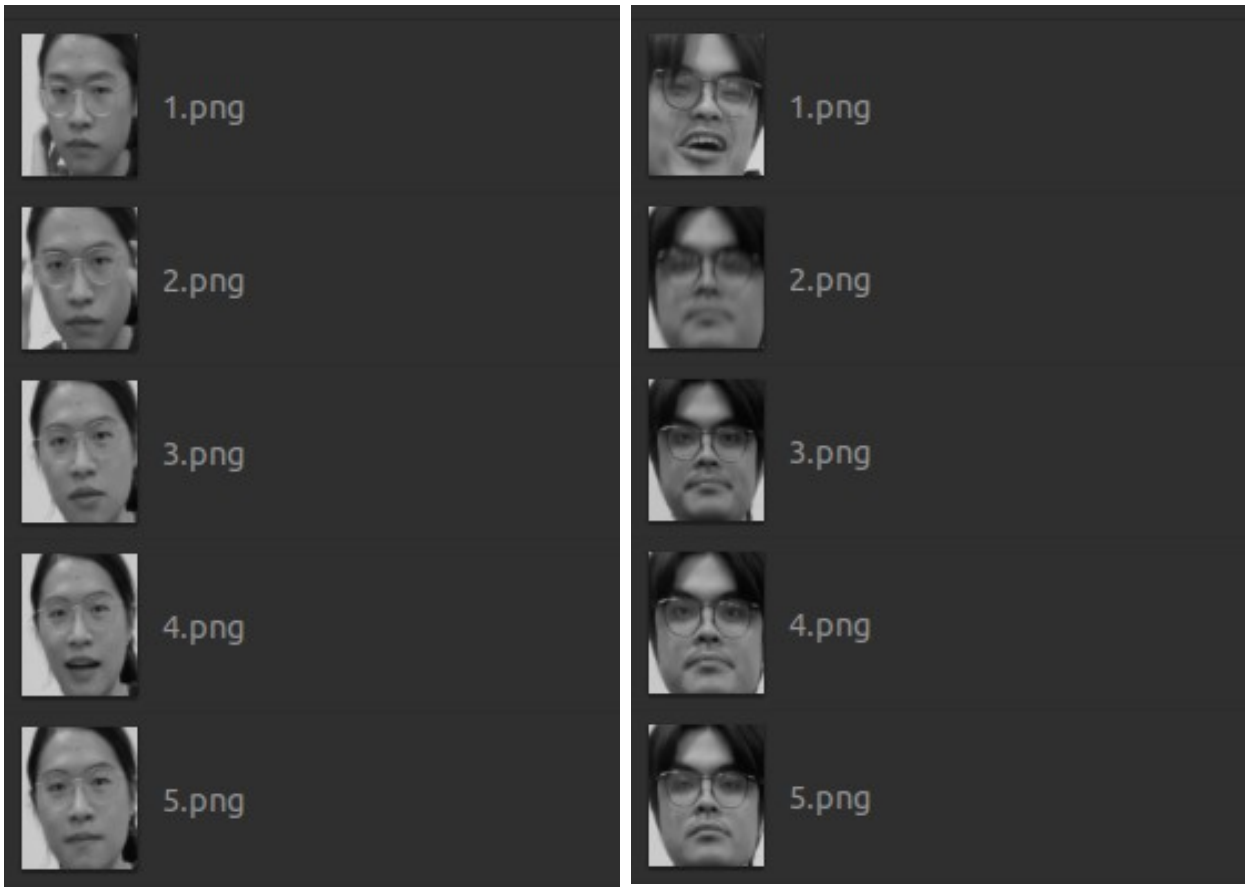
Final Project
第 11 組

313552041_洪日昇
313552042_陳品翰

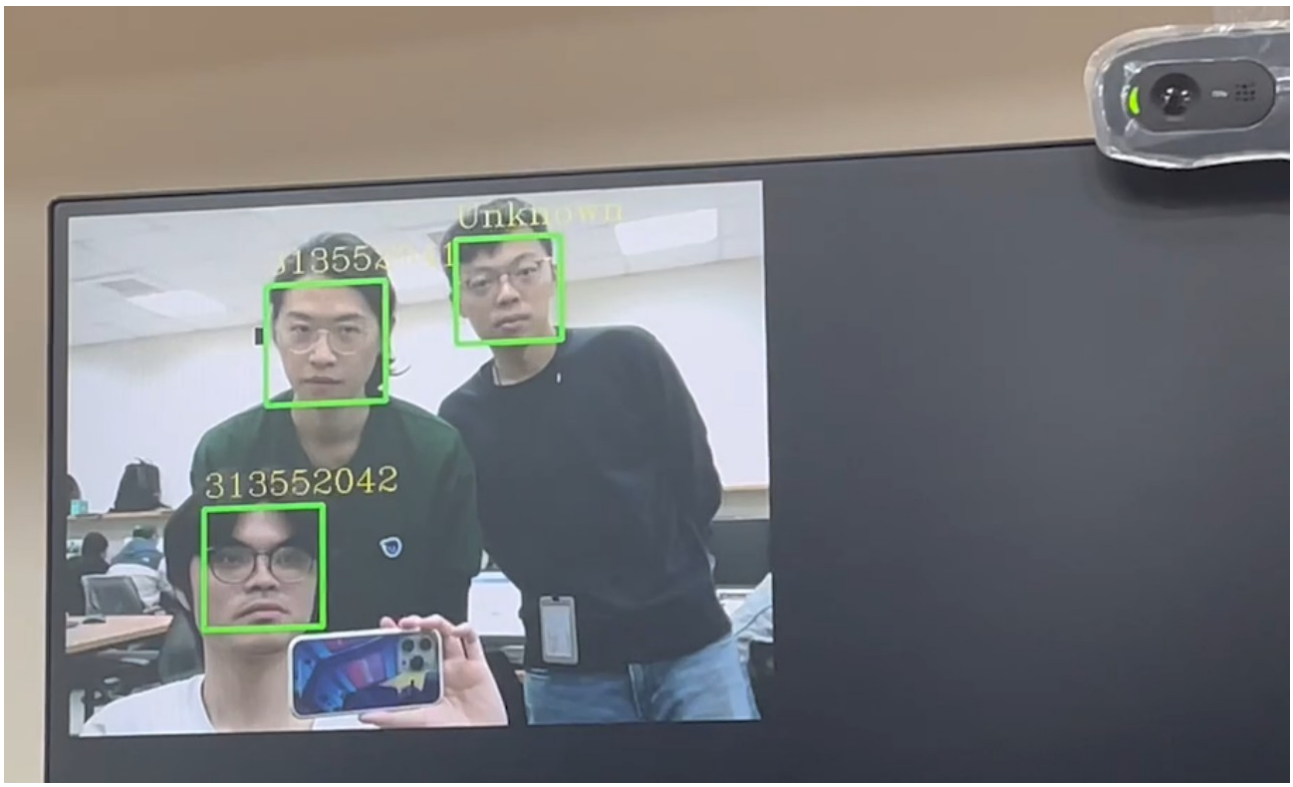
#Face Detection

在這次的專案中，我們實現了即時人臉識別的功能，利用 E9V3 實做人臉辨識，分辨出三個人的臉，兩個為組員，一個為 unknown。為了完成這項任務，我們採用了 OpenCV 函式庫中的 FisherFaceRecognizer 來進行模型的訓練和推論，我們主要參考[1]的作法，逐步完善整體流程。

在訓練部分，我們首先準備了一組人臉資料集。每個人拍攝了約 30 張臉部照片，這些圖像被用於模型的訓練。我們利用 OpenCV 中的 FisherFaceRecognizer 模組來建立和訓練人臉識別模型。過程中，我們呼叫了 FisherFaceRecognizer::create()建立模型，並使用 model->train(images, labels)方法完成訓練，其中 images 是已處理的臉部影像數據，labels 則是對應的分類標籤。



inference 時，透過網路攝影機取得即時影像，利用 OpenCV 的 cascade.detectMultiScale()函式偵測影像中所有不同尺寸的人臉區域，接著將檢測到的臉部影像傳遞至訓練好的模型進行預測。模型會輸出預測的類別標籤和信心分數，並根據這些結果在影像中標註對應的人臉名稱，標示為學號或 Unknown。



原本由 cmake-gui 編譯的 libopencv.so 檔案中並不包含 cascade.detectMultiScale()和 FisherFaceRecognizer 這些必要的函式。因此，我們需要額外進行 OpenCV 的交叉編譯，來生成完整的函式庫以支持這些功能。

下載 OpenCV Contrib 3.4.7 版本，並在 cmake-gui 中進行一些額外的配置，包括設置 OPENCV_EXTRA_MODULES_PATH 路徑、啟用 BUILD_opencv_face 模組、勾選 OPENCV_ENABLE_NONFREE，並取消 WITH_QT。最後，使用 CMake 進行交叉編譯，生成了適合在 e9v3 開發板上運行的 libopencv.so 函式庫。

git clone https://github.com/opencv/opencv_contrib

cd opencv_contrib

git checkout<3.4.7>

sudo cmake-gui

OPENCV_EXTRA_MODULES_PATH 的路徑，該路徑需要指向 opencv_contrib 資料夾中的 modules 資料夾，例如 /path/to/opencv_contrib/modules

新增

name：BUILD_opencv_face

type: bool

value: on

勾選
OPENCV_ENABLE_NONFREE

取消勾選
WITH_QT

make
sudo make install

 # Mask Detection
利用 E9V3 實做影像辨識，要求精度，因此使用 yolo v5 做測試，利用 lab3 已經做過的 Yolov5，使用[2]的 cpp code 以及使用[2]的 data set 與 model。

git clone https://github.com/doleron/yolov5-opencv-cpp-python.git
cd yolov5-opencv-cpp-python
g++ -O3 cpp/yolo.cpp -o yolo_example `pkg-config --cflags --libs opencv4`
./yolo_example

[3]有提供可以自己 train 更重型的 model，根據[3]的 Readme 教學。
git clone https://github.com/spacewalk01/face-mask-detection
cd face-mask-detection

Install yolov5
git clone https://github.com/ultralytics/yolov5
cd yolov5
pip install -r requirements.txt

Download Face-Mask dataset from Kaggle and copy it into datasets folder.

Execute the following command to automatically unzip and convert the data into the YOLO format and split it into train and valid sets. The split ratio was set to 80/20%.

cd ..
python prepare.py

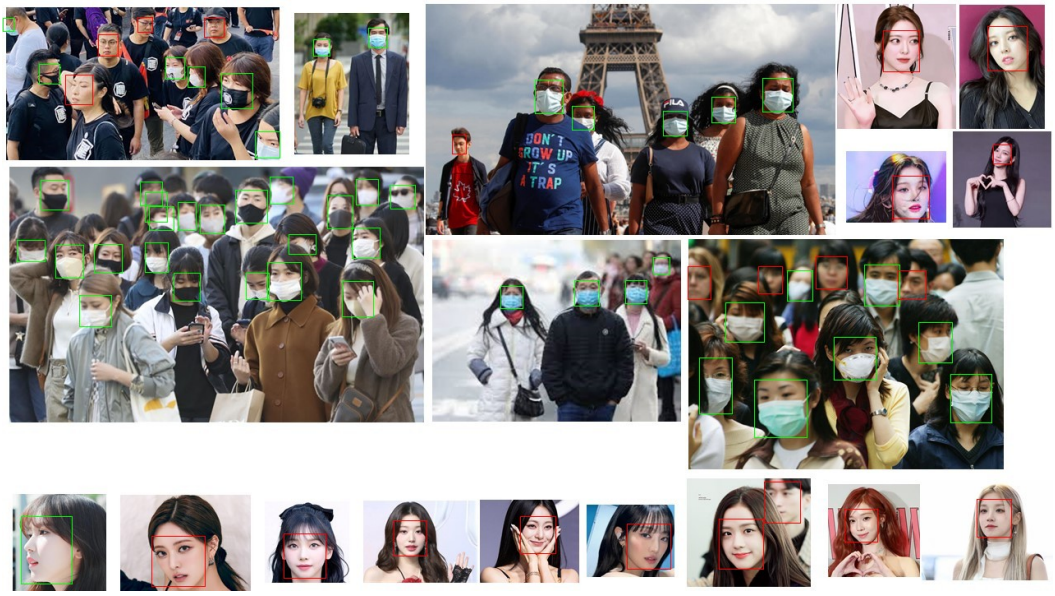
cd yolov5
python train.py --img 640 --batch 16 --epochs 100 --data ../mask_config.yaml --weights yolov5s.pt --workers 0

Yolov5 對應 opencv4.5.4 必須重新 compile，若遇到問題需要把出問題的部份註解即可。

重新 compile 按照之前課堂的 pdf 做並且記得要加入 png 以及影像辨識的該有的部份。

Yolo v5 有用到 openmp 必須加入 libgomp.so.1，這個可以在 arm-linux-gnueabihf 已經編譯過的電腦資料夾裡面去找，直接像是 libopencv_world.so.4.5，放在 sd 卡中的資料夾即可。

口罩偵測結果如下，共 68 個正確的圖片，為全班最多的偵測量，使用 yolov5x 的 pretrained model 自行訓練口罩照片而得到的。



Compilation
arm-linux-gnueabihf-g++ -O3 cpp/final.cpp -o final -I /opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/include/ -I /usr/local/arm-opencv4.5/install/include/opencv4/ -L /usr/local/arm-opencv4.5/install/lib/ -Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/arm-linux-gnueabihf/libc/lib/ -Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/qt5.5/rootfs_imx6q_V3_qt5.5_env/lib/ -Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/qt5.5/rootfs_imx6q_V3_qt5.5_env/qt5.5_env/lib/ -Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/qt5.5/rootfs_imx6q_V3_qt5.5_env/usr/lib/ -lpthread -lopencv_world -std=c++11

# Excecution
LD_LIBRARY_PATH=. ./your_code

＃Reference

[1] https://www.cnblogs.com/tony-yang-flutter/p/16246328.html

[2] https://github.com/doleron/yolov5-opencv-cpp-python

[3] https://github.com/spacewalk01/yolov5-face-mask-detection