

# Embedded System Design Lab5 Document

## 第八組

312553040 郭晉維 312551172 陳昱凱

### 1. Real-time Facial Recognition:

在即時人臉識別的部分，我們使用了 OpenCV 函式庫中的 FisherFaceRecognize( ) 函式來完成模型的 training 跟 inference，我們主要參考了 <https://www.cnblogs.com/tony-yang-flutter/p/16246328.html> 網站，再把人臉辨識模型跟 lab3 的即時物件偵測的程式碼做結合，即可在 E9V3 開發板上完成即時人臉辨識。

在 training 部分，首先要準備訓練用的人臉資料集，我們一人拍了約 80 張臉部的照片供模型進行訓練，如下面兩張圖所示，接下來執行 `Ptr<FisherFaceRecognizer> model = FisherFaceRecognizer::create();`  
`model->train(images, labels);`  
建立人臉識別的 model 跟進行 training。

在 inference 部分，利用 webcam 取得即時影像後，利用 OpenCV 的 `cascade.detectMultiScale( )` 函式先框出畫面中所有不同尺寸大小的人臉，接著使用 `model->predict(image, predictedLabel, confidence)` 來完成預測並標示出對應的 class (312553040, 312551172, Unknown)，即可順利完成即時人臉識別的任務。

由於原本 lab2 使用 `cmake-gui` 產生的 `libopencv.so` 檔中並沒有包含 `cascade.detectMultiScale( )` 跟 `FisherFaceRecognize( )` 兩個函式，因此必須另外 cross-compile 一個新的 `libopencv.so` 檔，下面是一些要額外做的步驟：

1. 下載 `opencv_contrib_3.4.7`
2. 在 `cmake-gui` 中將 `OPENCV_EXTRA_MODULES_PATH` 路徑設為 `/path/to/opencv_contrib`
3. 勾選 `BUILD_opencv_face`
4. 勾選 `OPENCV_ENABLE_NONFREE`
5. 取消勾選 `WITH_QT`

然後輸入下面的 command 即可完成 cross-compile 並生成可以在 E9V3 開發板上執行的檔案。

```
arm-linux-gnueabihf-g++ -std=c++11 lab5-1.cpp -o lab5-1 \
-I /opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/include/ \
```

```
-I /usr/local/arm-opencv-face/install/include/ -L /usr/local/arm-opencv-face/install/lib/ \
-Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabi/lib/arm-
linux-gnueabi/libc/lib/ \
-Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-
gnueabi/qt5.5/rootfs_imx6q_V3_qt5.5_env/lib/ \
-Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-
gnueabi/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-
gnueabi/qt5.5/rootfs_imx6q_V3_qt5.5_env/usr/lib/ \
-lpthread -lopencv_world
```

## 2. Facial recognition with masks:

### 1. Install yolov5-face-mask-detection:

```
git clone https://github.com/spacewalk01/face-mask-detection
cd face-mask-detection
```

### 2. Install yolov5:

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

因 yolov5 需要 OpenCV 4.5.4+的版本，因此要重新編譯 opencv，以上課提供的 pdf 為主，再另外勾選其他相關選項完成 opencv 準備。

參考 yolov5-opencv-cpp-python.git 來進行 cpp 的 cross compile

Install yolov5-opencv-cpp-python:

```
3. 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`
```

使用 ldd 檢查執行檔所需.so 檔，並將 libgomp.so.1, libopencv\_world.so.4.5 放入與執行檔同層位置。

執行此 crosscompile 指令即可生成可在 E9V3 上執行的檔案

```
arm-linux-gnueabi-g++ -O3 cpp/lab5-2.cpp -o lab5-2 -I /opt/EmbedSky/gcc-linaro-5.3-
2016.02-x86_64_arm-linux-gnueabi/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-gnueabi/lib/arm-linux-
```

```
gnueabihf/libc/lib/ -Wl,-rpath-link=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-  
linux-gnueabi/qt5.5/rootfs_imx6q_V3_qt5.5_env/lib/ -Wl,-rpath-link=/opt/EmbedSky/gcc-  
linaro-5.3-2016.02-x86_64_arm-linux-  
gnueabi/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-  
gnueabi/qt5.5/rootfs_imx6q_V3_qt5.5_env/usr/lib/ -lpthread -lopencv_world -std=c++11
```

Reference:

<https://www.cnblogs.com/tony-yang-flutter/p/16246328.html>

<https://github.com/spacewalk01/yolov5-face-mask-detection>

<https://github.com/ultralytics/yolov5>

<https://github.com/doleron/yolov5-opencv-cpp-python?tab=readme-ov-file>