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/oprncv\_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-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

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-gnueabihf-g++ -O3 cpp/lab5-2.cpp -o lab5-2 -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

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>