YOLO及FACENE 即時多人人臉識

系統說明文件

指導老師:黃雅軒

陳宏輔、劉慎軒、呂振華、何星緯

目錄

1. 硬體配置…………………p.2
2. 系統軟體配置……………p.3
3. Python套件………………p.4
4. 系統架構說明……………p.5
5. 系統架構圖………………p.6
6. 各個model說明…………p.7
7. 使用到之函式說明…p.8~p.9
8. 可更改之cfg檔說明……p.10
9. 安裝說明流程說明………p.11
10. 錯誤說明…………………p.12
11. 其他注意事項……………p.13

硬體配置

CPU:四核以上CPU即可

GPU:GTX1050或以上之NVIDIA GPU

Memory:DDR4 16G

系統軟體配置

CUDA: cuda9.0(9.0以上tensorflow\_gpu不支援)

CUDNN:cudnn7.5.0.56

OS:Windows10

如配置與以上不同可能需要重新依照配置製作yolo的dll檔

Python套件

tensorflow-gpu 1.12

scipy

scikit-learn

opencv-python

h5py

matplotlib

Pillow

requests

psutil

pyqt5

pyqt5-tools

Pandas

keras

在python中下指令:

pip install -r requirements.txt

即可安裝以上套件

系統架構說明

輸入

影像(大小沒限制，可在cfg中做修改)

網路攝影機(可以是有固定IP的攝影機)

偵測

本系統使用YOLO作為偵測的核心。

辨識

本系統使用FaceNet為辨識核心。

註冊

本系統註冊時會擷取數張人臉影像存置於資料庫，並將計算出的特徵值一並存入，以便下次讀取。

輸出

jeson檔格式:

[name ,minx ,miny ,w ,h]

系統架構圖

初始化系統

初始化YOLO

初始化facenet

初始化攝影機

讀取資料庫內資料

載入設定檔

註冊

FaceNet辨識偵測到的臉

輸出jeson

[name ,minx ,miny ,w ,h]

辨識

擷取臉的影像

計算影像的特徵值

將影像及計算出的特徵值儲存寫檔



1一般攝影機



2 IPCam

Yolo偵測臉的位置

各個model說明

darknet.py:

yolo的核心model，請勿修改。

facenet.py:

facenet的核心modle，請勿修改。

facialDectionCore.py:

系統偵測模組

facialRecognitionCore.py:

系統辨識模組

facialRecognitionProcess.py:

系統整合模組

facialRecognitionMain.py

系統demo的main function

會使用到之函式說明

**系統初始化: Recognition()**

在系統class中的建構子即系統初始化，請記得將class實體化。

**系統資料初始化: initAllData()**

初始化系統所需要的資料，如果回傳False表示有資料為正確載入

**攝影機初始化: initCapDevice()**

初始化攝影機，系統初始化時已執行，如果發現攝影機需要重新連接時可呼叫。

**開始:startCap()**

啟動系統辨識，開啟所有執行緒。

**暫停: stopCap()**

暫停系統。

**註冊: registered()**

需要註冊時呼叫，回傳兩個值，第一個為是否要取得名字，第二個取得註冊之照片。

**註冊後設定: setRegName(name)**

在註冊完後設定註冊者名稱，name為傳入註冊者文字。

**辨識: recognized()**

需要辨識時呼叫，回傳Json。

**判斷是否有影像:ret**

ret是一個參數，如果是true則表示有影像，如果false則是沒抓取到影像。

注意事項

1. 以上標記紅色之函式請記得在辨識開始前呼叫，以利系統做初始化及載入。
2. 以上函式都屬於Recognition class，故在使用前請先將Recognition 實體化，例如: faceRecoProcess = Recognition()
3. Recognition class存在於facialRecognitionProcess.py內，故在使用時要import進來，例如:

from facialRecognitionProcess import Recognition

1. 使用方式可參考facialRecognitionMain.py內之程式碼。

可更改之cfg檔說明

更改cfg之檔案:

.\facialDetectionGUI\cfg\myConfig.cfg

[Default]

device = 0 影像輸入裝置，填IP cam網址或填一個數字(看是哪個裝置)

devicew = 1920 影像輸入寬

deviceh = 1080 影像輸入高

maxfacenum = 10 註冊存取照片數量

saveinterval = 0.5 註冊時間格多少秒存一張圖

regAreaw = -1 註冊框寬，如果輸入-1為預設

regAreah = 20 註冊框高，如果輸入-1為預設

jsonfilepath = ./json/detectionData.json json檔儲存位置

writejson = True 是否要寫json檔

[DectionCore]

thresh = 0.6 偵測之IOU閥值，越高代表偵測到臉的嚴謹性越高，不得超過1

[RecognitionCore]

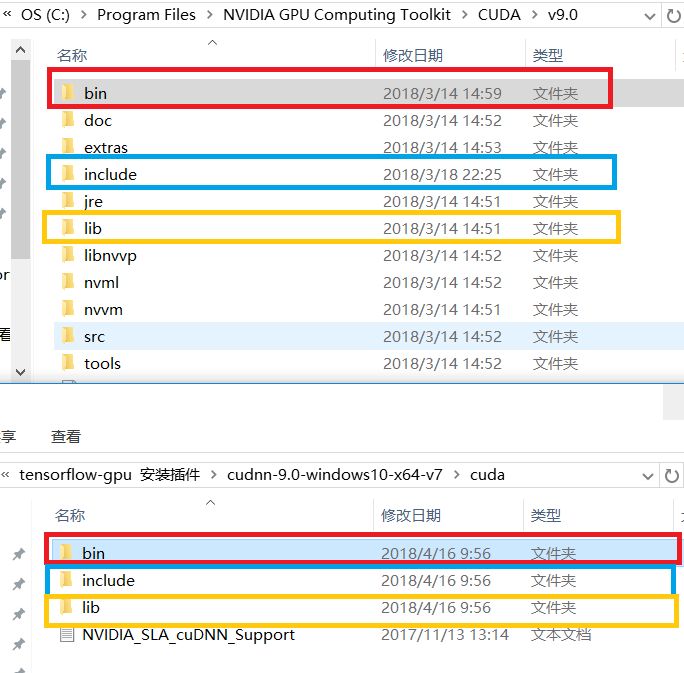
disttresh = 0.85 辨識之歐氏距離閥值，越低代表辨識嚴謹性越高

imagefolder = ./Image 註冊圖片儲存位置

安裝說明流程說明

step1.安裝cuda9.5

step2.下載cudnn7.5.0.56(只需下載base)

step3.將下載之cudnn資料夾解壓縮，並將裡面三個資料夾取代cuda裡的資料夾。如下圖: 

step3.安裝python(建議3.6)

step4.在python下安裝所需套件(可在cmd 進到此系統資料夾輸入pip install -r requirements.txt)

step5.安裝完成，可開始執行demo檔

錯誤說明

No fram:無法抓取到影像，請確定攝影機是否正常運作。

[WinErrow123]:一般為tensorflow\_gpu版本與cudnn不符，故請確認版本問題。

無法載入dll:請確認是否符合上述的版本。

其他說明

1. 如需查看資料庫，請查看image資料夾
2. 以上提供之系統軟體配置之版本需相同，不同可能會造成系統無法運行。
3. 在執行前請先確認影像輸入是否正常，如影像輸入不正常可能導致系統執行錯誤。
4. 系統主要是在GPU上運行，如果無GPU可能造成系統效能低落。