|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **2013軟體創作達人暑期成長營期末報告** | | | | | |
| **團隊名稱：** | | | | | |
| 身份別 | 姓名 | 學校 | 系別 | 連絡電話 | 電子郵件 |
| 指導老師 | 林朝興 | 國立臺南大學 | 資訊工程學系 | 0955786729 | mikelin@mail.nutn.edu.tw |
| 隊長 | 李侑軒 | 國立臺南大學 | 資訊工程學系 | 0988060853 | imshikar@gmail.com |
| 隊員1 | 江啟睿 | 國立臺南大學 | 資訊工程學系 | 0975122842 | ha9mv8c@gmail.com |
| 隊員2 | 胡銘軒 | 國立臺南大學 | 資訊工程學系 | 0910799735 | fake30820@gmail.com |
| 隊員3 | 曾楷珉 | 國立臺南大學 | 資訊工程學系 | 0978310723 | icarus0723@hotmail.com.tw |

|  |  |
| --- | --- |
| **原始專案內容** | |
| 專案題目 | 簡易人流計數器APP |
| 專案主持人 | 許顯達 經理 |
| 機構名稱 | 三商電腦股份有限公司 |
| 應用領域 | □雲端網際服務 □智慧感知與互動多媒體 ■行動終端應用  □現有開放軟體專案改善與其他應用軟體:\_\_\_\_\_\_\_\_\_\_\_\_ |
| 技術規格 | 產品或服務定義：  1.可紀錄與統計通過手機鏡頭人流  2.支援同步錄影，可驗證人流計數是否正確  3.支援遠端調閱記錄功能 |
| 技術功能描述：  需採用iOS (Objective-C) 或Android (Java) 原生程式進行開發 |
| 輸入輸出資料：  輸入：即時影像或預錄影片  輸出：人流紀錄、影片與人流統計 |
| 作業環境:  □Windows □FreeBSD □Linux □Java Environment  □.NET Environment □MacOSX □MacOS Classic ■其他 Android |
| 程式語言:  □Assembly □C ■C++ ■Java □Perl □PHP □Python □Ruby □其他\_\_\_\_\_\_ |
| 介面顯示支援語言(可複選)：■中文 ■英文 □其他\_\_\_\_\_ |
|  |  |
| **目前完成進度** (詳細說明已完成項目功能，另外可用文字或影音檔(AVI或MP4格式，最長三分鐘為限)補充說明系統完成雛型，請附件一併上傳)：70% | |
| 1. 影片來源選擇   可以讓使用者選擇已經拍攝完成的影片檔案，來進行人流分析，點選中間+按鈕系統會打開檔案瀏覽器，讓使用者選擇影片，目前可以支援多種影片格式  C:\Users\YH\Desktop\影片來源.png   1. 角度選擇   本系統提供三種不同角度的影片分析方式，使用者在選完影片後，需要根據影片拍攝的角度來選擇對應的分析方式，三種角度分別為：   1. 垂直拍攝→影片是由上往下拍的影片 2. 水平拍攝→影片是由人流向平行拍攝的影片 3. 傾斜拍攝→影片是由45度、60度等等非上面兩種方式所拍攝的影片   C:\Users\YH\Desktop\選擇角度.png   1. 影片分析   採用frame by frame的分析方式，針對每張frame進行分析，並且統計出整段影片所出現的人數，以及每分鐘平均的人數，可以隨時暫停或者繼續分析  C:\Users\YH\Desktop\影片分析.png | |
| **技術文件** (詳細說明所需相關軟體元件與系統軟硬體操作方式)：20% | |
| 本專案因為提供三種不同角度的分析方式，所以下列會分別針對三種角度的分析方式進行說明   * 垂直角度   C:\Users\YH\Desktop\Untitled (1).png   1. 影像灰階處理:    1. 讀取影像圖片    2. 再利用函式把圖片轉成灰階 2. 取邊界    1. 先將灰階影像進行高斯模糊    2. 利用Sobel分別取得水平與垂直方向的邊界    3. 將水平與垂直方向的邊界合併起來 3. 背景處理    1. 利用第一張灰階圖片當背景，之後每十張比較一次，每一像素都有一個最大值跟最小值，只要灰階圖片的該點像素大於十張圖片中該點最大值或小於十張圖片中該點最小值時更新最小值跟最大值，最後算出最大值跟最小值的差    2. 如果該點差異小於更新門檻值(10)，背景就會更新與此點相同座標的像素    3. 變數說明:       1. 該十張圖片像素值最大的點       2. 該十張圖片像素值最小的點       3. 十張圖片該點平均值       4. 背景更新後該點的像素值       5. 更新背景時，背景所佔的比例(0.3)       6. 原本背景該點的像素值 4. 量化差異    1. 利用公式把差異量化       1. 代表量化後的像素       2. 代表該點座標       3. 代表灰階圖片跟背景圖片該點像素的差異       4. 代表灰階圖片跟背景圖片像素差異的最大值       5. 代表灰階圖片跟背景圖片像素差異的最小值 5. 膨脹侵蝕    1. 先用侵蝕過濾掉雜訊    2. 再用膨脹把輪廓接起來 6. 二值化    1. 我們設定一個門檻值(20)，灰階圖片該點的像素，如果大於門檻值，像素變為255，反之，則變成0。 7. 取得前景邊界圖    1. 把二值化的結果跟邊界圖片結合即成為前景邊界圖。 8. 頭部樣板    1. 樣板的大小限定在200\*200       1. 有標準圓形       2. 頭+肩膀 9. 頭部偵測    1. 利用頭部樣板去偵測是否為人頭，    2. 再用 opencv 裡的matchTemplate 比對，該圖上每一點的像素代表灰階圖片跟頭部樣本的比對符合值，越符合值就越高    3. 利用 threshold 設門檻值把比對結果圖二值化    4. 設定一個相似度門檻值，若相似度大於此門檻，才確認為人頭 10. 預測位置     1. 以所偵測到的人頭座標，向右邊及下方各取5個點，再將這25個點的像素質存於陣列中，接著利用下方公式計算相似度來判別是否為同一點   r =  c(x, y) 目前所取的點像素  p(x, y) 前一張所取的點像素  當 r 越小時，則兩點相似度越高   1. 接著將偵測範圍設定以加快運算速度    * 1. 左右設為樣板寬度的1/2      2. 上下設為與樣板長度相同      3. 虛線內為有效的偵測範圍 2. 計算人數    1. 設變數 life 為偵測到的點的生命週期，每一張圖片增加 life 次數1次    2. 設變數 detected\_life 為偵測到的點又被追蹤次數，每追蹤一次 detected\_life 次數加1次    3. 降低誤判:       1. 當 life-dectected\_life > 4 時，認定此點為誤判，清除此點資料       2. 當座標過於接近時，以life大者為正確追蹤到的點，life小者於以清除       3. 設 bool 變數 count 為判別是否重複偵測    4. 當 count = false 且 點通過中線時，再以前後座標判別方向，以計算人數  * 水平角度   C:\Users\YH\Desktop\Untitled.png   1. 彩色影像:   由於HOG行人偵測採用彩色影像會有較好的效果，所以直接採用原本的影像不進行另外處理   1. HOG行人偵測   http://doi.ieeecomputersociety.org/cms/Computer.org/dl/trans/tp/2010/07/figures/ttp20100712397.gif  HOG即是採用圖像的「局部梯度」當成圖像的特徵，並且利用SVM來學習圖片中人的特徵，來找到圖像中的人，其步驟分為：   1. 計算每個像素的梯度:   利用未命名  ,做convolution。得到梯度後，便可以計算此pixel的方向跟強度。 其kernel公式為:  未命名   1. 切割cell做histogram   將圖像分成一格一格的cell，假設每格cell的大小為6\*6的pixel，在將這36個像素依照他們的方向(alpha)利用統計做一個histogram，而histogram的的X座標常見的是4-orientation或是8-orientation  未命名  其統計後得到histogram  未命名   1. 將cell分組在合併(比如說2\*2)當成block，而block的histogram也跟著合併，為代表此block 的直方圖，並且正規化其直方圖   未命名   1. 將所有已正規化的block直方圖串連成一個大的直方圖，此直方圖即為行人分析的特徵資料      1. 利用已經經由大量行人特徵資料訓練好的SVM來判斷，所取得的直方圖是否為行人，進而完成圖像中行人的偵測   (HOG資料參考來源：[1] Navneet Dalal, Bill Triggs, "Histograms of Oriented Gradients for Human Detection," International Conference on Computer Vision & Pattern Recognition (CVPR '05) 1 (2005) 886--893  [2] http://140.122.79.211/robotproject/report\_data/plan\_5\_research\_report\_2.pdf  [3] http://en.wikipedia.org/wiki/Histogram\_of\_oriented\_gradients)   1. HOG結果過濾:   因為HOG會有誤判的情況，所以我們採用與垂直角度相同的取得前景方式，來得到當前影像的前景圖，判斷HOG所得到座標在前景圖中，是否有像素點的存在，當其像素點總數大於門檻值及代表其HOG所得到的物體是會移動的，而其就有很大的機率為人。   1. 人數統計: 2. 假設*A*, *B*, *C* 三個變數，並且皆設為0 3. 當HOG方法偵測到影像中出現人，則*A*加上影像中人的數量，*B*加1 4. 當*B*的數值達到行人同過畫面所需張數的門檻值(此門檻值為一經驗值，以30 fps鏡頭拍攝的影片為例，經驗值平均為125張圖，因HOG偵測上會有誤差，所以行人同過畫面的時間約為4~6秒) ，則進行*A/B*並將值加入*C*，並且將*A*、*B*歸零，*C*即為目前的人流統計數量 5. 當到達最後影格時，再做一次C = *A* / *B*， C即為整段影片的人流總數   此方法在短時間的誤判上比較多，但在長時間的計算上，誤差隨著時間漸漸可被忽略掉。   * 傾斜角度   因為HOG的偵測方式可以適用於傾斜角度與水平角度，所以本專案的傾斜角度分析方式所採用的方法與水平角度類似，僅在一些門檻值上所設定的數值不同。 | |
| **軟體清單** (專案開發所需相關軟體元件清單) | |
| 1. Android SDK 2. Android NDK 3. OpenCV for Android 4. FFmpeg 5. JavaCV for Android 6. Eclipse | |
| **專案成果預定授權條款** (請務必依循與尊重現有專案的授權模式，避免不相容的議題) | |
| 因採用的JavaCV for Android所採用的授權方式為GPLv2，所以本專案由原本的Apache License 2.0授權方式改成為GNU General Public License version 2 (GPLv2) | |
| **專案開發討論紀錄** (參與團隊須全歷程專案專案管理開發討論紀錄平台網址)：10% | |
| <http://of.itsa.org.tw/projects/1310> | |