

AOI 影像處理技術於雲端人臉偵測之應用

吳明芳* 陳俊榮* 陳添智** 林旭政*** 黃景賢****

*崑山科技大學電機系 副教授

**崑山科技大學電機系 教授

***工研院南分院 研究員

****崑山科技大學電機系 研究生

摘要

傳統的監視系統大部分都是人們需要坐在監控端前面盯著螢幕，監視時間一旦久了，注意力就會相對的不集中，同時會造成即使物體進入監視畫面也會不易發現的情況發生，進而造成許多的意外事故或社會案件的發生；因此本文提出一個可人臉偵測的智慧雲端監視架構的系統，希望藉由雲端的高運算能力來實現遠端監視系統人臉偵測，以降低人工監視之缺點。

在人臉偵測方面，本文使用了 Apache 公司所發展出來的軟體 Hadoop，使用此軟體是因為 Hadoop 屬於分散式運算平台，其能有效的將數據量大的資料透過 Map/Reduce 的平行化處理方式，能比傳統的運算方式更加快速的將處理結果運算出來，對於人臉影像處理的速度有很大的幫助。

本文主要發展一套以雲端為中心的智慧型監控系統；此雲端系統可分為三大部分，第一部分是 CCD(Charge-Coupled-Device)攝影機控制端，負責傳送影像資料；第二部分為影像處理端，負責處理影像中偵測人臉的部分，處理後的人臉數據保留在伺服器中，而人臉偵測結果則透過第三部分網路應用伺服器來顯示。

關鍵詞：影像處理、人臉偵測、雲端、監視系統、Hadoop

壹、緒論

人臉偵測是透過影像去搜尋人臉的部分，一般常見方法是透過皮膚膚色、形狀及臉部特徵等資訊去做判斷[1]。近幾年來人臉偵測的方法不斷更新進步，使得人臉偵測及追蹤的應用逐漸廣泛，例如監視系統、保全系統、人臉辨識系統等等...都是透過人臉偵測等技術的應用範圍。傳統的監視系統大部分都是人們需要坐在監控端前面盯著螢幕，根據 Goolkasian[2]提到，當人們的眼睛不停的注視監控端螢幕，注意力集中的時間大約是 20 分鐘，一旦超過時間，就算有物體進入監控螢幕，也不會去注意到；因此智慧型



監視系統的研究逐漸開始興起，如文獻[3]以移動目標平移法與移動邊緣檢測法對物體作追蹤，但此方法只要移動物體就會去進行追蹤，非常容易發生錯誤；後續文獻[4]利用膚色去辨識出可能為人臉的區域。

雲端服務是指人們可以透過網路來利用伺服器本身運算能力所衍生出來的一種服務，人們所需要的資料或檔案不是儲存在個人電腦上，而是放在網路的「雲」上面，在任何可以使用網路的地方就可以存取。所謂的雲代表了規模龐大的運算能力，由伺服器服務商建造大型機房，提供各類軟體應用，讓用戶隨時可以使用媲美超級電腦的運算能力與最新應用軟體，但是卻不曉得伺服器的位置或資料的所在，就好像是天上的雲一般，虛無飄渺但抬頭卻看得見。

目前市面上所出現的人臉辨識系統，大多是一套系統必須搭配一台運算主機，例如機場人臉辨識系統、保險箱人臉辨識系統等等...都是需要各一台主機；假設一家銀行分行越多，所需要的成本就會相對提高。

雲端運算[5]是由許多擁有儲存容量高、彈性高及運算效能高所組合而成的運算伺服器，在日常生活中雲端運算就隨時伴隨著人們，就像[6]中所說的，日常使用的 Google 瀏覽器就是使用雲端運算來處理每天的資料搜尋、資料結合等程序[7]，並且將這一種屬於平行分散式處理架構的技術方法命名為 Map/Reduce；Hadoop 的主要架構[8]是從 Google 所發表的 Map/Reduce 等概念實作而成的；目前 Hadoop 使用在如 WordCount、Sort 等低複雜度且高密集度的運算以及其他相關需要排程[9][10][11]的應用程式。

本論文所提出的可人臉偵測之智慧雲端監視系統，就是利用 Apache 公司所開發的分散式運算平台 Hadoop 來加快影像處理速度並實現人臉偵測目的，再將其運算結果結合網際網路，使得監控者可以透過網頁，隨時隨地得到監控結果的資訊，達到雲端服務的目的。

貳、人臉偵測技術概述

『人臉偵測』是指在一張影像中去搜尋一個或多個可能為人臉的區域；在人臉偵測過程中，大致上可以利用臉部膚色、頭部形狀，臉部的特徵及臉部動作表情等資訊去判斷是否為一個人臉區域，並且利用此結果來衍生出追蹤的方法；『影像』大致上可分為靜態影像及動態影像，靜態影像就是一般有照相功能所拍出的單張照片，而動態影像就是所謂的錄影或影片模式，是將許多張靜態影像組合而成的。人臉偵測的方式主要是利用影像處理技術完後所產生的圖像再去做人臉區域的分析以及定位，本節首先敘述本研究所使用的影像前處理技術，然後介紹人臉區域偵測技術的實現方法。

一、影像前處理技術

在人臉偵測中，影像處理是最重要的角色，因其影響了程式運算的時間長短及人臉偵測的結果。一個好的影像處理結果會使人臉偵測快速檢測完成；相反的，不好的影像處理方式會造成人臉區域不易判斷或者判斷不出來，造成系統處理的時間將拉長及容易判斷錯誤。本文所使用的影像處理技術有彩色影像轉灰階影像、低通濾波、邊緣檢測與二值化、外形影像運算等影像處理技術，分別敘述如下：



1. 彩色影像轉灰階影像

大部分做影像處理的人會採用灰階影像做處理的原因，主要是因為彩色影像處理的演算法是灰階影像處理的三倍(RGB 各需一個演算法)，相對的處理速度也需要花費更長的時間去做運算。彩色影像就是在每一點的二維陣列中摻入了三原色 RGB 的值，所以只要將每一個二維陣列予以灰階化即可轉化成灰階影像[14]，其轉換公式如(2-1)式所示。

$$\text{Grayscale}(x, y) = 0.299R + 0.587G + 0.114B \quad (2-1)$$

其中 $\text{Grayscale}(x, y)$ 為灰階影像之二維陣列在座標 (x, y) 時的灰階值， R 、 G 、 B 則分別代表彩色影像該對應座標的紅色值、綠色值及藍色值。例如圖 1(a)的彩色影像經(2-1)式灰階化的結果如圖 1(b)所示。

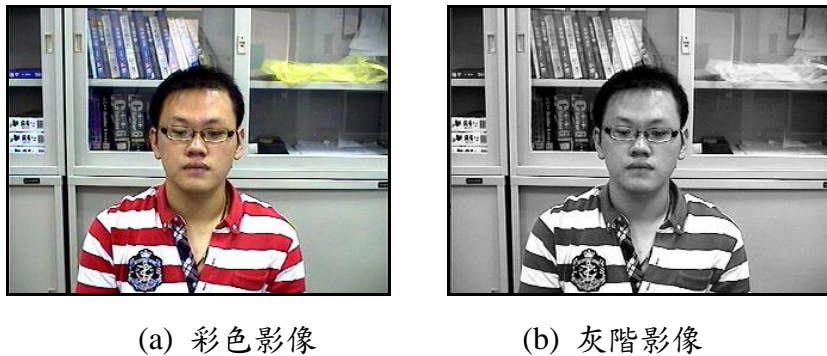


圖 2-1 彩色影像與灰階影像

2. 低通濾波、邊緣檢測與二值化

一般拍攝影像時，影像容易受雜訊的影響，雜訊會干擾影像中高頻的部分，雜訊越多，干擾越大，進而造成邊緣檢測產生許多雜訊點。為了避免雜訊點的干擾，本文先採用均化濾波技術進行影像低通濾波處理，來降低雜訊之強度，以利進行邊緣檢測之後續處理。邊緣是指在兩個區域中有著相當不同灰階特徵的邊界所組合而成的，一般使用 Sobel 方法[12]來實施邊緣檢測，此方法利用一階導數的大小來偵測出邊緣的所在處，將偵測出的邊緣利用連結的方式連結成邊界。Sobel 方法運用了梯度的概念，經 Sobel 處理後的結果，可再利用二值化取一個門檻值來突顯其效果。Sobel 邊緣檢測中一個重要的訊息就是一階導數向量的強度大小(magnitude)，我們通常稱為梯度(gradient)，如(2-2)式所示。

$$M = |G_x| + |G_y| \quad (2-2)$$

其中 M 表示梯度大小， G_x 與 G_y 則分別表示水平與垂直成分的梯度值，其在數位影像中的近似運算可分別由圖 2-2 所示的 Sobel 水平及垂直運算遮罩來達成。



$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

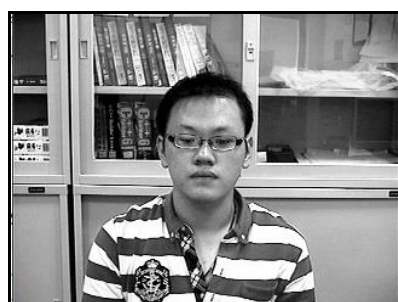
(a) 水平遮罩

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

(b) 垂直遮罩

圖 2-2 Sobel 運算遮罩

例如圖 2-3 所示為灰階影像經低通濾波、邊緣檢測與二值化後的結果。



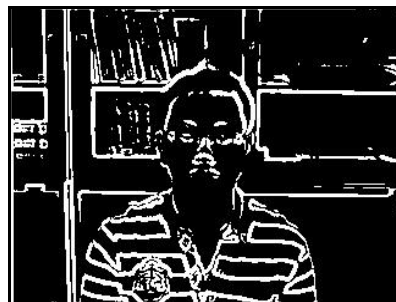
(a) 灰階影像



(b) 影像低通濾波



(c) Sobel 邊緣檢測



(d) 二值化影像

圖 2-3 灰階影像經低通濾波、邊緣檢測與二值化後的結果

3. 外形影像運算

以二值化影像而言，人類視覺可以輕易判斷出影像中的位置、大小、以及形狀，但假如是用機器視覺來判斷的話，將難以直接快速並準確分辨出物體，因此本研究施以一系列的外形影像處理，包含開閉運算、連接元區域標定等，分述如下：

(1) 開閉運算

開閉運算為形態學影像外形處理技術之一，開閉運算主要是利用膨脹(dilation)與侵蝕(erosion)兩種方法的結合運算，在數位影像處理中稱之為數學形態學(mathematical morphology)；在形態學中的斷開(opening)與閉合(closing)運算常用來消除影像中常常存在的一些小雜點或雜訊點；膨脹運算的功用主要是將外形均勻的加粗；侵蝕運算的功用相反，主要將外形均勻的縮小，至於斷開運算的功能可去除影像孤立點；而閉合運算的效果可以將稍微不連續的部分連



接起來。本文之膨脹與侵蝕運算可分別由(2-3)及(2-4)式來實現：

$$D(i, j) = x \cup (x_0 \cup x_1 \cup x_2 \cup x_3 \cup x_4 \cup x_5 \cup x_6 \cup x_7) \quad (2-3)$$

$$E(i, j) = x \cap (x_0 \cap x_1 \cap x_2 \cap x_3 \cap x_4 \cap x_5 \cap x_6 \cap x_7) \quad (2-4)$$

其中 $D(i, j)$ 跟 $E(i, j)$ 分別代表膨脹與侵蝕運算後的新影像， x 為每一 3×3 遮罩之中心點， $x_0 \sim x_7$ 則為 x 之八個鄰邊點。斷開運算是先侵蝕運算後再經過膨脹運算處理，假如影像物體外部有些雜訊小點，可透過斷開運算處理把外部這些雜訊小點消除；閉合運算則是將影像先經過膨脹運算後再經過侵蝕運算，假如在物體內部有雜訊小點，可透過閉合運算處理將這些雜訊小點消除。通常斷開運算及閉合運算常常被一起使用，如先執行斷開再經過閉合運算可以達到平滑的效果。

(2) 連接元區域標定

本研究所使用之連接元區域標定(connected component labeling)[13]，主要是把影像中屬於同一個物件的互相關連像素找出來，以獲得物件的範圍、面積、高度和寬度大小等相關資訊，並藉此用來標定出影像中各個獨立的物件。主要做法為掃描整張影像的像素值，從左上角開始，由左而右，由上而下，像素值為 1 時，先將此點紀錄下來，並依序去檢查該點的上、下、左、右是否有同樣為 1 的像素值點，並且將剛剛所記錄的像素值設定為 0。然後依遞迴的方式去檢查剛所記錄點之上、下、左、右是否也有 1 的點存在，如果也有，將該點紀錄下來，並將其值設為 0。持續整個遞迴的檢查，直到所有紀錄點之周圍都不再有 1 的點出現為止，即停止遞迴整個檢查的動作。當影像都掃描完畢，並完成所有檢查工作後，就可以統計出影像中所有的物件面積、高寬的範圍等資訊。

二、人臉區域偵測技術

本研究在人臉偵測的過程中主要運用了影像前處理處理、膚色偵測、橢圓遮罩比對、嘴唇偵測及眼睛偵測等運算法將人臉區域搜尋出來，再加以判斷是否為人臉[1]。

如圖 2-4 所示為人臉偵測流程圖，一開始先對輸入的影像作顏色分割，再收尋膚色區域，並且利用低通濾波及 Sobel 等方法偵測影像邊界，此步驟是為了方便橢圓遮罩在膚色區域中做比對，當膚色區域符合橢圓形比對條件，則判斷可能為人臉區域，然後再對此區域做嘴唇與眼睛的特徵偵測，以確定是否為真的人臉區域。圖 2-4 之膚色區域標定、橢圓頭部比對以及眼睛嘴唇特徵偵測與定位等各流程之詳細原理及演算法在文獻[1]中已有非常詳細的推演。

例如圖 2-5 所示，圖 2-5(a)之原始人臉影像經膚色區域標定、橢圓標定及眼睛嘴唇標定後之結果分別如圖 2-5(b)~(d)所示，最後人臉偵測結果則如圖 2-5(e)所示。



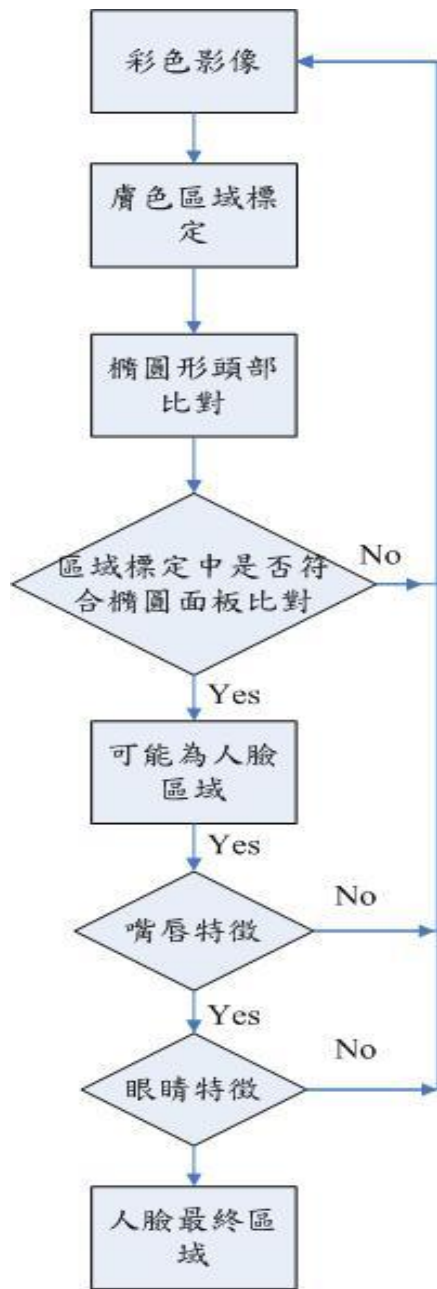


圖 2-4 人臉偵測判斷流程圖[1]

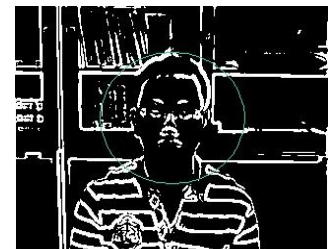
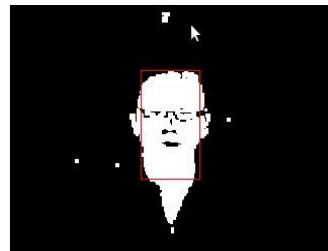


圖 2-5 人臉判定結果



參、Hadoop 雲端系統概述

為了達到雲端運算的功能及目的，本文將人臉偵測技術結合 Hadoop 去做影像運算，並建置在雲端機器內，透過網路傳輸進行影像處理等工作。Hadoop 是經由 Google 雲端架構之啟發進而所開放的原始碼計劃，目前有許多組織參與 Hadoop 的研究開發(例如:Yahoo、Amazon 等...)，並以 Hadoop 做為雲端運算的平台。

Hadoop 是 Apache 軟體基金會 (Apache Software Foundation) 底下的開放原始碼計劃(Open source project)，最初是用來做為開放原始碼 Nutch 的搜尋引擎一部份，Hadoop 是利用 java 語言撰寫而成的，主要是負責提供處理大量資料的分散式運算環境；Hadoop 的主要架構是利用 Google 發表的 BigTable 及 Google File System(GFS) 等文章提出的概念實作而成，所以跟 Google 內部使用的雲端運算架構非常相似；目前 Yahoo! 及 Cloudera 等公司都有開發人員投入 Hadoop 的開發團隊，也有將近一百個公司或組織公開表示使用 Hadoop 做為雲端運算平台，就連 Google 及 IBM 也使用 Hadoop 平台做為教育合作環境。

表 3-1 所示為 Hadoop 與 Google 架構比較表，Hadoop 中包括許多子計劃，其中 Hadoop MapReduce 如同 Google MapReduce，主要提供分散式運算環境；Hadoop Distributed File System(HDFS) 如同 Google File System(GFS)，主要提供大量儲存空間；HBase 是一個類似 BigTable 的分散式資料庫，還有其他部份可用來將這三個主要部份連結在一起，方便提供整合的雲端服務。

表 3-1 Hadoop 與 Google 架構比較

Develop Group	Google	Apache
Algorithm Method	MapReduce	Hadoop MapReduce
File System	GFS	HDFS
Storage System (for structure data)	BigTable	HBase

本研究主要是利用 MapReduce 的機制來進行人臉偵測的影像處理。圖 3-1 所示為 MapReduce 流程圖，MapReduce 是一個分散式程式框架，主要是讓開發者可以很簡單的撰寫程式，再利用大量的運算資源，加速處理龐大的資料量，一個 MapReduce 的運算工作可以區分成 Map 及 Reduce 兩個部份，大量的資料在運算一開始的時候，會被系統轉換成多組 (key, value) 的序對並自動切割成許多部份，分別傳給不同的 Mapper 來處理，當 Mapper 處理完成後也會將運算結果整理成多組 (key, value) 的序對，再傳給 Reducer 整合所有 Mapper 的結果，最後才能將整體的結果輸出。



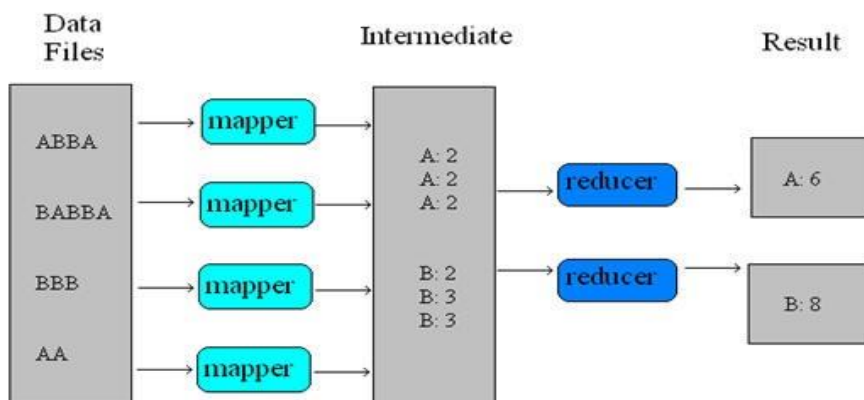


圖 3-1 MapReduce 流程圖

肆、雲端人臉影像偵測實驗結果

一、系統架構

本研究在雲端人臉影像偵測實作方面，可以區分為三大系統架構，如圖 4-1 所示，第一部份為攝影機客戶端模組、第二部分為影像處理伺服器模組及第三部份影像監視模組，以下分別描述其功能：

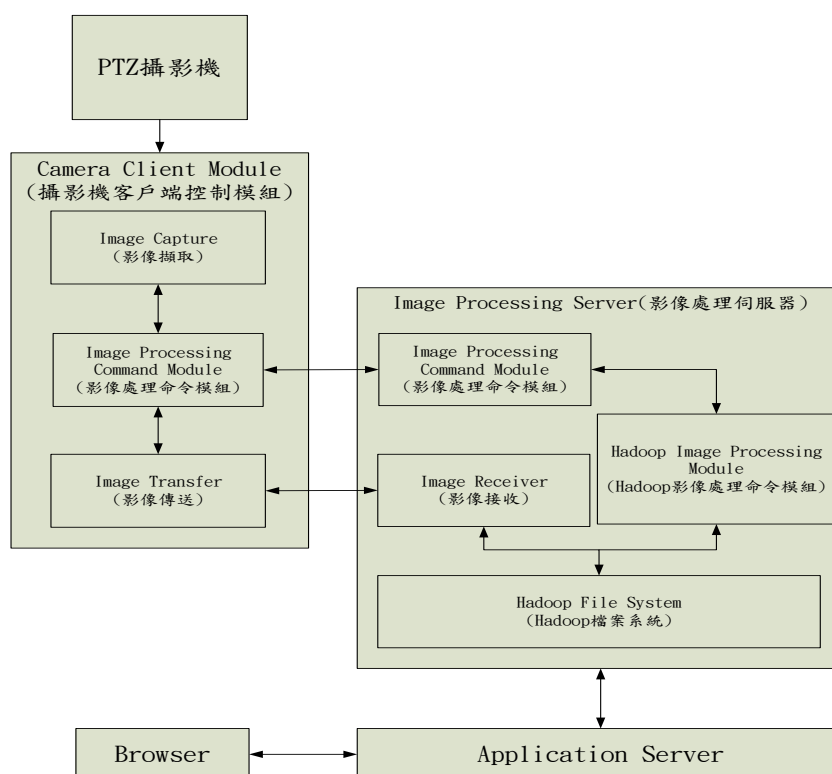


圖 4-1 系統架構圖



1. 攝影機客戶端模組

本研究使用 CCD 攝影機做為影像擷取裝置，其後端使用 AV 端子做為輸出介面；本文再使用 UPMOST UG300A 的 AV 轉 USB 轉接卡連接到 PC，以 PC 做為模擬 CCD 攝影機的控制系統，此控制系統以 Windows 為作業系統，並在 Borland C++ Builder 開發環境下配合 Microsoft 的 Windows Multimedia[21]函式庫，如圖 4-2 所示做為擷取影像的方案，透過此函式庫，控制系統可以選擇所要擷取影像的解析度以及 Color Space 等視訊格式設定，如圖 4-3 所示，此函式可以依照不同的演算法及影像色度空間所需而調整。以本文來說，使用的是 Sony D70[22]的 CCD 攝影機，此攝影機的訊號為 NTSC，必須將接收的訊號透過影像擷取卡轉換成原始的 RGB24 格式，並輸出成 320x240 圖片大小。Image Capture 模組實作此影像擷取及設定的功能。

capDlogVideoFormat(vcHWND);

圖 4-2 設定影像格式的 Windows 函式

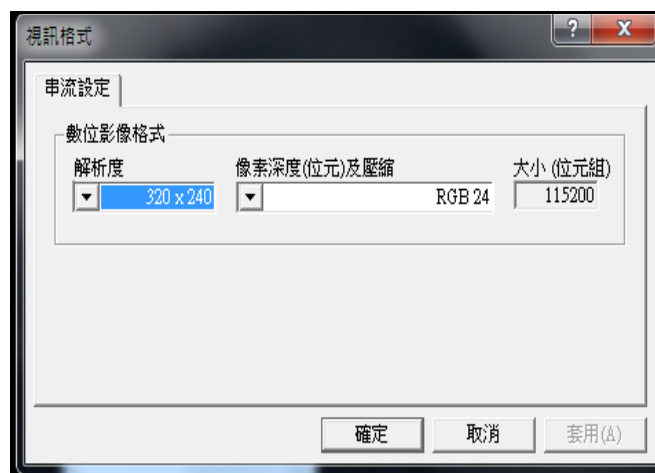


圖 4-3 設定影像格式的使用者介面

圖 4-1 中的 Image Processing Command Module 為控制模組的核心，主要是負責維護目前已經擷取到的影像檔案列表 FIFO Queue，並連接雲端伺服器相對應的 Image Processing Command Module，然後傳送影像檔到雲端伺服器，並下達影像檔已傳送完成命令。這些動作為多執行緒同步工作，透過事件驅動的方式達成訊息的交換。

圖 4-1 中的 Image Transfer 模組是透過持續監聽 FIFO Queue 來確認是否有來自 Image Processing Command Module 所下達的控制命令，FIFO Queue 若不是空的話，Image Transfer 模組則會將 FIFO Queue 裡所指定的 CCD 攝影機擷取到的影像資料檔案傳送到雲端伺服器，雲端伺服器會有相對應的接收模組去儲存影像檔。



2. 影像處理伺服器

相對於 Camera 端的 Image Processing Command Module，雲端伺服器也提供相對的模組，做為雙方交換訊息之用。當有來自 Camera 端要求處理影像的控制命令，會觸發 Hadoop Image Processing Module 來處理影像的資料，Hadoop Image Processing Module 使用 Apache Hadoop project [12]來做開發。

Image Receiver 相對於 Camera 端的 Image Transfer 模組，做為接收來自 Camera 端所傳送過來的影像，當接收到由前端所傳送的影像資料檔之後，就會將這些影像檔放到指定的 HDFS 當中，並更新位於 HDFS 中的影像索引檔，Image Receiver 會根據收到的影像，解析必要資訊並持續更新索引檔，該索引檔案的格式如表 4-1 所示。

表 4-1 索引檔案的格式

Image Information	Image HDFS Location
capture.jpg,320x240,datetime1	hdfs://site/capture.jpg
capture2.jpg,320x240,datetime2	hdfs://site/capture2.jpg
...	...

表 4-1 所示的內容為 Image Receiver 所更新的索引檔案內容，第一個欄位為 Image Information 記錄一個 Camera 畫面的檔案名稱、圖形大小與接收時間，例如：capture.jpg 畫面大小為 320x240，接收時間為 datetime1 等。索引檔案會持續寫入，當 Hadoop Image Processing Module 要執行影像處理工作前會先通知 Image Receiver，當 Image Receiver 接收到指令後會關閉該檔案，開啟新的索引檔案繼續等待資料做索引，並通知 Hadoop Image Processing Module 該開啟哪個索引檔案做分析，這段程式碼位於 Hadoop Image Processing Module 的 main 函數內，屬於啟動 MapReduce 的前置工作。

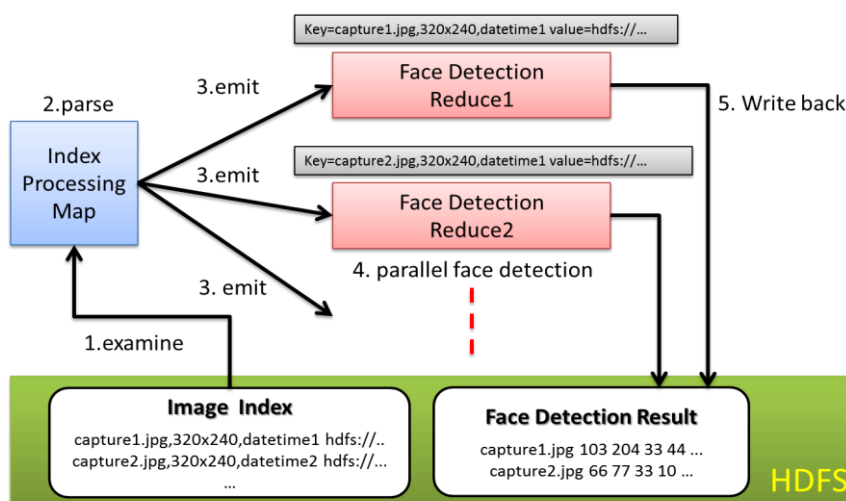


圖 4-4 Hadoop Image Processing 流程圖

圖 4-4 所示為 Hadoop Image Processing Module 的工作流程，Hadoop Image



Processing Module 的工作原則是以前 MapReduce 的方式執行影像分析的工作，為了達到最大平行化本文利用在 Map 函數內讀取索引檔，並且檔案名稱當作鍵值 (Key)，檔案資訊與檔案位置當作參數發送給 Reduce 做處理。

3. 影像監視模組

影像監控模組主要是透過網路瀏覽器來觀看人臉偵測結果，透過選擇欲預覽的時間來觀看。

二、人臉偵測實驗

本實驗主要驗證本研究提出的雲端人臉影像偵測架構對人臉影像的膚色偵測、橢圓遮罩偵測、嘴唇偵測及眼睛偵測等進行實驗。為了測試人臉偵測的效率，本實驗分為成三個實驗組，分別是純人臉偵測、戴眼鏡之人臉偵測及戴帽子之人臉偵測，最後將這三個實驗整理成人臉偵測結果來證明本研究之雲端人臉偵測技術理論的實用性及可行性。

1. 純人臉雲端偵測實驗

本實驗影像張數主要依據影像監視模組一分鐘所擷取的畫面，一分鐘 60 秒，每一秒由 30 張影像組成，總數為 1800 張；實驗目的在於測試當人臉無其它干擾物 (如眼鏡)，並且左右移動之人臉偵測效率，圖 4-5 為純人臉偵測結果，實驗數據如表 4-2 所示。

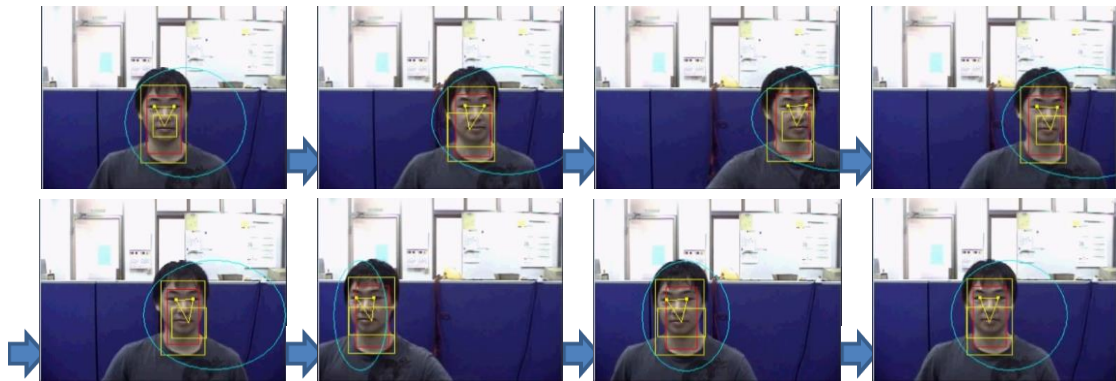


圖 4-5 純人臉偵測結果

表 4-2 純人臉雲端偵測實驗數據

測試影像張數 1800 張			
偵測實驗名稱	偵測成功張數(n)	偵測失敗張數	偵測成功率(n/1800)
膚色偵測	1800	0	100%
橢圓遮罩偵測	1656	144	92%
嘴唇偵測	1647	153	91.5%
眼睛偵測	1638	162	91%

表 4-2 所示為純人臉偵測各種偵測的結果統計，由表中會發現橢圓偵測的結



果會影響後面的嘴唇偵測及眼睛偵測結果，如果加大橢圓遮罩的搜尋範圍，嘴唇及眼睛偵測的效果越好，但相對所花費的時間也會相對的提高，因此可依照所需來調整橢圓收尋的範圍。

2. 戴眼鏡之雲端人臉偵測實驗

本實驗目的在於測試當人臉有較小干擾物(如戴眼鏡)，並且左右移動之人臉偵測效率，圖 4-6 為戴眼鏡之人臉偵測結果，實驗數據如表 4-3 所示。

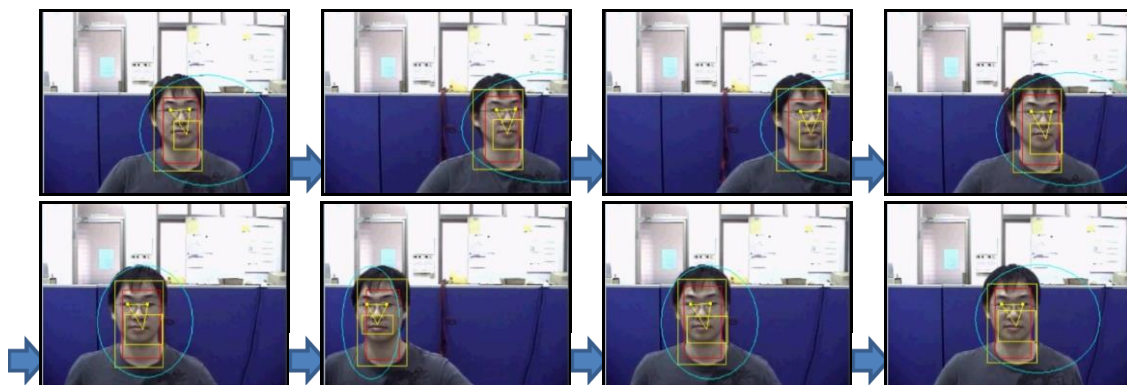


圖 4-6 戴眼鏡之人臉偵測結果

表 4-3 戴眼鏡之雲端人臉偵測實驗數據

測試影像張數 1800 張			
偵測實驗名稱	偵測成功張數(n)	偵測失敗張數	偵測成功率(n/1800)
膚色偵測	1800	0	100%
橢圓遮罩偵測	1638	162	91%
嘴唇偵測	1629	171	90.5%
眼睛偵測	1620	180	90%

表 4-3 所示為戴眼鏡之人臉偵測各種偵測的結果統計，由此表與表 4-2 互相比較會發現戴眼鏡之人臉偵測出來的數據會稍微低於純人臉偵測，主要是受到戴眼鏡之干擾造成橢圓偵測的結果會影響後面的嘴唇偵測及眼睛偵測結果，如果加大橢圓遮罩的搜尋範圍，嘴唇及眼睛偵測的效果越好，但相對所花費的時間也會相對的提高，因此亦可依照所需來調整橢圓收尋的範圍。

3. 戴帽子之雲端人臉偵測實驗

本實驗目的在於測試當人臉有較大干擾物(如戴帽子)，並且左右移動之人臉偵測效率，圖 4-7 為戴帽子之人臉偵測結果，實驗數據如表 4-4 所示。



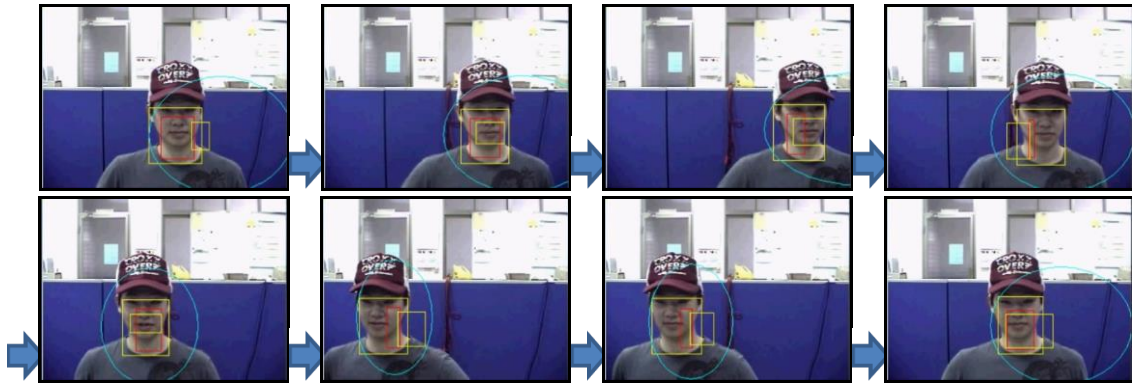


圖 4-7 戴帽子之人臉偵測結果

表 4-4 戴帽子之雲端人臉偵測實驗數據

測試影像張數 1800 張			
偵測實驗名稱	偵測成功張數(n)	偵測失敗張數	偵測成功率(n/1800)
膚色偵測	1800	0	100%
橢圓遮罩偵測	1576	224	87.6%
嘴唇偵測	769	1031	42.7%
眼睛偵測	0	1800	0%

表 4-4 為戴帽子之人臉偵測各種偵測的結果統計；由表中會發現膚色偵測成功率亦為 100%，代表本文所使用的演算法有一定的強健性；橢圓遮罩偵測因受到帽子陰影影響，成功率降為 87%；嘴唇偵測因橢圓偵測成功率較低的關係，造成收尋人臉嘴唇區域位置與實際位置有所偏移，其成功率僅為 42.7%；至於眼睛偵測，會發現當戴上帽子時，因為眼睛受到帽子陰影干擾甚鉅，加上橢圓遮罩偵測之成功率較低的影響，造成人臉眼睛區域位置的偵測失敗。以圖 4-8 來加以說明，會發現雖然膚色區域、橢圓遮罩及嘴唇區域還可偵測出來，但會發現因為戴帽干擾的關係，致使偵測到的嘴唇比實際人臉嘴唇的位置常有大量偏移的現象，因此，整體而言偵測出來之人臉區域(紅色框框所示)，顯得較為不理想，其原因主要是當戴上帽子易產生陰影(圖中眼睛到鼻子部分)，已經超過膚色標定區域之範圍值，所以系統會將此陰影排除在膚色之外，這是造成眼睛及人臉偵測失敗之最大因素。

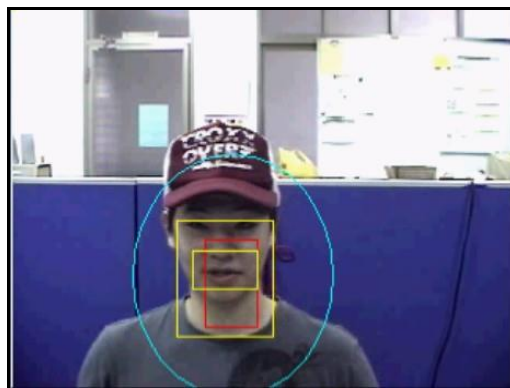


圖 4-8 戴帽子之人臉偵測結果



4. 整體雲端人臉偵測結果(不含戴帽子情況)

表 4-5 綜合了表 4-2 及 4-3，為純人臉偵測及戴眼鏡之人臉偵測各種偵測之結果統計，可以發現膚色偵測率為 100%，橢圓遮罩偵測率及嘴唇偵測率分別為 91.5% 及 91%，眼睛偵測率也有 90.5%，驗證了本研究之雲端人臉偵測方法有相當的實用性。

表 4-5 整體雲端人臉偵測實驗結果(不含戴帽子情況)

測試影像張數 3600 張			
偵測實驗名稱	偵測成功張數(n)	偵測失敗張數	偵測成功率(n/3600)
膚色偵測	3600	0	100%
橢圓遮罩偵測	3294	306	91.5%
嘴唇偵測	3276	324	91%
眼睛偵測	3258	342	90.5%

三、雲端單緒影像處理

單緒是單一執行緒的簡稱，是指程式在同一時間內，只執行一項工作；本實驗主要測試在同一台雲端機器上，運算不同資料量所花費的時間，藉以證明平行化應用的實用性，表 4-6 為實驗結果所產生的數據，實驗結果的處理秒數包含 Hadoop 資料佈署，影像分析處理及處理完的資料結果回存到資料庫的時間。

表 4-6 單緒處理結果

張數	1	5	10	50	100
秒數	18.72	18.26	18.27	18.41	18.53
張數	200	300	400	500	1000
秒數	18.38	18.37	18.35	18.33	18.25

圖 4-9 所示為由表 4-6 數據所繪製之單緒處理曲線圖。從圖 4-9 可發現處理一張圖所花費的時間大約是 18 秒左右、處理 500 張大約 18 秒左右，而 1000 張所花費的時間也大約是 18 秒左右。從此數據看來，可以了解到 Hadoop 平行化處理的優勢，應用在影像中，不管同時處理幾張，所花費的時間相差不遠，對於提升大量影像的處理速度有很大的幫助。

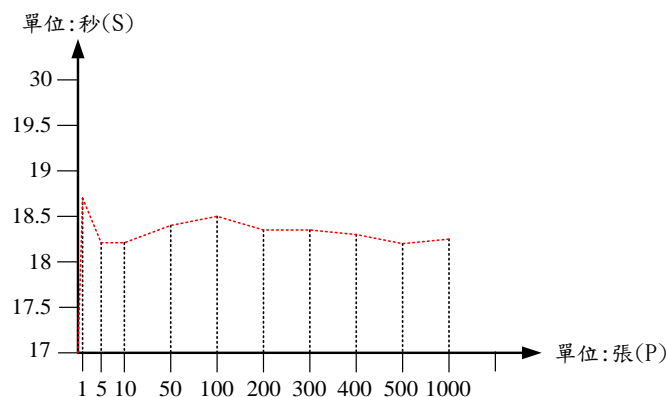


圖 4-9 單緒處理結果曲線圖



三、雲端多緒影像處理

多緒是多執行緒的簡稱，是指程式在同一時間內，執行多項工作；本實驗主要測試在同一時間，同一台雲端機器，運算不同筆資料，所花費的時間；表 4-7 為實驗結果所產生的數據，其中，同時執行一項工作的意思是指 Hadoop 同一時間，處理一秒(14 張圖)的資料量；同時執行兩項工作的意思是指 Hadoop 同一時間，處理 2 秒(共 28 張圖)的資料量，以此類推，並且繪製其平均值的曲線圖，如圖 4-10 所示。

表 4-7 多緒處理結果

同時執行/花費時間	工作一	工作二	工作三	工作四	工作五	平均花費時間
一項	20.3					20 秒
二項	28.9	29.1				29 秒
三項	31.2	32.4	46.3			36 秒
四項	35.4	36.1	55.4	56.3		46 秒
五項	37.4	38.4	57.3	59.1	60.2	50 秒

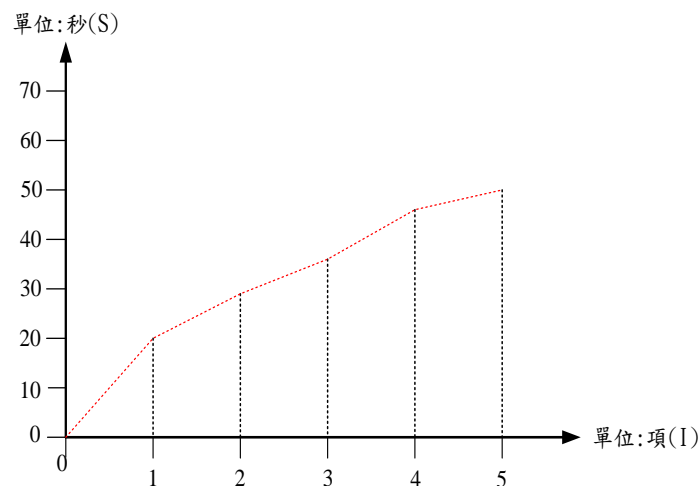


圖 4-10 多緒處理結果平均曲線圖

從圖 4-10 中可以發現，當 Hadoop 處理一項目所花費的平均時間大約是 20.3 秒，同時處理三項所花費的平均時間大約為 36 秒，而處理五項所花費的平均時間則是 50 秒；由此數據可知道，當攝影機每一秒擷取 14 張圖為一項工作，下一秒所擷取的圖為第二項工作，Hadoop 運算方式是先處理完一批的運算資料，才會再進行處理下一批的運算資料，因此會發現 Hadoop 的工作特性是當工作項目越多，運算時間也就越久。

伍、結論

本文所提出的雲端人臉影像偵測方法成功實現了運用雲端運算技術 Map/Reduce (Hadoop)來處理人臉偵測的目的，並形成一個以雲端運算為中心之雛型雲端人臉偵測及監視系統。



此監視系統包含了 Camera 端的控制模組及以雲端影像處理與控制伺服器；Camera 端的控制模組將擷取的影像透過網路傳送到雲端影像處理與控制伺服器，Camera 端的控制模組包含使用者介面、Image Capture Module (影像擷取模組)、Image Processing Command Module(命令控制模組)以及 Image Transfer Module(影像傳輸模組)；雲端影像處理與控制伺服器上包含了 Image Processing Command Module(命令控制模組)用以接收 Client 的連線及欲處理的影像名稱、Image Receiver(影像接收模組)、以及 Hadoop Image Processing Module(Hadoop 分散式影像處理模組)；另外在雲端影像處理與控制伺服器上實作了 Web Application 提供使用者透過網路觀看處理的結果。

此監視系統目前以單一 CCD 攝影機連結雲端，使用 Hadoop 分散式運算處理影像分析，由於 Hadoop 的 MapReduce 是一個分散式程式框架，具有可以提供大量資料的分散式運算環境，讓服務開發者可以很簡單的撰寫程式，利用大量的運算資源，加速處理龐大的資料量，當雲端基礎建設完備之後，可以處理較多個攝影機來援，因此，可以未來可以做進一步雲端伺服器的能力測試，量測出在單位雲端伺服器之下可以處理的攝影機數量，甚者，當未來攝影機的解析度提升之後，可以達到同樣的效果

最後本研究發現 MapReduce 固然在批次巨量資料的表現上面非常優異，但是如果達成即時分析，可能需要額外調整參數才能達到目標，因為原本的 MapReduce 模型就是假設所有必要的檔案都已經備齊在 HDFS 上面，每次執行 MapReduce 工作都會產生的大量的 I/O 行為，加上 Hadoop 本身有防止執行錯誤的機制，有時候會無法預期何時預算會結束，這與即時分析的限制還有一段距離。而因為人臉辨識只需要分析當前的影像，因此本研究未來規劃採用新版本的 MapReduce，把所有運算都集中在記憶體當中，避免大量的 I/O 讀取，並希望未來這套系統可以實際運用在社會當中，會整體人類貢獻一份心力。

陸、參考文獻

- [1] 張英傑，“智慧型監視系統之多人人臉即時循序偵測及追蹤”，崑山科技大學電機工程系碩士論文，2006。
- [2] P.Goolkasian, “Processing Visual-Stimuli Inside and Outside the Focus Fattention”, Bulletin of the Psychonomic Society, vol. 29,1991,pp.510-515.
- [3] 蔡博智，“影像追蹤方法應用在監控系統之研究”，中原大學機械工程學系碩士論文，2002。
- [4] 黃敏峰，“人臉追蹤法應用於監視系統之研究”，成功大學電機工程學系碩士論文，2003。
- [5] NovQinlu He,Zhanhuai Li’Xiao Zhang, “Study on Cloud Storage System based on Distributed Storage Systems”, Computational and Information Sciences(ICCIS), 17-19 Dec 2010, pp. 1332 – 1335.
- [6] Kevin D.Bowers,Ari Juels, and Alina Oprea., “HAIL: A HighAvailability and Integrity Layer for Cloud Storage”, Computer and communications security (CCS), Nov 2009, pp.



187-198.

- [7] Mingyue Luo, Gang Liu, “Distributed log information processing with Map-Reduce: A case study from raw data to final models”, Information Theory and Information Security(ICITIS), 17-19 Dec. 2010, pp.1143-1146.
- [8] Afrati, Foto N. Ullman, Jeffrey D., “Optimizing Multiway Joins in a Map-Reduce Environment”, IEEE Transactions on Knowledge and Data Engineering, vol. 23, Issue. 9, Sept. 2011, pp. 1282-1298
- [9] Chen Zhang, De Sterck, H., “CloudBATCH: A Batch Job Queuing System on Clouds with Hadoop and HBase”, Cloud Computing Technology and Science (CloudCom), Nov. 30 2010-Dec. 3 2010, pp.368-375.
- [10] Jorda Polo, David Carrera, Yolanda Becerra, Malgorzata Steinder, and Ian Whalley., “Performance-driven task co-scheduling for mapreduce environments”, Network Operations and Management Symposium(NOMS), 19-23 Apr 2010, pp.373 –380.
- [11] Thomas Sandholm and Kevin Lai., “Dynamic proportional share scheduling in hadoop”, Job Scheduling Strategies For Parallel Processing (JSSPP), vol. 6253, 2010, pp. 110-131.
- [12] 廖紹綱, “數位影像處理”, 台灣培生教育, 2004。
- [13] L. G. Shapiro, G. C. Stockman, Computer Vision, pp.65-68, Prentice –Hall, NJ, 2001.
- [14] 黃建邦, “彩色視覺技術於多人人臉辨識上的應用”, 崑山科技大學電機工程系碩士論文, 2005



Application of AOI Image Processing Technique on Cloud-based Face Detection

*Ming-Fang Wu *Chun-Jung Chen **Tien-Chi Chen
Hsu-Cheng Lin *Jing-Shian Huang

*Associate professor of Department of Electrical Engineering, Kun Shan University.

**Professor of Department of Electrical Engineering, Kun Shan University.

***Researcher of South Campus, Industrial Technology Research Institute.

****Postgraduate of Department of Electrical Engineering, Kun Shan University.

ABSTRACT

Traditional surveillance systems are mostly people need to sit staring at the screen in front of monitoring terminal. Once a long time for staring at the screen, attention will be relatively non-centralized, and it will cause the situation happened that even objects into the monitor screen will be difficult to find, and then cause the emergence of a lot of contingencies or social cases. Therefore this thesis proposed a cloud-based intelligence surveillance system for face detection. In this system, the face detecting function of remote monitoring system can be achieved through the high computing ability of the cloud to reduce the shortcomings of manual monitoring.

In face detection, this research uses the Apache software developed by the company Hadoop, the reason for using this software is that Hadoop belong to a distributed computing platform, which can effectively process a large quantity of data through the Map/Reduce parallel processing mode, and the computing result is more fast than the traditional operation way, so it is very helpful for the face image processing speed.

The purpose of this thesis is to develop a cloud-centric intelligent monitoring system. This cloud system can be divided into three parts. The first part is the CCD(Charge-Coupled-Device) camera control side, which is responsible for transmitting the image data. The second part is the image processing end, which is responsible for the image data computing of face detection, and the treated face data will be remained on the server. Finally, the third part is the network application server, which is used to display the face detecting results.

Keywords: Image processing, Face detection, Cloud-based, Surveillance system, Hadoop

