

Classiface: Real-time Face Recognition Based on Multi-Task Convolution Neural Network

Yi-Ki Wang, Yu-Xiang Zheng, Chow-Sing Lin *

Department of Computer Science and Information Engineering
National University of Tainan

Abstract

Facial recognition technology has received extensive attention and massively developed in recent years. Facial recognizing with a traditional feature extractor, such as Local Binary Pattern (LBP), leads to a strong demand for massive training data. As data increase, the size of training model become even bigger which might more easily cause the drop on the effect of loading system. Recently, the breakthrough of face recognition application development may refer to the evolution of deep learning algorithms. Therefore, we propose the “Classiface” face recognition that includes data processing, face detection, and recognition. We collect the training data straightly from videos so that we can process the samples more efficiently in less time. We also use data augmentation technology to increase the number of samples and to make CNN even more robust. We adopt MTCNN which detects human faces fast and accurately. We also utilize the CNN to automatically extract features from images and use fully connected neural network for face classifying. At last, the experimental results show that our proposed method can achieve average 98% accuracy of face classification in real-time without training more than 3000 samples.

Keywords: Deep Learning 、 CNN 、 Preprocess 、 Face Detection 、 Face Recognition



* Corresponding author: mikelin@mail.nutn.edu.tw

DOI : 10.3966/222344892018040801002

Classiface: 基於多工卷積神經網路之即時人臉辨識

王義凱、鄭宇翔、林朝興*

國立臺南大學 資訊工程學系

摘要

人臉辨識技術於近年來快速發展且逐漸受到重視。若以傳統的特徵擷取方法達到人臉辨識，例如：局部二值模式，需蒐集大量訓練資料，且其訓練模型大小容易受資料量影響而變大，使該人臉辨識系統載入電腦時間延長，影響使用體驗。近年來，深度學習演算法的突破，漸漸發展出基於深度學習的人臉辨識，因此本研究提出「Classiface」人臉辨識方法，其主要包含資料預處理、人臉偵測、人臉辨識。預處理方面可細分為訓練與測試神經網路所需資料收集及數據增強方法，我們透過攝影機錄影的方式蒐集大量人臉資料，再以人臉偵測方法擷取影片中每一幀影像之人臉，並使用數據增強方法，擴增資料量及使訓練資料多樣化，提升神經網路的泛化性；人臉偵測部分，我們採用快速且錯誤率低的MTCNN方法；辨識方面，透過卷積神經網路，自動提取訓練資料特徵，並建構全連接神經網路，實現人臉分類器。最後，我們以3人且每人3000張人臉資料訓練神經網路，達到98%以上的辨識準確率，並能以Real-time進行人臉辨識。

關鍵詞：深度學習、卷積神經網路、預處理、人臉偵測、人臉辨識



一、緒論

隨著科技進步，硬體設備的運算能力大幅提升，使用人力的方式辨識人物身分，已不符合現今成本效益，為追求方便、迅速、降低人力成本，逐漸發展出人臉辨識技術。近年來，市面上常見的人臉辨識通常都藉由深度相機、紅外線等方式取得人臉的深度，進而取得精準的辨識結果，但以上方式的運算量極大，導致鮮少使用於即時系統上，通常都應用於保安系統等不須刻意追求辨識速度的科技上。更因為深度學習演算法技術的突破，使計算機視覺領域發展迅速，其中知名影像識別競賽 ImageNet 於 2012 年以後之參賽團隊，皆以深度學習方式達到極高之辨識準確率。此外，我們亦受到 Tensorflow 官方 MNIST 手寫辨識[1]的啟發，聯想到其理念亦可應用於人臉辨識，因此對此方面產生研究的興趣。本研究之人臉辨識方法將針對訓練模型大小、人臉辨識運算耗時的問題進行改善，透過深度學習方式，降低人臉定位、特徵偵測、光影變化等易造成辨識系統誤判之影響因素。最終目的是使此人臉辨識系統能快速的載入電腦並以 Real-time 方式進行辨識，且辨識成功率高於傳統機器學習方法，例如：Haar Feature-based Cascade Classifier、Histogram of Oriented Gradients。

根據[2]提及人臉辨識技術的困難點，例如：人臉定位、特徵偵測、光影影響、人臉方向影響，由於深度學習是以神經網路方式自動尋找圖片特徵，再加上充足的訓練資料，即可將以上問題的困難度與影響降低，且根據[3]的實驗結果顯示，使用深度學習方法處理圖片分類問題，優於使用傳統的機器學習方法。

訓練資料預處理方面，[4]提及許多數據增強(Data Augmentation)方法，根據[5]的驗證，其對於神經網路的訓練無疑是一大幫助，提升神經網路的泛化性以及降低過擬合(Overfitting)的現象。主要是隨機調整訓練資料的亮度(Brightness)、對比(Contrast)、旋轉(Rotation)、翻轉(Flip)、縮放(Zoom)、噪音(Noise)、裁切(Crop)、顏色(Color)，產生更多的數據資料，以提高訓練資料的多樣性。

人臉辨識方面，傳統機器學習方法—局部二值(Local Binary Patterns, LBP)為計算機視覺中的特徵提取方法，適合應用於紋理分類，該方法介紹由 Ojala et.提出[6]，其原理如圖 1 所示，透過各像素與其周圍像素之比較，進而取得該像素之特徵。Rahim et.[7]亦透過 LBP 方法實作人臉辨識，並達到了 99%的準確率，應證了此方法用於人臉辨識的可行性。但透過此方法訓練後的模型(Model)大小，會根據其訓練資料量提升而逐漸增加，最終則導致使用辨識系統時的載入電腦時間大增，嚴重影響使用體驗。

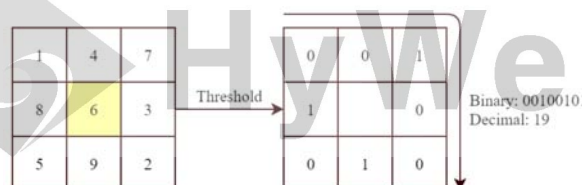


圖 1 LBP 特徵提取法示意圖

目前亦有使用深度學習方法進行人臉辨識，如 Google 團隊提出的 FaceNet[8]，將人臉圖片輸入至神經網路轉換為歐幾里得空間(Euclidean Space)之訊息，而歐式空間中的距離代表人臉圖片的相關性，即可透過此資訊達到人臉辨識，如所示圖 2 所示，兩張人臉圖片之間的值為經過計算之歐式距離，數值越小表示兩張圖片相似度越高。由於此方法之圖片運算時間稍長，較難達到 Real-time 的效果。

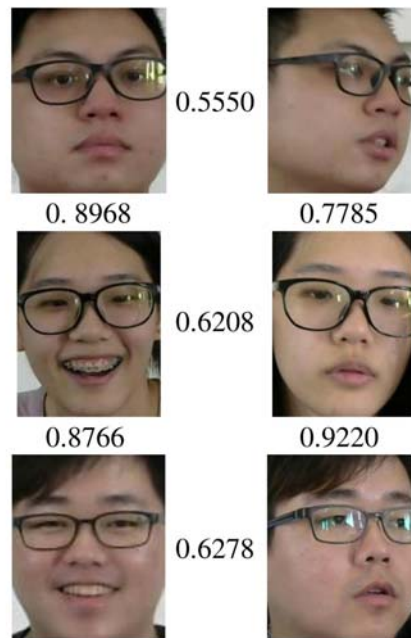


圖 2 FaceNet 歐式距離比較示意圖

因此，本研究將使用有別於以往固定式演算法的機器學習，採用深度學習的方式實作人臉辨識，透過訓練神經網路權重，由於其訓練模型大小主要根據神經網路結構而改變，於大量訓練資料的情況下，相較於局部二值方法，能夠有效減少訓練模型大小，降低辨識系統的模型載入時間。此外，本研究有別於 FaceNet，使用分類的方式，建構較少層的神經網路，以減少運算量，降低圖片運算耗時，達到 Real-time 的人臉辨識效果。

二、 Classiface

2.1. 系統架構

Classiface 主要分為兩大流程，如圖 3 所示，其中之一為「訓練人臉辨識神經網路」，首先載入 MTCNN 人臉偵測系統蒐集訓練資料，再將資料進行數據增強，最後輸入至人臉辨識神經網路進行訓練，並保存訓練後的模型；另一流程為「開始辨識已訓練之人臉」，最初亦先載入 MTCNN 人臉偵測系統擷取人臉影像，再將其人臉影像輸入至人臉辨識神經網路中，最終取得人臉分類結果，達到辨識效果。

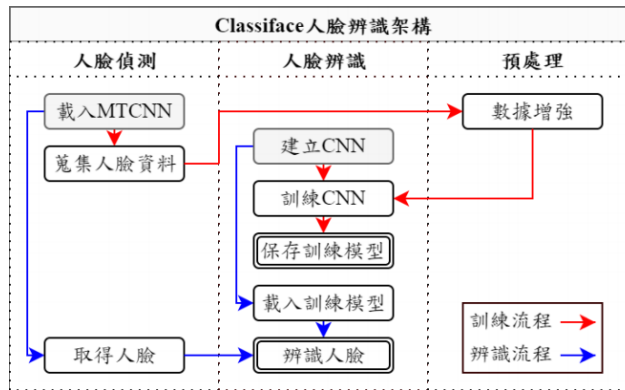


圖 3 Classiface 人臉辨識架構圖

2.2. 訓練資料預處理

我們先透過攝影機錄影的方式人臉之影片，後續再使用 MTCNN 人臉偵測方法，從人臉影片中的每一幀取得人臉，不僅減少蒐集資料的時間，還能準確且快速的取得影片中之人臉。

根據[4]提及的數據增強方法，能有效增加訓練樣本或是提升樣本的多樣性，因此本研究使用同樣的數據增強方法，將人臉資料進行隨機亮度與對比的調整，結果如圖 4 所示。



圖 4 人臉資料隨機調整亮度對比結果

由於數據增強需耗費大量 CPU 資源，且不同函式庫的效率也不盡相同。因此，使用 OpenCV、PIL、自行設計的數據增強方法進行測試，處理 10000 張解析度為 128x128 的影像，其中自行設計數據增強法是以遞迴方式遍歷影像中的每個像素，對該像素值進行調整。以上方法耗時比較與多執行緒下的表現如表 2 所示，測試所使用的 CPU 為 i7-4790。

表 1 數據增強方法與執行緒數目耗時比較

方法	Single-thread	8-threads
OpenCV	12 秒	3 秒
PIL	12 秒	6 秒
遞迴	250 秒	247 秒

根據測試結果，使用 OpenCV 函式庫搭配多執行緒進行數據增強，明顯比其它方法更有效率。因此，本研究採用 OpenCV 函式庫協助處理人臉資料。

2.3. 人臉偵測

由於現實場景中，大量且複雜的光影變化，導致人臉偵測錯誤率提高。因此，必須選擇對環境影響容忍度高的方法來進行人臉偵測，本研究以 OpenCV、dlib、MTCNN 人臉偵測方法來互相比較其準確率與運算速度，並分析其中的優缺點。

OpenCV 人臉偵測方法透過哈爾特徵(Haar Feature-based Cascade Classifier)方法[9]進行人臉偵測，其優點：運算速度極快；缺點：錯誤率高。偵測準確率容易受其訓練資料或人臉偵測當下環境之光影變化所影響，較適合使用於簡單且變化少的環境。

Dlib 人臉偵測方法透過 HOG-based(Histogram of Oriented Gradients)方法[10]的進行人臉偵測。其優點：準確度高；缺點：運算速度慢。此方法於現實生活中使用時，不易受到環境因素所干擾，且較能正確的標記出人臉，但若希望以 Real-time 的條件下進行人臉偵測，必須使用額外的技巧或適當的硬體設備來輔助。

Multi-task CNN(MTCNN) [11]人臉偵測方法使用卷積神經網路進行人臉偵測，透過三階級聯(Three-stage Cascaded)的方式，其特點為神經網路結構從簡單到複雜、閾值從寬鬆到嚴格，於最初階段排除大量不符合人臉特徵之資料，留下少量的資料再輸入更複雜且嚴格的神經網路，藉此大幅降低設備運算量，減少運算所需時間。若 MTCNN 經過適當事前訓練，其優點為：準確度高、運算速度快；缺點：使用此方法前，事先安裝軟體過程冗雜。實際使用此方法進行人臉偵測時，建議搭配 GPU 等硬體設備，以獲得良好的表現。而 MTCNN 人臉偵測方法網路架構如圖 5 所示，其主要步驟可分為以下三個階段：

- 第一階段：P-Net(Proposal Net)，利用 FCN(Fully Convolutional Network)檢測圖像，並透過 Bounding-box Regression 與 NMS(Non-maximum Suppression)進行校正
- 第二階段：R-Net(Refine Net)，取得第一階段的輸出，再透過 Bounding-box Regression 與 NMS 校正
- 第三階段：O-Net(Output Net)，取得第二階段的輸出，除了使用 Bounding-box Regression 與 NMS 校正，還加入五個人臉關鍵點作為額外限制



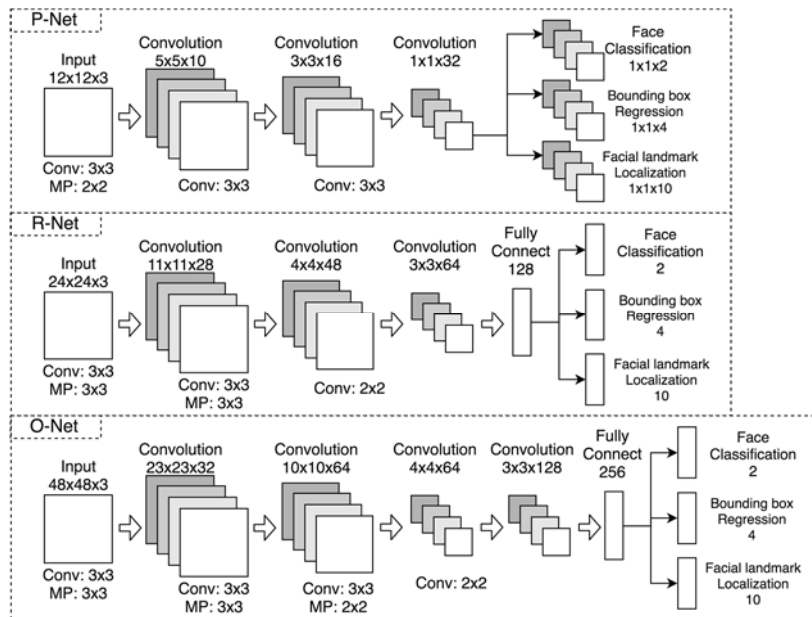


圖 5 MTCNN 神經網路架構

我們事先以錄製單一受測者之臉部於鏡頭可視範圍內隨機移動之影片，其包含 10000 個影格(Frame)且解析度為 1620×1080，再使用以上提及之人臉偵測方法進行偵測，其耗時比較如表 2 所示，初步了解到各方法對於單人影像之運算耗時。

表 2 偵測人臉 10000 次耗時比較

方法	耗時
OpenCV	約 10 分鐘
dlib	約 60 分鐘
MTCNN	約 20 分鐘

為了進一步了解以上三種人臉偵測方法處理包含多人臉影像的耗時關係，我們輸入包含 33 人且解析度為 1620×1080 的影像，各人臉偵測方法依據輸入的影像數量所產生之運算耗時關係如圖 6 所示。dlib 方法耗時依偵測圖片數量趨近線性成長；由於 MTCNN 方法是透過 GPU 輔助運算，耗時的成長依偵測圖片數量逐漸趨緩；而 OpenCV 方法則是使用多執行緒優化，使其擁有極少耗時。

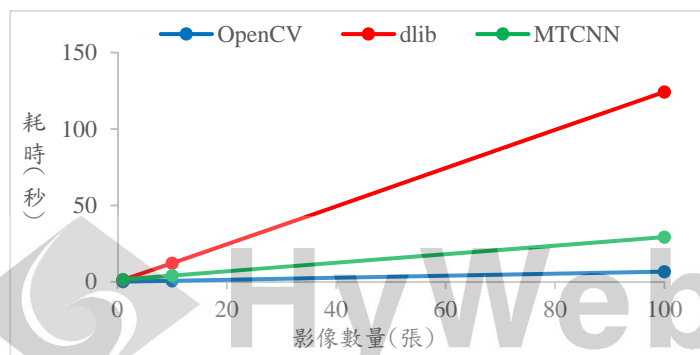


圖 6 人臉偵測方法耗時與照片數量關係

最後則是使用三種人臉偵測方法比較其偵測準確率，同樣輸入包含 33 人且場景稍微複雜的影像，較能符合現實應用。若偵測方法無錯標人臉我們以無誤判表示，其結果

如圖 7 所示。OpenCV 偵測出 16 人，無誤判；dlib 偵測出 32 人，無誤判；MTCNN 偵測出 32 人，無誤判。由於偵測方法、訓練數據不同，造成以上方法偵測結果不盡相同。由此測試得出，OpenCV 偵測準確率遜色許多，而 dlib 及 MTCNN 偵測準確率處於伯仲之間，若導入上述耗時比較結果，MTCNN 偵測方法則略勝一籌。



圖 7 人臉偵測方法結果

根據以上實驗結果，本研究採用 MTCNN 方法進行人臉偵測，並使用 GitHub[12]上由作者：Davidsandberg 基於 Tensorflow 實現此功能的開源程式碼。

2.4. 人臉辨識

人臉辨識步驟可拆分為建構與訓練神經網路，最後則進行辨識。由於本研究的人臉辨識方法與深度學習手寫辨識理念相似，皆從圖片中尋找特徵，以達到辨識的效果。因此建構出人臉辨識神經網路，如圖 8 所示，其架構可根據訓練資料量及欲辨識之人臉數量進行縮減或擴充，以得到更好的人臉辨識結果。

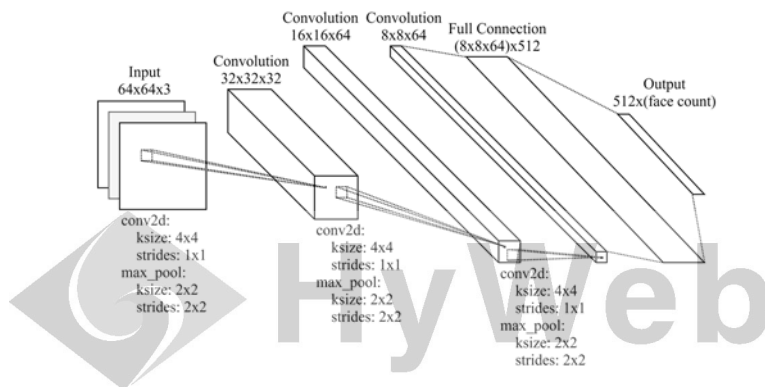


圖 8 人臉辨識神經網路理論架構圖

該神經網路包含輸入層、三個卷積層、三個池化層、一個全連接層、輸出層。其中，卷積層的滑窗(Filter)長寬皆為 4×4 ，移動步長(Strides)皆為 1×1 ；池化層的滑窗長寬皆為 2×2 ，移動步長皆為 2×2 。

輸入圖片後，經過三次特徵提取與維度降階，使圖片縮小並往第三維度擴充，大量降低資料運算量。將卷積層輸出的三維資料扁平化(Flatten)，作為全連接層之輸入，並開始進行人臉分類，最後於輸出層得到辨識結果。

除此之外，根據[13]對 Dropout 的分析，藉由導入 Dropout，降低神經網路過擬合的情況，以提高辨識準確率。本研究於池化層後以及全連接層皆有使用 Dropout。

訓練該神經網路之前，必須先定義損失函數(Loss Function)，其用來評估訓練模型之預測值與真實值的差距，當 Loss 值越小，表示此神經網路的強健性(Robustness)越好，受訓練資料、環境干擾所影響的程度越小。此外，本研究訓練神經網路時，當 Loss 值達到穩定狀態，即表示神經網路辨識能力已趨近極限，必須透過更多訓練資料或是修改網路結構才得以增加其辨識能力。本研究使用在圖像分類方面有良好表現的 Softmax Cross Entropy 作為損失函數，並取得神經網路輸出層(Output Layer)的張量(Tensor)作為其輸入，並根據 Ycszen[14]對於優化器優缺點的比較，採用 Adam[15]優化器將 Loss 值最小化。且每訓練 100 步(Batch)，輸入測試用人臉資料，並將其標記(Label)與辨識結果做比較，計算出當前神經網路的辨識準確率。最後則設定輸入訓練圖片的批量大小(Batch Size)，即可開始訓練神經網路。

以下使用兩人各 10000 張訓練資料，並使用 Sklearn 提供的方法隨機取出其中 1000 張影像作為測試資料，其餘為訓練資料進行測試。結果如圖 9、圖 10 所示，當訓練步數到達 600 步(Batch)時，已達到預期準確率 99.5%，耗時大約 30 秒，且 Loss 的值也趨近於穩定狀態。

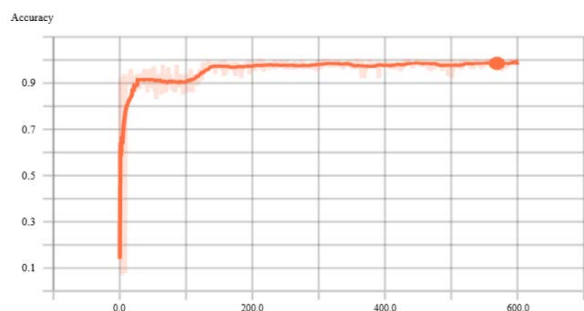


圖 9 人臉辨識訓練時的準確率(X 軸: 步數, Y 軸: 機率)

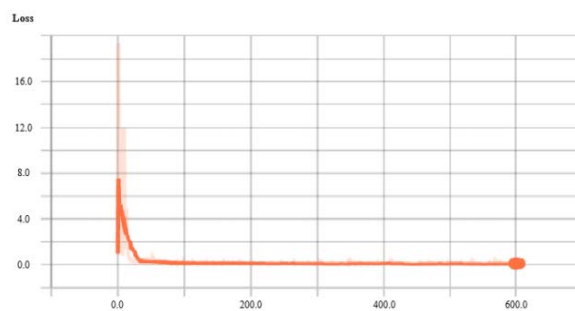


圖 10 人臉辨識訓練時的 Loss(X 軸: 步數, Y 軸: Loss)

訓練結束後利用 Tensorflow API，保存神經網路權重，待後續人臉辨識或繼續訓練神經網路使用。欲改變人臉分類數目，則必須重新訓練此神經網路。

本研究使用 Webcam 進行 Real-time 的人臉辨識測試，結果如圖 11 所示，其中每一張圖片代表一個影格的辨識結果，左邊的人標記為「me」，右邊的人標記為「brother」。



圖 11 臉辨識成果展示圖

使用此系統進行人臉辨識實驗，顯示卡規格為 GTX 1060 3G DDR5，實驗辨識 10000 張影像，平均每張辨識時間為 0.0403 秒，平均辨識幀率(FPS)約為 25 張影像，而人眼視覺暫留(Persistence of vision)時間約為 1/16 秒，此人臉辨識幀率大於 16 張影像，因此辨識結果可視為流暢。若使用更高規格的硬體設備或優化神經網路，效果肯定更加理想。

2.5. 系統討論

根據以上針對訓練資料預處理、人臉偵測、辨識方法進行說明及測試，我們得出以下結論：(1)蒐集完人臉資料後，使用 OpenCV 函式庫協助預處理，透過隨機調整圖片亮度、對比，或隨機裁切圖片大小，以增加訓練資料量及多樣性，再將資料輸入至人臉辨識神經網路訓練；(2)人臉偵測部分採用 MTCNN 方法，以達到極高的偵測正

確率及極快的偵測速度蒐集人臉資料或是取得欲辨識之人臉資料；(3)人臉辨識部分則使用自行參考文獻設計的神經網路，並經過適當訓練，即可達到辨識效果。

三、系統實驗

由於 2.4 小節提及之網路結構，每人必須使用大量的訓練資料，才能使人臉辨識的準確率提高。因此，實驗時嘗試將神經網路的卷積層擴充至 4 層，並新增 1 層全連階層以可視化本實驗之人臉分類據類關係，架構如圖 12 所示，成功將每個人的訓練資料量從 10000 減少至 3000 張圖片，且與原始網路架構辨識效果接近。以此推測，若有系統性的加深神經網路，或許能再降低訓練資料量。由於資料量減少，造成訓練時的人臉辨識準確率與 Loss 波動較為明顯，此問題將會於以下實驗提及。

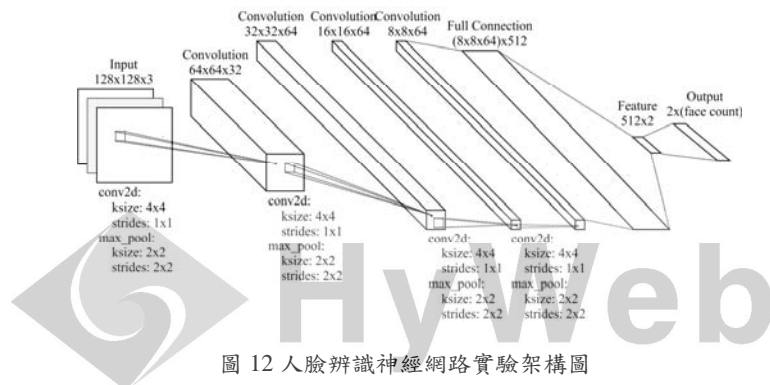


圖 12 人臉辨識神經網路實驗架構圖

本研究以每人 1000、2000、3000 張照片的資料集進行測試，詳細配置如表 3 所示。

表 3 實驗資料詳細配置(單位：張)

每人資料	訓練總資料	測試總資料	總資料量
1000	2850	150	3000
2000	5700	300	6000
3000	8550	450	9000

根據以上資料量配置進行實驗，三種資料量的辨識準確率比較，以及與訓練步數的關係分別如圖 13 所示，X 軸為訓練步數，Y 軸為辨識準確率，橘色、藍色、紫色曲線分別表示每人使用 1000、2000、3000 張人臉資料進行訓練。

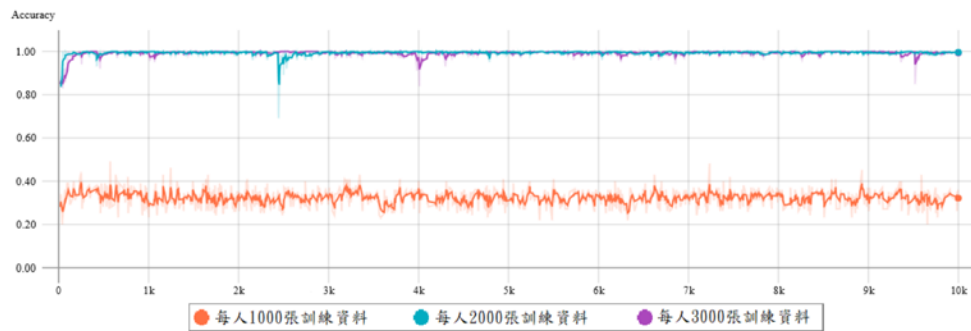


圖 13 資料量實驗辨識準確率比較

根據以上訓練 10000 步的資料量實驗，若每人進行訓練的資料量為 1000 張照片，辨識準確率不及 50%。欲使辨識準確率達到 90% 以上，每人使用的訓練數據量至少為 2000 張照片。由此可知，驗數據量越大，訓練時的辨識率震盪越不明顯；而訓練步數越多，辨識準確率波動則更趨穩定。

為了進一步觀察此人臉辨識之神經網路分類狀況，我們使用 3 人且每人 3000 張訓練資料，訓練 10000 步該神經網路後，其分類可視化結果可透過全連接層顯示。我們輸入測試資料，並將該全連接層輸出之二維特徵映射至座標平面，結果如圖 14 所示，其中不同人之特徵點以不同顏色表示，可以發現不同類別間的距離小且同一類別間的距離大，這可能於現實使用時，誤判機率增加，欲修正此問題我們必須從神經網路的損失函數著手，後續將透過 Center Loss[16]的輔助，減少類內距並增加類間距，以提升人臉分類準確率。

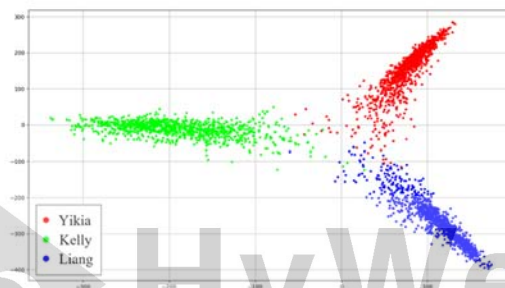


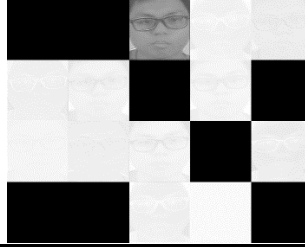

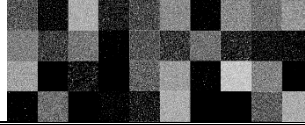

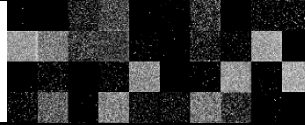

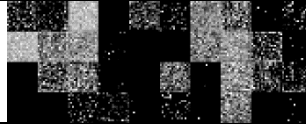
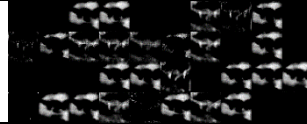


圖 14 人臉分類可視化結果

除此之外，我們更可以從神經網路之卷積層取得其所提取之人臉圖片特徵，本實驗包含 4 層卷積層，以下將依序顯示每層卷積神經網路所提取之特徵，並分為該神經網路是否經過訓練，結果如表 4 所示，由於篇幅關係，此處僅顯示部分濾波器提取之特徵影

像。其中，第一層卷積包含 32 個濾波器，從輸入之影像提取 32 個特徵影像，而第二至第四層卷積則包含 64 個濾波器，皆從前一層卷積層之輸出提取 64 個特徵影像，且我們可以發現，神經網路的訓練與否，亦大幅影響其所提取之特徵。

表 4 卷積網路提取之特徵示意表

Before Training	After Training
Input	
	
Conv 1	
	
Conv 2	
	
Conv 3	
	
Conv 4	
	

我們亦使用「The Extended Yale Face Database B」人臉資料庫[17]進行分類數量實驗，該資料庫包含 28 人且每人 576 張人臉圖像，每人取其中 500 張圖像進行訓練，測試該神經網路對於不同人數所訓練之人臉辨識準確率，結果如圖 15 所示，X 軸為訓練步數，Y 軸為辨識準確率，橘色、咖啡色、粉紅色、藍色曲線分別表示使用包含 5、10、15、20 人之入臉資料進行訓練。

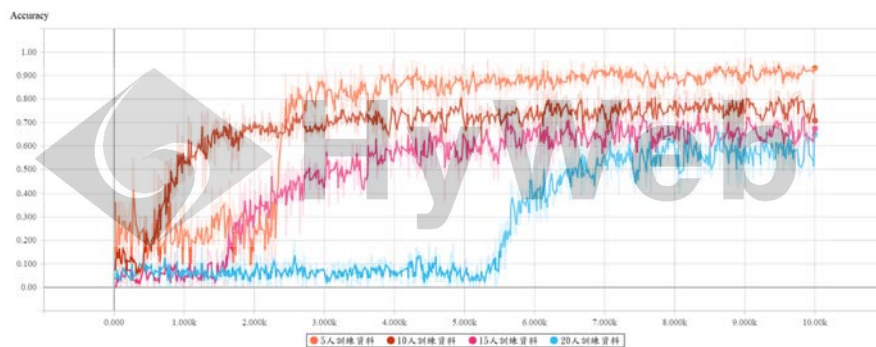


圖 15 分類數量實驗準確率

根據以上訓練 10000 步的分類數量實驗，欲辨識之人數越多，其訓練後之辨識準確率越低。相較之下，[18]使用相同人臉資料庫所有資料，透過 LC-LBP(Logically Concatenated LBP)方法實現人臉辨識，雖然辨識準確率達到 100%，卻忽略其辨識系統效能，例如：訓練模型大小、人臉辨識速度等。而我們的方法雖犧牲些許辨識準確率，但換來了更彈性的人臉辨識系統，其特點包含：訓練模型小、辨識速度快，甚至能夠根據欲辨識之人臉數量進行神經網路架構及參數的調整。其中，欲使人臉辨識速度加快，可透過減少神經網路層數、縮小卷積核等方式達成；欲使人臉辨識準確率增加，添加卷積層則是不二法門。而辨識速度及準確率本是互相影響，因此，欲透過本研究方法建構辨識多人之人臉辨識系統，除了加深神經網路提高辨識準確率外，更需調校網路參數、架構，甚至使用額外方式以提升人臉辨識速度。

最後，本研究亦與現有人臉辨識技術 LBP、FaceNet 進行比較，我們針對人臉辨識的訓練模型大小與即時性方面深入探討，結果如表 5 所示。由於本研究使用淺層神經網路進行特徵提取，使其擁有較小的訓練模型與特徵運算時間，進而降低此人臉辨識系統載入時間並提升人臉辨識即時性。

表 5 人臉辨識方法比較

	Classiface	LBP	FaceNet
Model Size	26MB	300MB	110MB
Load Time	<5s	>10s	>30s
Compute time	0.04s	0.16s	2s
FPS	25	6	0.5

四、結論

雖然神經網路透過持續訓練可使辨識準確度提高或更趨穩定，但只以同一個資料集重複訓練多次，亦可能產生過擬合的問題，於現實生活中使用時，容易造成辨識錯誤率提升，因此建議以更多資料進行訓練。

本研究所描述之人臉辨識系統，目前採用每人 3000 張資料訓練神經網路，並使用 Webcam 進行測試，多數情況皆可正確辨識，初估準確率達到 90% 以上；少數情況，例如：Webcam 曝光嚴重，即可能影響到此系統的辨識準確率，但選用搭載自動調整曝光功能的鏡頭，即可解決此問題，而日後將持續優化此辨識方法。最後的目標是希望能夠將此系統的效能提升，並減少資料蒐集或預處理的時間，降低使用此系統的困難度。

五、致謝

本研究感謝科技部「大專學生研究計畫」(MOST 106-2813-C-024-024-E)以及科技部「專題研究計畫」(MOST106-2221-E-024-003-)支持。



六、參考文獻

- [1] “Deep MNIST for Experts,” 2017. [Online]. Available: https://www.tensorflow.org/get_started/mnist/pros#deep-mnist-for-experts
- [2] M.A. Turk, A.P. Pentland, “Face recognition using eigenfaces”, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.
- [3] “Image Classification in 5 Methods,” 2017. [Online]. Available: <https://medium.com/towards-data-science/image-classification-in-5-methods83742aeb3645>
- [4] 黃文堅 and 唐源, Tensorflow 實戰. 北京市: 電子工業出版社, 2017.
- [5] “前饋神經網絡泛化性能能力的系統分析,” 2017. [Online]. Available: http://123.57.41.99/jweb_xtgcllysj/CN/article/downloadArticleFile.do?attachType=PDF&id=109012.
- [6] T.Ojala, “Pergamon A COMPARATIVE STUDY OF TEXTURE MEASURES WITH CLASSIFICATION BASED ON,” vol. 29, no. 1, 1996.
- [7] T.Wahid, “Face Recognition using Local Binary Patterns (LBP),” vol. 13, no. 4, 2013.
- [8] Florian Schroff, Dmitry Kalenichenko, James Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering”, In Proc. CVPR, 2015
- [9] P.Viola, O. M.Way, and M. J.Jones, “Robust Real-Time Face Detection,” vol. 57, no.2, pp. 137–154, 2004.
- [10] N.Dalal and B.Triggs, “Histograms of Oriented Gradients for Human Detection.”, In CVPR, 2005.
- [11] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Yu Qiao, “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”, *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.
- [12] “Face alignment using MTCNN,” 2017. [Online]. Available: <https://github.com/davidsandberg/facenet>
- [13] Haibing Wu, Xiaodong Gu, “Towards Dropout Training for Convolutional Neural Networks”, *Neural Networks 71*, pp. 1–10, 2015
- [14] “深度學習最全優化方法總結比較,” 2017. [Online]. Available: <https://zhuanlan.zhihu.com/p/22252270>
- [15] Diederik Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization”, *ICLR*, 2015.
- [16] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A Discriminative Feature Learning Approach for Deep Face Recognition,” *In European Conference on Computer Vision (ECCV)*, 2016.
- [17] Athinodoros Georghiades, Peter Belhumeur, and David Kriegman, “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pos,” *PAMI*, 2001.
- [18] Rinku Datta Rakshit, Subhas Chandra Nath & Dakshina Ranjan Kisku, “An improved local pattern descriptor for biometrics face encoding: a LC–LBP approach toward face identification,” *Journal of the Chinese Institute of Engineers*, 2017.