國立臺灣師範大學
資訊工程研究所碩士論文

指導教授：黃文吉　博士

Feedforward Neural Networks
於連續手勢辨識之研究

Continuous Hand Gesture Recognition By
Feedforward Neural Networks

研究生：朱晏呈 撰

中華民國 一〇八 年 六 月

摘要

本論文提出基於感測器的連續手勢辨識系統，使用者右手握著智慧手機做出動作，透過收集搭載在手機上的感測器－三軸陀螺儀與三軸加速度計產生的訊號，總計六維度的資料來構成手勢。面對手勢這種時間序列資料，加上為解決連續手勢中找出切割點(Spotting)的問題，本論文中提出基於 Feedforward Neural Networks 建立出的深度學習模型，整合現行架構中已被證實能夠更加有效利用 Convolutional Neural Networks 的結構－ResNet、GoogLeNet 與 Inception-ResNet，將這些概念與 PairNet 做結合。

實驗中使用透過手機蒐集的11種手勢，在測試時一次會輸入含有1~4個手勢的資料，進到事先訓練好的類神經網路模型之中，再經由後處理得到辨識結果，而這樣的演算法則能處理傳統方法上無法有效解決的 Spotting 問題。另外，根據提出的模型 ResPairNet 在連續手勢上的辨識率，比 LSTM 高出7%以上的結果也可推得－Feedforward Neural Networks 在時間序列資料的處理上，比 Recurrent Neural Networks 更加強大、有效，將這些原本應用於影像領域的結構，套用到處理時間資料的問題上，能夠更進一步提升 Feedforward Neural Networks 得學習能力。

關鍵字－Feedforward Neural Network, Continuous Hand Gesture Recognition, Deep Learning, Human-Machine Interface

目錄

表目錄

圖目錄

第一章　簡介

本章將會對整個研究的基礎進行介紹。1-1 介紹人機介面的定義，手勢在這之中能扮演著什麼角色；1-2 說明本論文研究的核心，在連續手勢辨識上面臨著哪些挑戰，探討如何透過連續手勢辨識系統建立驗證系統，保障使用者的隱私；1-3 針對本論文提出的法則對連續手勢辨識上的改變，以及有哪些層面的創新進行說明。

1-1 研究背景

人機介面 (Human-Machine Interface, HMI) 使人能夠與機器進行溝通，人能夠透過介面向機器下達指令，而機器則是透過介面接收、並且回傳執行指令後的結果給使用者。其中，「手勢」作為乘載訊息的媒介有著便利及高效率的優點，在人機介面領域長時間受到關注。

手勢作為一種溝通語言，涉及到手指、手臂甚至是全身的動作，細微的變化都可能使手勢代表的意義改變，而同樣的手勢在不同文化中不一定有著相同的含義。連續手勢辨識 [1] 則是由使用者做出一段手勢後，交由機器辨識其中的含義，從專業學術到電子遊戲領域，都能看到相關的應用，例如：

- 手語辨識 [2, 3, 4]：使用者輸入手勢，讓機器轉換成相對應意義的語句。
- 互動遊戲 [5, 6]：透過人的雙手做為控制器，做出手勢來進行遊戲。

連續手勢辨識在日常生活中也隨處可見，例如：圖 1.中所示，組合多個手勢來與他人溝通、揮動遊戲手把來操作角色，或是以手勢經由智慧家電系統控制洗

衣機等，連續手勢扮演的角色可取代傳統上需要實體按鈕的操作，為使用者帶來便利。

圖 1. 連續手勢應用例子。從日常生活到學術研究領域，連續手勢被廣泛的應用。

1-2 研究目的

手勢是一種時間序列 (time-series) 資料，數據之間有時間上先後順序的關係，根據取得手勢的方式分為基於視覺 (Vision-based) 與基於感測器 (Sensor-based) 兩種。視覺介面是透過攝影機以照片或影片的方式，來記錄使用者做的手勢；感測器介面則是透過手套、手機等裝置，記錄使用者做手勢時產生的訊號。

由影像記錄的手勢資料比較大，獲取資料十分容易，但實際應用時反而會有侵犯使用者隱私的危險，畢竟影像能夠紀錄非常多的資訊，沒有人能接受在自己家中裝數十隻攝影機，只為能透過手勢操作東西；經感測器收集的手勢雖然使用上限制較多，但也只會收集到相關的資料，這樣的特性對隱私有更多的保障，另外感測器獲得的資料維度也比影像小，處理所需要的運算量較低。

連續手勢辨識上，最重要的是找到手勢與手勢之間的切割點 (Spotting)，將複數手勢切割成單一手勢再個別進行辨識，這問題對基於感測器收集的資料上是一

大挑戰。如圖 2.中所示，即使已知這串波形代表 4 個手勢的組合，也沒辦法輕易地找出某個時間點會對應到的手勢種類。

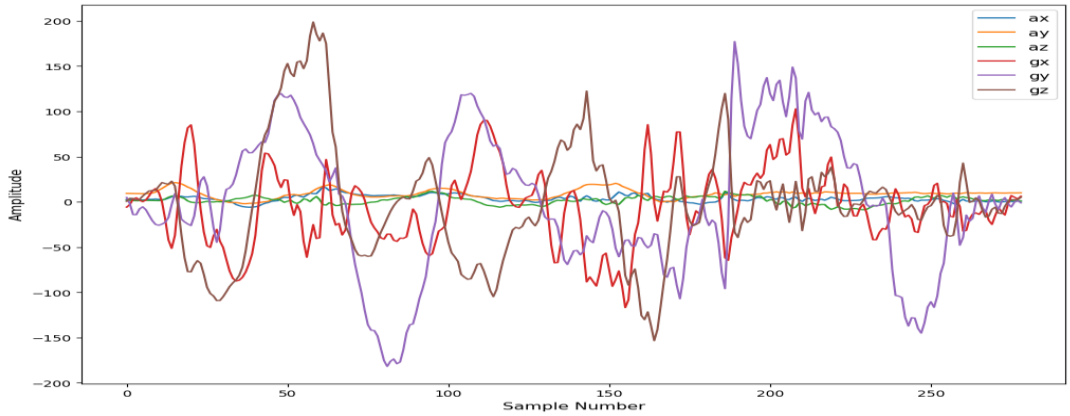

圖 2. 連續手勢資料之視覺化。

另一個重點，是透過實驗提出一套基於連續手勢的驗證系統，來保障使用智慧家電上的安全。本論文中採用智慧型手機來收集手勢，輕盈的設計原本是方便使用者隨身攜帶，但若今天沒有保護措施，只要任何人拿到這台手機就可以操縱已經連接上的裝置，試想這對個人隱私及居家安全會有多大的威脅。

智慧手機本身會對使用者進行身分識別，藉此使裝置能夠私人化，如圖 3.中提到的兩種方式為最常見。第一為透過輸入事先設定好的密碼，通常是由一組有限制長度的數字組成；第二則是要藉由智慧型手機的觸控螢幕，畫面上會顯示九個圓點，而使用者通過按住並拖曳在任意兩點之間形成線，最終會形成一連續線段組成的圖形，裝置便會將這當作使用者的密碼。

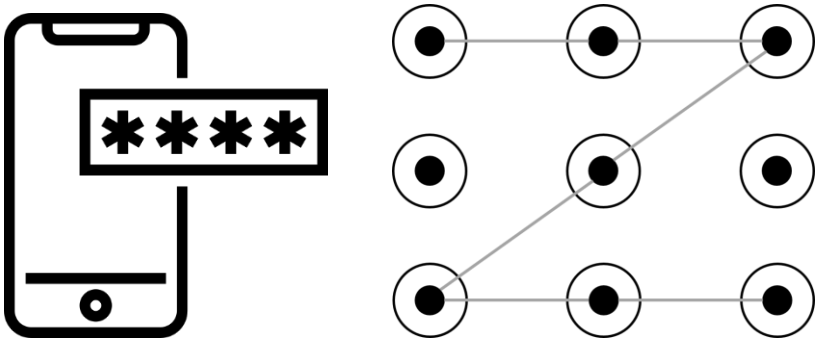

圖 3. 手機解鎖方式。左圖為文字密碼，右圖為圖形密碼

這兩套機制看似完整，仍舊容易被有心人士破解，例如：在使用者輸入時偷窺並記下內容、使用暴力破解法強硬突破、藉由手機內部儲存設計上的漏洞獲取密碼等，而這些方法能夠突破加密系統，便是起因於密碼本身難度不夠高，容易被他人複製的特性所導致。

本論文中提出以智慧型手機在三維空間中揮動，以過程中產生的連續手勢做為驗證系統的密碼，甚至期望能更進一步在使用者操作手機的過程中，每個一段時間主動認證使用者的身分，這樣的技術稱作主動式驗證 (Active Authentication, AA) [7, 8, 9, 10, 11, 12]，透過 AA 能降低使用者疏失導致的安全問題，並且隨著做的手勢長度拉長，他人想要複製這段密碼的難度也會隨之大幅提升，不論是想要透過暴力破解法或是在旁偷看學習法來破解都十分艱困，另外透過即時辨識使用者身分，這樣的設計也能避免透過裝置存取設計上的漏洞來破解，形成在各方面上都兼顧的驗證系統。

1-3 研究貢獻

考量到基於視覺介面得到手勢中含有太多不必要的資訊，本論文建立一個基於感測器的連續手勢辨識系統。透過 Samsung Galaxy S8 與 HTC ONE A9 這兩部行動裝置上搭載的三軸陀螺儀 (Three-axis Gyroscope) 和三軸加速度計 (Three-axis Accelerometer)，組合不同時間點連續產生的訊號作為連續手勢的輸入資料。

基於深度學習模型的法則，使得不必為再將連續手勢預先切割成數個單一手勢，省略掉額外處理直接交給模型去學習，透過訓練好的模型即可得到準確的連續手勢結果。另外，提出能夠穩定、有效解決連續手勢辨識問題的類神經網路模型，從中探討前饋式神經網路 (Feedforward Neural Network)在時間序列資料上的處理能耐，透過與其他現行著名法則的整合，甚至能夠遠比遞迴神經網路 (Recurrent Neural Network) 更強大。

第二章　基本理論介紹

本章將說明連續手勢辨識領域相關研究，及深度學習模型介紹。2-1 介紹連續手勢的分類；2-2 說明傳統上是如何處理連續手勢，並且需要克服哪些挑戰；2-3 介紹深度學習中的模型架構，比較前饋式神經網路和遞迴神經網路兩者的差異，並且探討這些法則在連續手勢辨識上的可能性。

2-1　連續手勢介紹

語言是人類彼此之間交流的媒介，最主要以說話的方式來進行溝通。相較於話語有著地域、文化上的限制，肢體語言有著更高的跨國際性，能更直接表達情緒，甚至是內心中真實的想法，而「手勢」在其中占有中要地位。廣義上的手勢泛指任何透過手部做出的動作，而狹義上手勢指的是帶有意義的動作，連續手勢則是組合這些有意義手勢而成。

蒐集手勢資料的方式主要分為兩種—基於視覺(Vision-based)和基於感測器(Sensor-based)系統，如圖4.所示，使用者透過這兩種不同的介面來輸入手勢，各自擁有不同面相上的優勢。

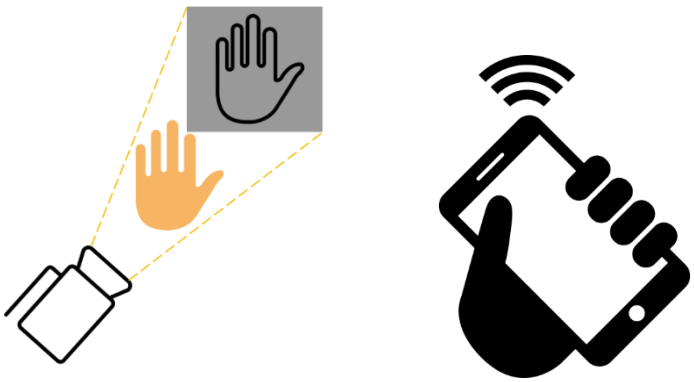

圖4. 連續手勢收集方式。左圖為基於視覺系統，右圖為基於感測器系統。

市面上販售有攝影功能的設備取得容易，現代社會中人手一支的手機普遍也都富有照相機，加上影像形式的資料也比較直觀，使得基於視覺系統的研究比較普及。接下來將介紹兩者相關的研究及差異：

I. 基於視覺系統 (Vision-based System) [13, 14, 15, 16, 17]

- 資料格式為平面、立體的圖片或者是影片，手勢記錄在影像中
- 由於影像不單只有手勢，連背景等其他資訊也會收進去，為解決這樣的問題，通常只能仰賴使用較簡單的背景，減少整張圖中不必要的訊息，或透過影像分割 (Segmentation) 將手從影像中分離出來。

II. 基於感測器系統 (Sensor-based System) [5, 18, 19, 20]

- 資料格式為一連串的數據，根據時間先後排序而成，這種資料形態稱為時間序列 (Time Series)。
- 根據不同的感測器類型，能夠以不同的資料來代表手勢特徵，例如：三軸陀螺儀 (Three-axis gyroscope)、三軸加速度計 (Three-axis accelerometer)、肌電圖 (Electromyography)。
- 根據手部動作產生的數據，不同於影像會有不必要的部分，只會記錄手勢的資訊。

其中還有一點，也就是若要將連續手勢用於智慧家電的控制上，需要考量到家庭的隱私問題，或許基於視覺的系統比較容易獲得手勢，但也因此容易對使用者的隱私造成侵害，例如：24小時不間斷地以數十隻的鏡頭拍攝來得到手勢，這

無疑會對日常生活產生莫大的壓力。在這問題上基於感測的系統就有很大的優勢，限制感測器只能在特定區域使用，加上感測器的特性，使他人很難從感測器產生的訊息推得其他資訊，這也是本論文中採用基於感測器介面，做為收集連續手勢的系統。

2-2 傳統處理連續手勢的方法

連續手勢辨識根據獲取手勢的過程，各自有不同的課題需要去解決。傳統上主要以分類 (Classification) 問題的角度去處理，傳統上藉由隱藏式馬可夫模型 (Hidden Markov Model) [15]、支援向量機 (Support Vector Machine) [21] 來處理。即便有上述的演算法則，還是會面臨到一些問題：

A. 需要將手勢從不必要的資料 (例如：雜訊、背景) 中切割出來。為此必須另外設計一套演算法來處理，例如：基於視覺的資料可透過利用膚色偵測找出手的位置 [15] ；基於感測器的資料則是利用閾值區分出手勢與非手勢 [18]，而這些前處理演算法若設計不當，將會連帶影響最終的結果。

B. 延續上一點提到的，傳統方法上無法只單獨使用一個演算法就能處理整個問題，而是將問題切割成數道小問題，每個小問題對應到不同的處理方法，這樣額外的負擔對研究上是一種阻礙，因為研究人員必須要顧及到所有細節，若有一個環節沒設計好，就會導致全盤皆輸的狀況。

在過去仍舊有許多研究人員提出相關的辨識系統，或許他們也在實驗當中取得某種程度上的成功，但無法建立一個穩定且高泛化性 (Generalization) 的法則，當要將這些成果實際用到現實生活中，便會遇到許多問題及挫折。

2-3 深度學習與連續手勢辨識

深度學習這項技術其實很早就被提出來，但在早期礙於硬體效能及尚未成熟的技術，使它發展一直受到限制，直到 30 多年前由 Yann LeCun 提出的 LeNet-5 [22]模型，為近代深度學習奠定了穩固的基礎。透過調整模型中的隱藏層，能使模型能夠面對不同難度的問題，相較以往的技術在彈性及正確率上大大提升，而其中的類神經網路模型在訓練階段時，會學習資料並從中找出特徵，也因此能降低對資料前處理的依賴性。

在網路上有非常多龐大的開源資料庫，而有許多研究也以此為評比開發模型的標準，例如：包含 1400 萬張圖片及超過 2 萬個種類的影像資料庫—ImageNet [23] ，根據該資料庫提出的的競賽—ImageNet Large Scale Visual Recognition Competition (ILSVRC) 主導了近年來卷積式神經網路在電腦視覺領域的發展，每年的第一名都會被公開發表作為新的神經網路架構，歷年得獎者都為深度學習領域帶來不小的變革，例如：開創了卷積神經網路大時代的 AlexNet [24] 就是 2012 年的第一名，本論文中提出的模型是以同樣出自 ILSVRC 的 GoogLeNet 與 ResNet 為理論基礎，進化而成的。

2-3-1　前饋神經網路與遞迴神經網路

在深度學習當中有許多不同的模型結構，通常是根據資料類型以及任務的目標來決定，遇到圖片、影片等使用卷積神經網路 (Convolutional Neural Network, CNN)，遇到文章、語音等使用遞迴神經網路 (Recurrent Neural Network, RNN)，而本論文中的手勢也是時間序列資料，使用 RNN 能學習到資料上的前後關係，除了主要使用的資料型態不同外，這兩者在結構上也有很大的不同。

RNN 會將運算完的結果，送到下一個時間點做為輸入，這樣的狀態可在圖 5. 中看到，橘色的隱藏層在每次運算後，不只有綠色的輸出，也會透過虛線傳送特徵做為接下來的輸入，也就是當前的輸出不只受輸入影響，也受前一個時間點的輸出影響，在訓練的過程中不單單只有資料本身的特徵，連時間上的先後順序都是訓練的資料。

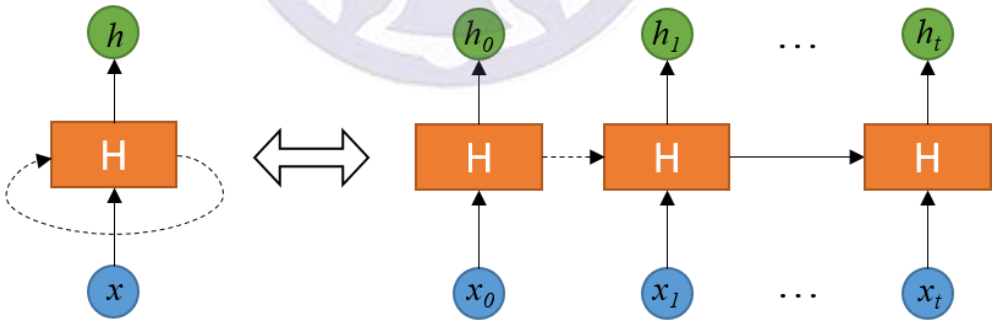

圖 5. Recurrent Neural Network 結構圖。

CNN 主要是透過數個不同的卷積核在資料上擷取特徵，在架構上與 RNN 不同，隱藏層的輸出不再會到下一個時間點做為輸入，每一層經過運算後的結果，只會往單一個方向傳遞，也就是輸入與輸出獨立，此模型結構被稱為前饋神經網路(Feedforward Neural Network, FNN) [25, 26]，結構見圖 6.所示。

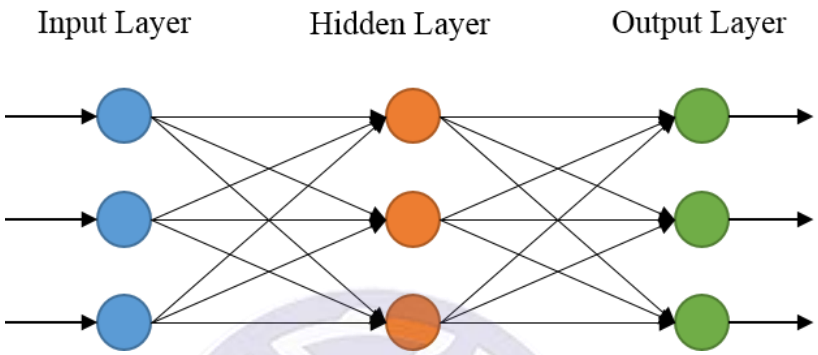

圖 6. Feedforward Neural Network 結構圖。

CNN 在空間上找特徵能力較強，透過滑動視窗 (Sliding Window) 將時間序列資料轉成「固定時間長度」的資料，這樣就可以用 CNN 來做學習及辨識，詳細方法會在後面章節中說明。

2-3-2　長短期記憶網路與雙向長短期記憶網路

RNN 能夠學習時間序列資料中時間的概念，但隨著資料的長度提升，它會在訓練過程中遇到梯度爆炸(Exploding gradient)或梯度消散(Vanishing gradient)的問題 [27, 28]，導致模型辨識能力驟降。為了解決這個問題科學家對 RNN 的結構進行調整，使其能夠處理更長的時間序列資料，而產生出長短期記憶網路 (Long-Short Term Memory Network, LSTM) [29, 30]。

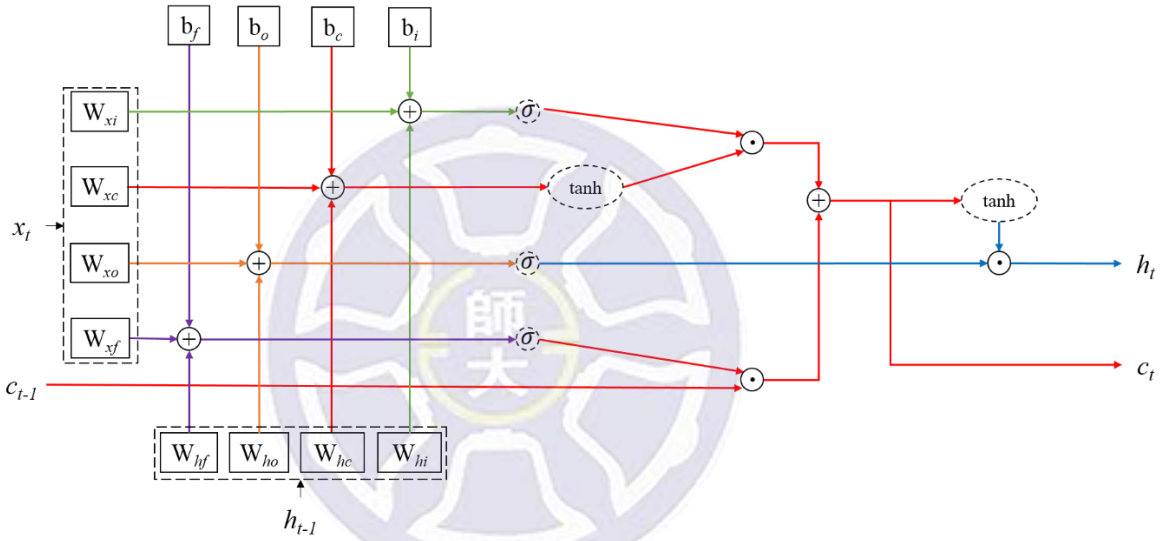

圖 7. Long-Short Term Memory 結構圖。

LSTM 由三個 Gate 和兩個 State 組成，結構如圖 7.所示，而這些單元扮演著不同的角色，各自做的運算如下：

Input Gate：$i_t = \sigma (x_t W_{xi} + h_{t-1} W_{hi} + b_i)$

Output Gate：$o_t = \sigma (x_t W_{xo} + h_{t-1} W_{ho} + b_o)$

Forget Gate：$f_t = \sigma (x_t W_{xf} + h_{t-1} W_{hf} + b_f)$

Cell State：$c_t = f_t * c_{t-1} + i_t * tanh (x_t W_{xc} + h_{t-1} W_{hc} + b_c)$

Hidden State：$h_t = o_t * tanh (c_t)$

透過 Gate 來決定這個時間點的資料與前一刻 Hidden State 的輸出，要保留多少比例，然後由 Cell State 調整要記憶的特徵，接著由 Hidden State 輸出這個時間點上的資料特徵，透過兩個 State 的輸出串起不同時間上特徵的傳輸。

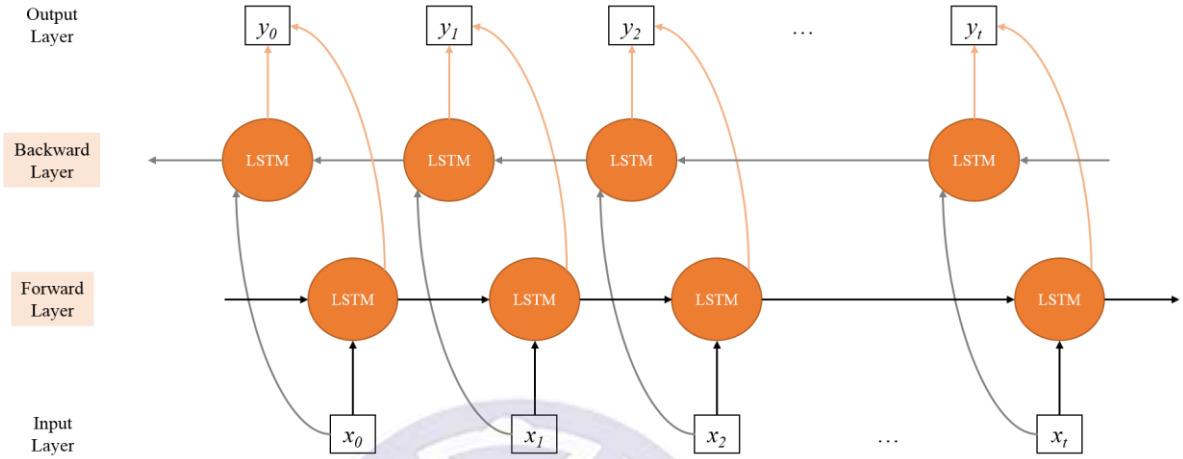

圖 8. Bidirectional Long-Short Term Memory 結構圖。

LSTM 會考量在某個時間點與之前資料的關係，然而時間序列資料不單只與前段時間有關，例如：文章的撰寫上需要承先啟後，除了前面的文字外，也要考慮到後面的句子想要表達的內容，對原本的遞迴神經網路進行修改，提出了雙向遞迴神經網路(Bidirectional Recurrent Neural Network) [31]，而將此結構套用到長短期記憶網路上，形成雙向長短期記憶網路(Bidirectional Long-Short Term Memory Network, Bi-LSTM)。

如圖 8.中的架構，Forward Layer 就是基本 LSTM 的運算流程，在此之上加入了兩個步驟－第一，透過 Backward Layer 獲得由後往前時間上的順序關係；第二，將往前和往後兩種順序得到的特徵結合，通常是直接串接在一起，藉由這樣的方式加強 Bi-LSTM 對時間上關係的敏感度。

2-3-3 典型卷積神經網路

卷積神經網路 (Convolution Neural Network, CNN) [32] 透過複數個過濾器 (Kernel)，在資料上擷取特徵，並且能夠根據輸入資料形態的不同，調整這些過濾器的維度數，例如：1 維卷積層處理時間序列資料、2 維卷積層處理平面圖形、3 維卷積層處理 3D 圖形或是影片，這樣的結構使 CNN 有很大的彈性。

從圖 9.能知道，二維 CNN 每層輸出都是二維度的特徵圖，專門學習圖形資料的特徵，除了圖片中做為目標的物件外，還能夠學習物件在圖片中的位置，甚至是與其他物件的關聯性。

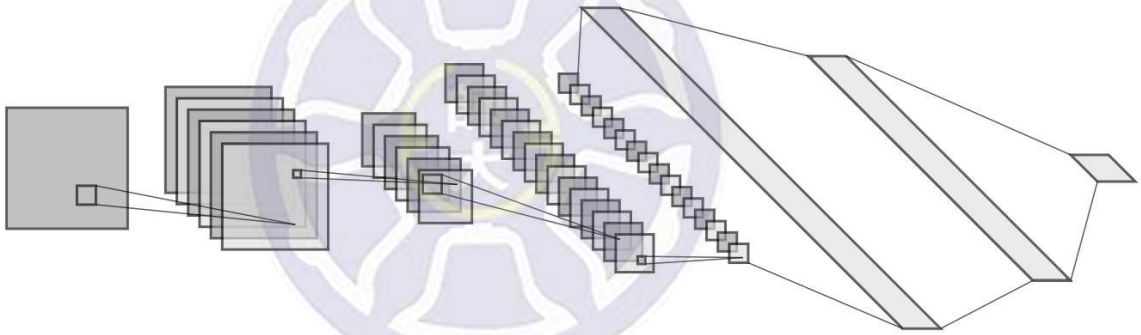

圖 9. 二維 CNN 模型圖。

圖 10.則是一維 CNN 結構的視覺化，與圖 9.比較，很明顯能看出在卷積核及輸入資料上的不同，由於無法像遞迴神經網路直接處理時間序列資料，在使用上需要先經過 Sliding Window 處理，將原本一段時間的資料，轉換成數個相同時間長度的資料，方能用 CNN 來處理，將對空間特徵擷取的能力，應用到連續手勢辨識上。

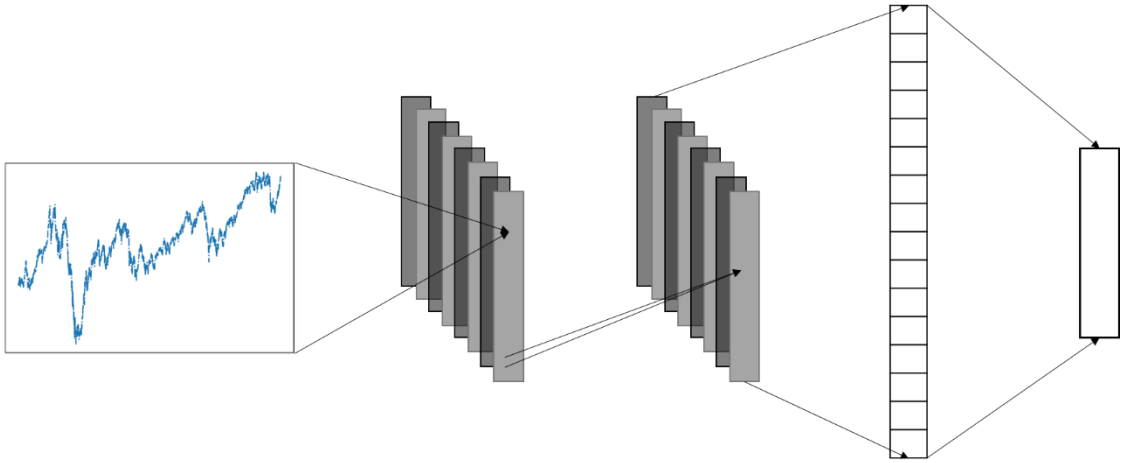

圖 10. 一維 CNN 模型圖。

從運算的角度上，時間序列資料的複雜度比圖形資料低，而一維 CNN 的參數量也比較少，所以在訓練時相對二維的模型更加容易。

2-3-4 PairNet

為了提升一維 CNN 的效能，在傳統的模型上進行修改並加入不同的結構，產生圖 11.中的 PairNet [33] 。

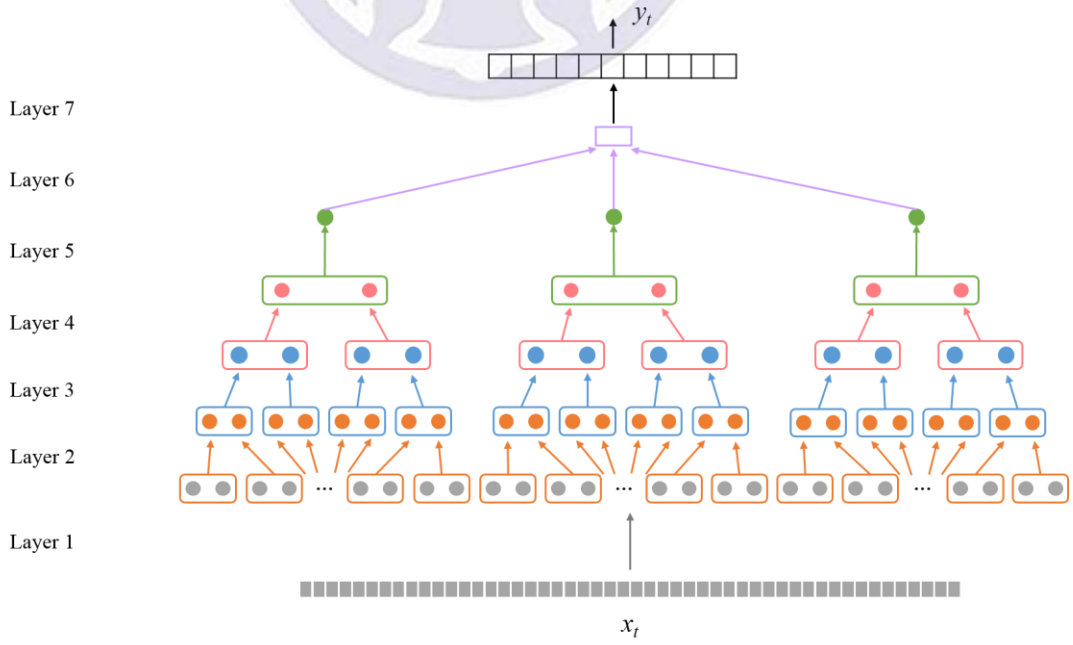

圖 11. PairNet 模型架構圖。

除了第一層使用 1x3 大小的過濾器，並且步伐設為 1 的交疊式(Overlapping)卷積層，第二到五層都使用 1x2 大小並且步伐為 2 的非交疊式(Non-overlapping)卷積運算，透過這樣的結構提高網路層的可見範圍(Receptive Field)，當可見範圍提高時便能使模型在萃取特徵時，能夠接觸到原始特徵圖的範圍越大，這意味著學習到的權重更加接近圖片特徵的分布。

從圖 11.中可以看到在進到第 7 層的全連接前，產生了 3 顆樹狀結構，為能夠有效統合不同顆樹狀結構之間的特徵，使用全域平均池化層(Global Average Pooling)取代對同一個特徵圖運算的平均池化層(Average Pooling)，在邏輯上形成類似聚合 3 個不同模型的效果。上述的設計使 PairNet 有著許多優點－第一，使用非交疊式卷積層降低的模型參數量，因此與交疊式卷積層相比較不會產生過度擬合(Overfitting)的現象；第二，在相同大小的輸入下，PairNet 能夠使用更少的模型層數與更低的參數量，減少運算所需的資源；第三，CNN 無法做到像 RNN 結構中，對時間概念上的記憶，透過 PairNet 的結構越深層能夠擁有更大的可見範圍，使卷積層能夠學習到不同尺度手勢的特徵。

2-3-5 ResNet (Deep Residual Network)

深度學習模型為了能夠解決更複雜的問題，會使用增加深度的方式來讓模型能夠看更多不同尺度的特徵，但是在加深網路上遇到了瓶頸，這個問題將會嚴重限制深度學習的發展，因此才會有 ResNet [34, 35] 的結構被提出來。

ResNet 論文中提出當他們想要透過提高神經網路深度，來增加模型辨識能力時，換來的卻是 56 層的模型辨識率比 22 層的模型低，而且是在訓練集與測試集上都發生這樣的現象，說明了問題不是因為參數提高導致的過度擬合(Overfitting)，他們稱此狀況為退化 (Degradation) 問題，也就是辨識率達到飽和，甚至變得比改變前還要更差。

為解決模型退化問題，Kaiming He 等人提出了圖 12.中的架構，將淺層的特徵透過 Shortcut 的形式直接傳遞到深層的地方，中間不經過任何運算，因此不會有任何參數最佳化的問題來阻擋特徵傳遞。而這樣的結構在訓練階段能讓特徵更有效地傳遞，結合不同深度的特徵來學習，另外更重要的是在調整參數階段的反向傳播運算 (Backpropagation) 上，這樣的設計解決梯度消失與梯度退化的問題，使得很深的模型能夠更容易地被訓練。

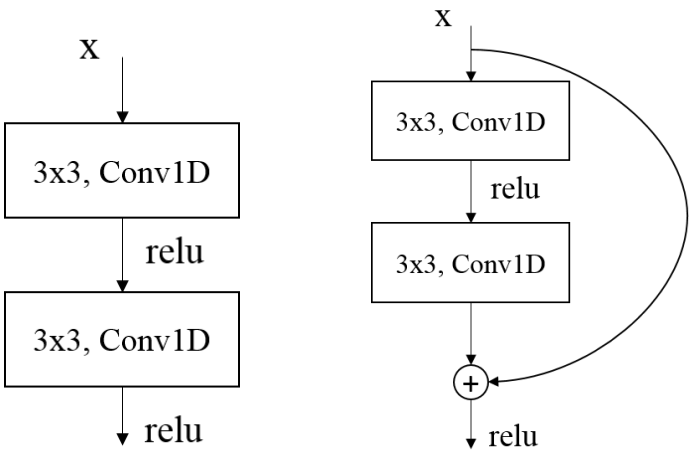

圖 12. ResNet 基本單元結構。左圖為典型卷積神經網路層結構，右圖為加入 Residual Learning 後的結構

ResNet 的論文中提出許多不同深度的模型，例如：32 層、56 層、110 層等，結果證明深度與模型辨識能力終成正比，越深層的模型能夠獲取更高的辨識率，甚至還提出超過 1000 層的模型，但當使用更深的模型 (例如：1202 層) 還是會遇到相同的問題，不過 ResNet 的殘差式學習 (Residual Learning) 結構確實解決了深度的限制，使科學家能夠透過調整深度有效解決問題。

2-3-6 GoogLeNet (Google Inception Network)

一般的 CNN 是透過縱向堆疊不同卷積核大小及輸出維度的卷積層，藉此讓模型學習不同尺度大小特徵，但要如何得知現在選擇的卷積核大小就是最佳的呢？為解決這個問題，Google 在 2014 提出在同一層中整合不同大小卷積核的 GoogLeNet [36]，其中核心結構 Inception 結構如圖 13.所示。

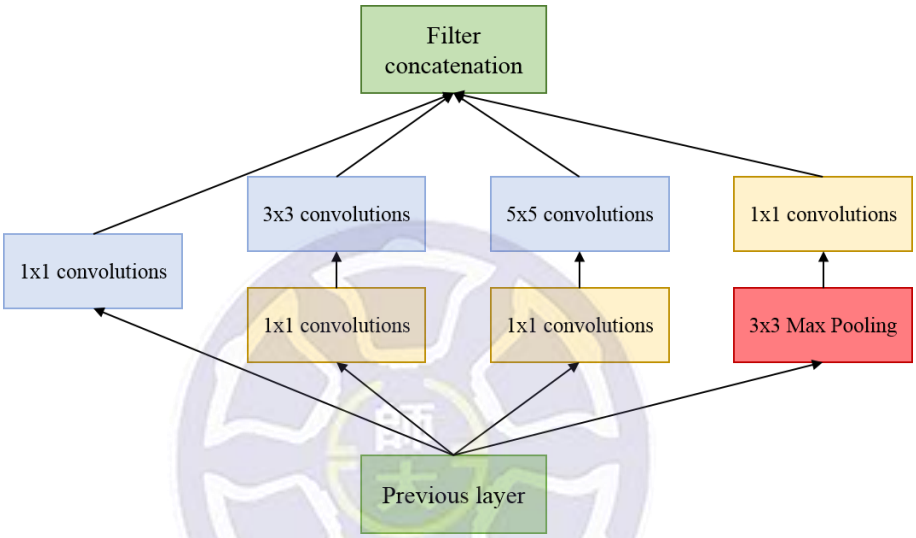

圖 13. Inception 結構圖。

Inception 採用 1x1、3x3 和 5x5 大小的卷積層，以及 3x3 大小的最大池化層(Max Pooling)，為了整合運算後的結果，加入了 1x1 的卷積層，對輸出的維度進行調整，使模型相對擁有更高的彈性，同時以較低的參數達到更高效率的運算。

總結 GoogLeNet 帶來以下貢獻－第一，使用 Inception 結構化深度學習模型，方便根據資料或目的調整；第二，以平均持化層(Average Pooling)取代傳統模型中最後使用的全連接層，大幅降低模型的參數量，並使特徵圖能包含不同類別的資訊；第三，堆疊許多小網路來行成大網路，使得參數的使用率提升。

2-3-7 Inception-ResNet (Inception-v4)

根據前面兩節提到的結構，ResNet 提高模型的深度，而 GoogLeNet 提高模型的寬度，那如果將這兩者的優點結合，形成一個又寬又深的深度學習模型，也就是由 Google 提出的 Inception-ResNet(又稱作 Inception-v4) [37]，並且在其中還包含了在 Inception-v2 與 Inception-v3 中提出的其他結構。

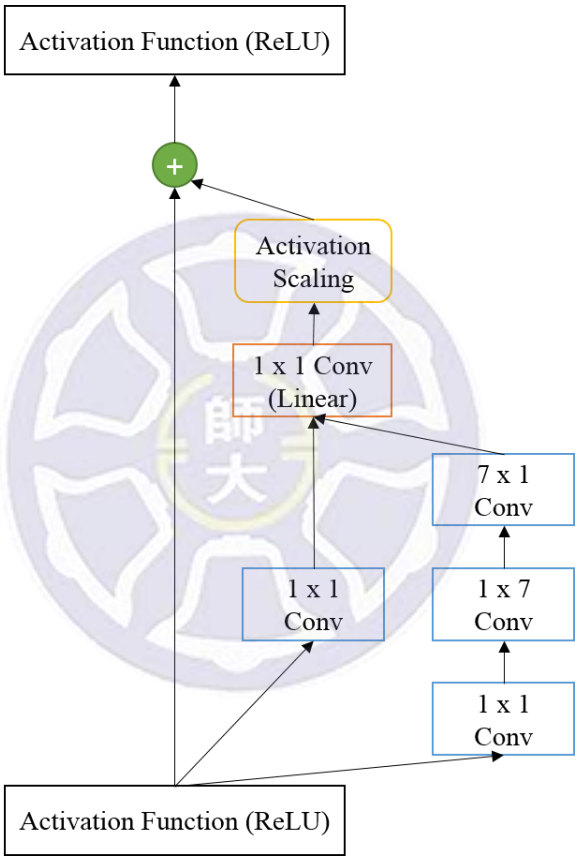

圖 14. Inception-ResNet 單元結構圖。

此模型為 Inception 系列之集大成者者，綜合了許多技術，例如：Inception-v2 中提出的 Batch Normalization，除了加速模型訓練時收斂的速度，某種程度上取代 Dropout 在解決過度擬合問題的地位；Inception-v3 中提出對卷積層的拆解(Factorization)，將一個二維卷積層拆解成兩個一維卷積層，減少模型參數的同時，

也透過增加一層卷積層使模型的非線性程度更高，進而提升模型辨識能力。

在圖 14.中可以看到將原本 7x7 的卷積層，拆解成 1x7 和 7x1 的卷積層，並透過一個線性的卷積層整合多個卷積層的輸出，在進到類似 ResNet 把深層與淺層特徵相加之前，Google 透過對實驗的觀察，選擇加入了 Activation Scaling 對 Inception 結構的輸出進行比例縮小，透過這樣的設計使模型在訓練時能夠更加穩定，收斂到更好的結果。

第三章　演算法則介紹

本章節將說明實驗的細節。3-1 描述整個實驗中是在怎樣的環境下進行；3-2 說明手勢收集的方法；3-3 到 3-5 說明如何把 ResNet、GoogLeNet 和 Inception-ResNet 整合進 PairNet 的架構中；3-6 說明後處理如何將模型的輸出，轉換成辨識結果。

3-1　實驗環境

實驗過程使用 NVIDIA GeForce GTX 1070 加速神經網路運算，演算法的部分皆基於 Python3 來撰寫；深度學習模型的部分，則是使用 Python 中的深度學習套件－Keras 實作。在面對不同階段的實驗中，使用的資料內容也不一樣，例如：訓練時採用單一手勢，而測試則是使用一段含有 1~4 個不等的手勢。

3-2　連續手勢資料收集與前處理

實驗中的手勢收集流程詳見圖 15.。透過智慧型手機 HTC ONE A9 與 Samsung Galaxy S8，由實驗室的同學協助搜集手勢，實驗過程中以 off-line 的方式將結果傳到電腦上，實際應用時接收手勢的平台除了電腦，也可以是嵌入式系統，例如：樹莓派 (Raspberry Pi)。

手勢由智慧手機上搭載的三軸陀螺儀(Three-axis gyroscope)與三軸加速度計(Three-axis accelerometer)產生的訊號做為輸入，採樣頻率為每秒採集 50 個點，而且每筆手勢的長度皆不固定。

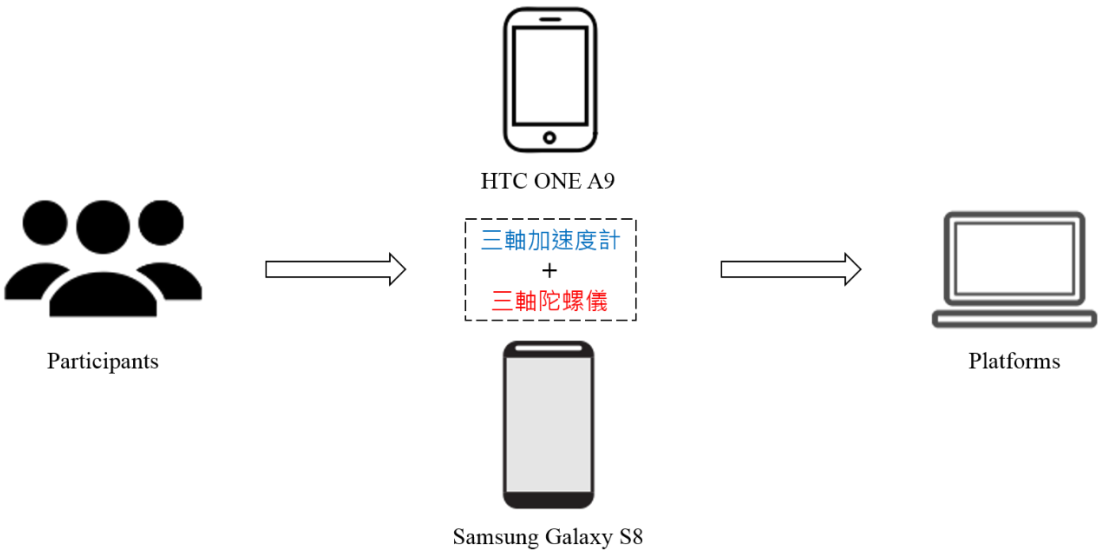

圖 15. 實驗中收集手勢的流程。

收集時由使用者以右手握住手機，手掌與螢幕朝同一個方向，如圖 16.中所示。按下應用程式中設計的開始鍵然後做動作，做完後維持停止動作一段時間，應用程式此時會去偵測有 1 秒以上的動作停滯，便認為使用者已經做完手勢，按時間先後堆疊三軸加速度計與三軸陀螺儀資料，將剛剛得到的資料寫到文件中。

圖 16. 手機產生手勢資料圖示。

然而判斷停止條件的 50 筆資料也會被一起寫入資料中，如圖 17.中左邊所示，而這些資料是無法表示特徵的無效資料，所以在實驗中會透過前處理的方式去掉這部分，切除後得到圖 17.中右圖的結果。

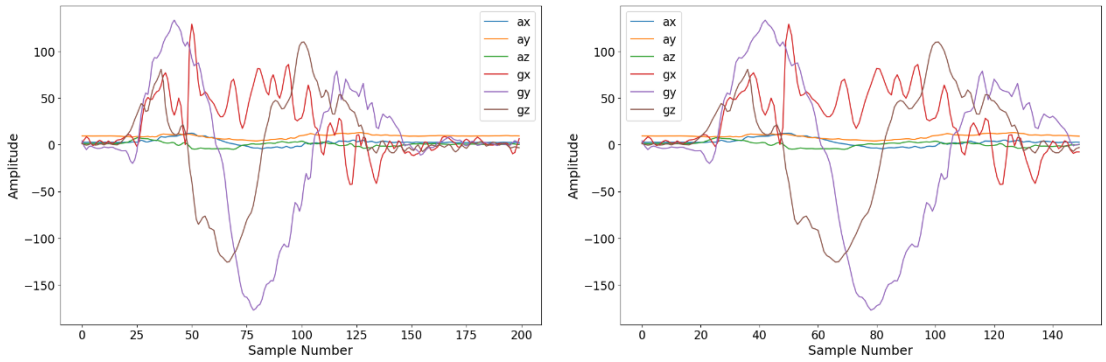

圖 17. 手勢波形前處理視覺化。

3-3 辨識模型－ResPairNet

為能使深度有效發揮對辨識能力的影響，將 ResNet 採取的殘差式學習(Residual Learning)結構整合到 PairNet 上，結構如圖 18.所示。

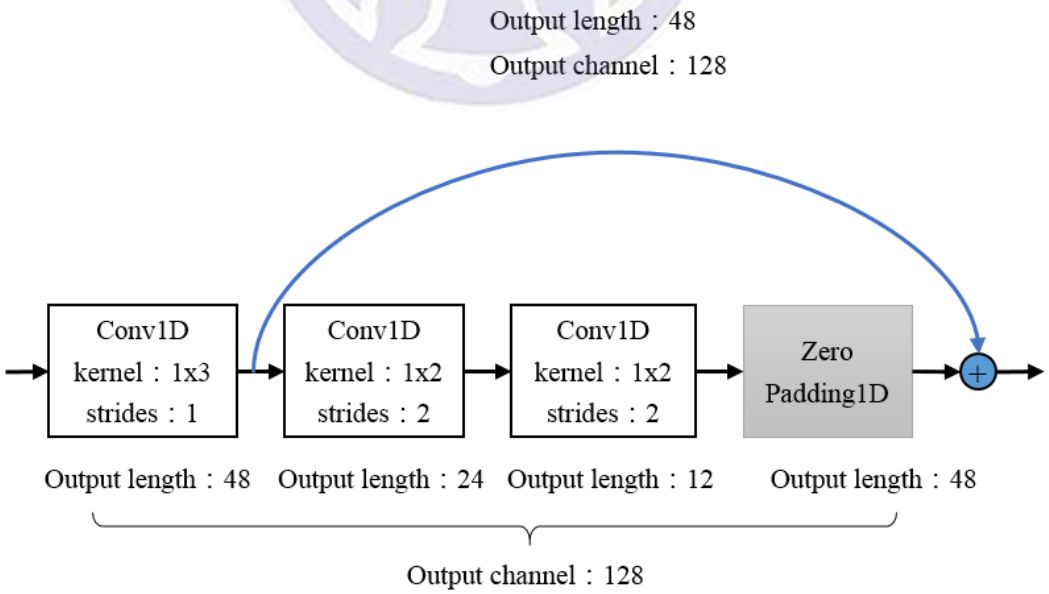

圖 18. 一般 ResPairNet 結構圖。

殘差式學習中最核心的技術，是透過 Shortcut 將淺層與深層的特徵結合，不過由於 PairNet 使用非交疊摺積運算的特性，會使特徵圖的大小在過程中迅速縮降，因此額外加入 Zero Padding 在前後補 0，來維持特徵圖的大小，使不同深度的特徵能夠結合，讓模型能夠找到更適合的權重。

PairNet 也與一般的 CNN 的設計相同，隨著模型的深度越深，卷積層輸出的通道數(Channels)也會提升，意味著使用更多數量的摺積核來擷取特徵，但 Zero Padding 無法解決通道數的差異，原本的 Shortcut 結構在此便會遇到問題。為處理這個狀況，便在圖 19.中上圖的藍線上加入了 1x1 的卷積層，提升淺層的特徵通道數，以圖 19.結構為例，將原本 48 x 128 的特徵圖，增加成大小為 48 x 256 的特徵圖，藉以維持 ResPairNet 中深、淺層特徵結合的結構。

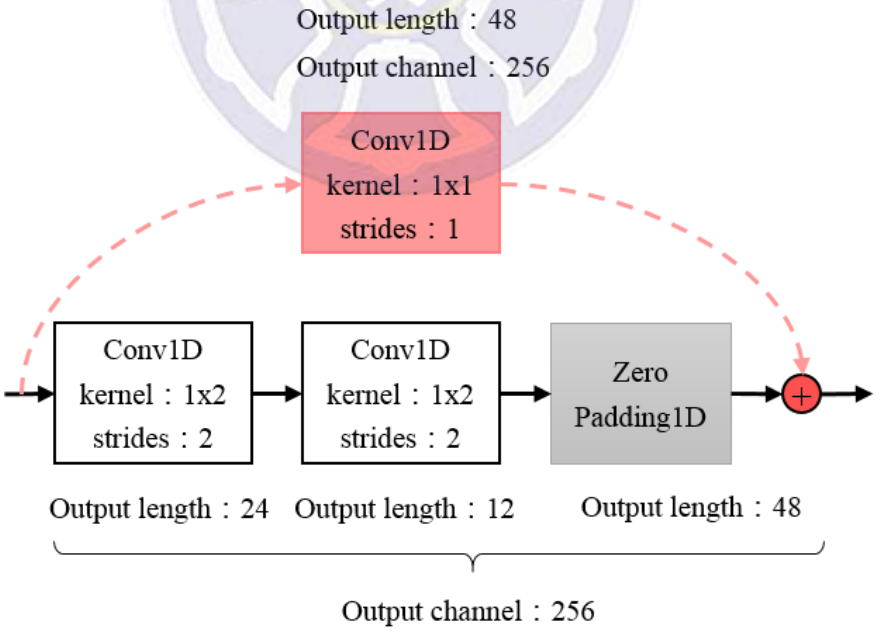

圖 19. 調整輸出通道數的 ResPaiNet 結構。

3-4 辨識模型－IncePairNet

在 PairNet 模型結構中除第一層使用 1x3 的卷積核，其餘皆使用 1x2 大小，這種同一深度中使用固定大小卷積核的設計，在典型的卷積神經網路中很常見，例如：LeNet、AlexNet、VGGNet 等，而將 Inception 中的結構整合進 PairNet，形成 IncePairNet 的結構，結構如圖 20.，是希望能夠對同一深度的特徵，使用不同大小卷積核作運算，將原本垂直堆疊不同大小卷積核，改成水平連接的結構。

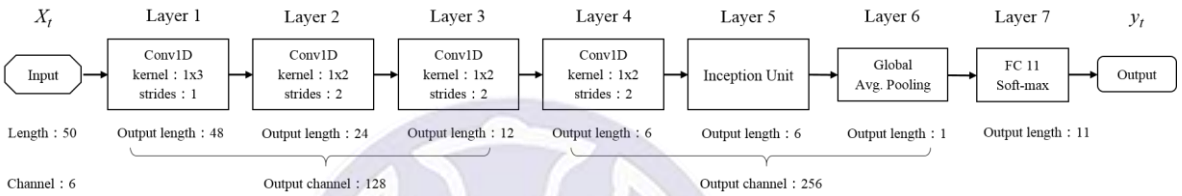

圖 20. IncePairNet 模型結構圖。

在 IncePairNet 中，使用三種不同大小運算－1x2 的卷積核、1x3 的卷基核與 1x3 的最大池化層(Max Pooling)，同樣使用 Zero Padding 來調整特徵圖大小，統一特徵圖的長度方便後續的計算，並且加入 1x1x 的卷積層降低特徵的通道數，這兩者能夠增加模型結構上的彈性，面對不同大小的輸入資料時只要微調就能使用，最後在透過基於通道串接 (Concatenate) 不同大小的特徵整合，藉此讓 IncePairNet 獲得不同視野下的特徵，如圖 21.中所示。

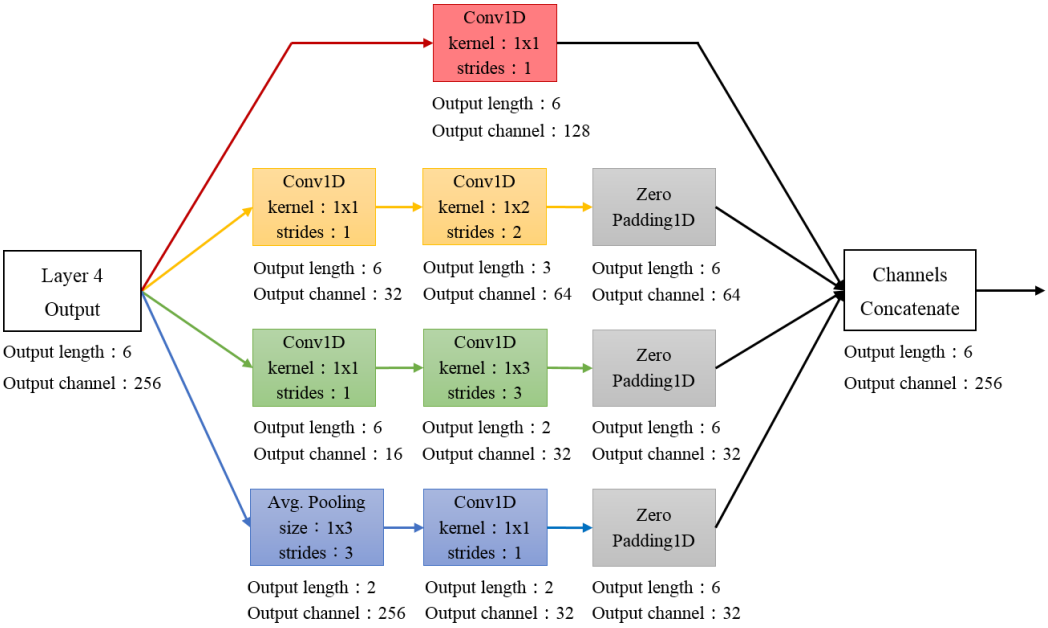

圖 21. IncePairNet 中 Inception Unit 結構圖。

在加入 Inception 的結構後，使得模型權重與整合前的 PairNet 相比降低不少，同時也提升了卷積層運算上的效率，另外也讓模型在學習中能夠看到更多樣性的特徵，從而增加辨識能力。

3-5 辨識模型－Ince-ResPairNet

在 3-3 與 3-4 中提到 ResNet、GoogLeNet 與 PairNet 的整合，將這些原本為處理影像問題而設計的模型結構，有效利用到時間序列資料處理上，讓前饋式神經網路能夠擁有更強的辨識能力，結合兩者優點的模型 Inception-ResNet 在深度與廣度上都有所提升，擁有低模型複雜度、高卷積層運算效率的優勢，並且在訓練階段梯度能夠更有效的傳播，而 Ince-ResPairNet 就是將 PairNet 與 Inception-ResNet 整合後的模型，模型圖見圖 22.。

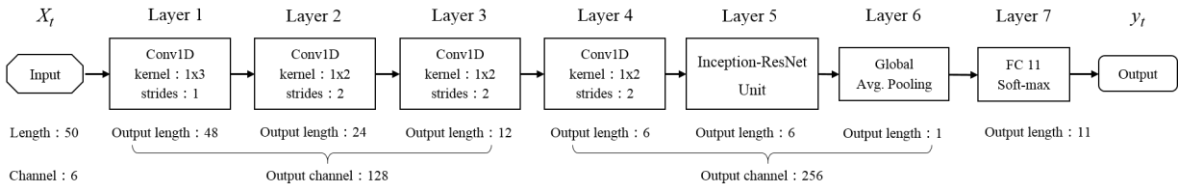

圖 22. Ince-ResPairNet 模型結構圖。

Ince-ResPairNet 中採用 1x2 與 1x3 兩種大小的卷積核，後面一樣有 Zero Padding 層來調整兩者以順利結合特徵，接著透過 1x1 的卷積層幫助使之能夠與淺層的特徵結合，這裡在設計上還有另外一個特別之處－透過加入一個縮小特徵值的 Scaling Down 層，使訓練過程更加穩定，原始論文中提到此值的範圍介於 0.1 到 0.3 之間，實驗後決定在此採用 0.2，上述的架構可在圖 23.中看到。

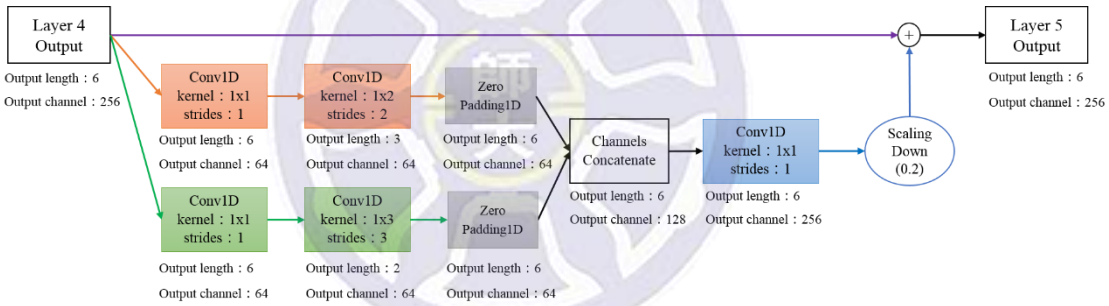

圖 23. Inception-ResNet Unit 結構圖。

3-6 模型辨識結果之後處理

本實驗中提出圖 24.中所示的演算法架構，來解決連續手勢的問題。令 X 為一測試集的資料，x_t 表示第 t 個時間點的資料，而 T 則表示此資料總長度。透過前處理將輸入資料切成 T 筆長度為 N (=50)的子資料，傳給事先訓練好的深度學習模型進行辨識，產生結果 $Y=\{y_1,\ldots,y_T\}$，其中 y_t 代表模型對第 t 個時間點資料的辨識結果。 y 是經過 Softmax 函數輸出的機率矩陣，分別對應到手勢 1~11，接著根據 Maximum A Posterior (MAP) estimation 取其中最大值，以該項所代表的手勢種類，作為此段輸入資料的辨識結果。

另外，若直接使用 Sliding Window 最後輸出的長度不會是 T，有鑒於陀螺儀與加速度計在停止狀態的值都是 0，於是在資料前後補上一定數量的 0，使輸入和輸出的長度能夠相同。這樣的設計雖使結果含有不必要的資訊，但從實驗結果發現影響不大，且能有效解決長度不一致的問題。

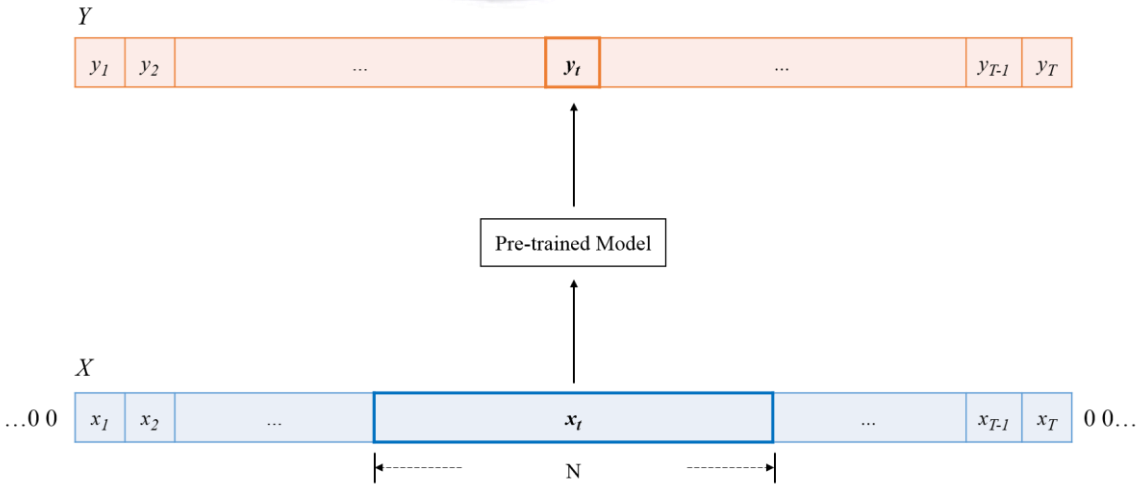

圖 24. 連續手勢辨識流程示意圖。

第四章　實驗結果與分析

本章節著重在深度學習模型於資料集上辨識結果的比較。4-1 說明本次實驗中採用的資料集，包含訓練集和測試集。4-2 說明實驗進行方法與如何評估實驗結果。4-3 說明本次實驗中採用的深度學習模型。4-4 將透過各種數據比較深度學習模型的表現。

4-1　手勢訓練集與測試集

本實驗中的資料集包含 11 個手勢，如圖 25.所示，訓練集與測試集同樣是以這 11 個手勢組成。

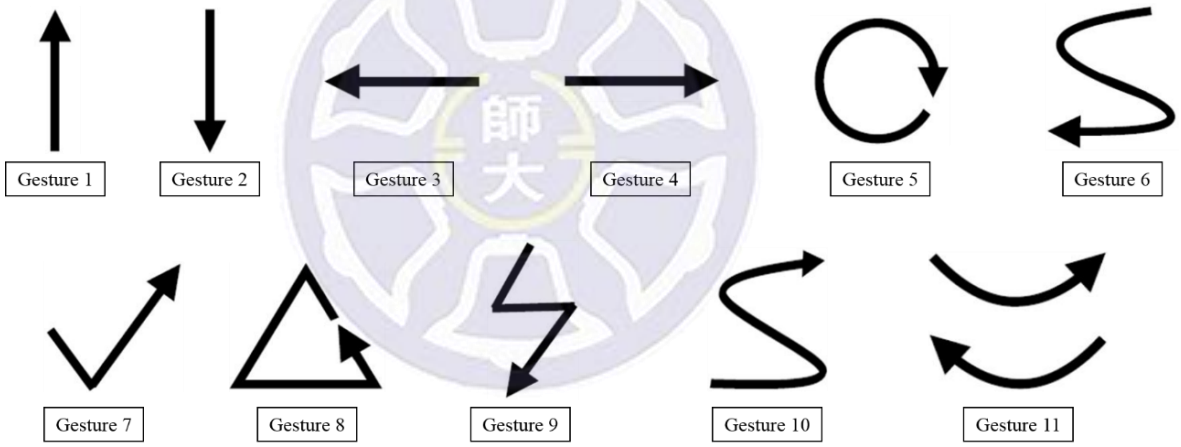

圖 25. 實驗採用的 11 種手勢。

訓練集由兩位同學參與收集，每一筆資料都只有 1 個手勢，總計有 1100 筆訓練資料；測試集由六位同學參與收集，每一筆資料含有 1~4 個手勢，手勢與手勢之間並沒有特別停頓，總計有 3404 個手勢。根據圖 26.中將訓練集每個種類選取單一筆資料視覺化，可以發現不同種類的手勢有屬於自己的波形特徵，而神經網路透過學習這些特徵，得以在測試時區分不同種類手勢之間的差異。

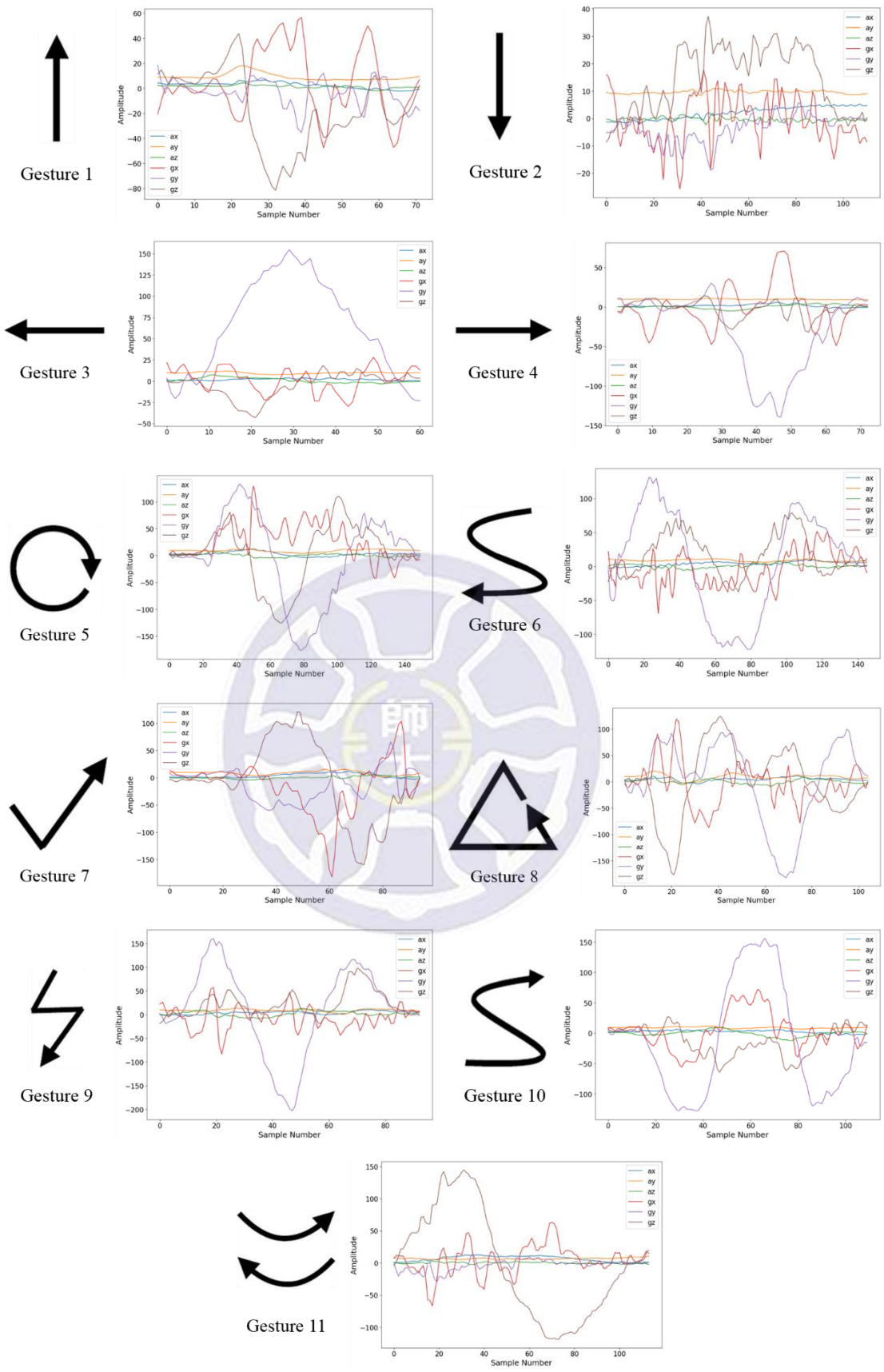

圖 26. 訓練集手勢 1~11 單筆資料中三軸陀螺儀與三軸加速度計之視覺化。

4-2 實驗流程說明

測試集中的手勢由於沒有特別標示，所以很難直接從波形上區分出手勢之間的切割點(Spotting)，如果今天是採用影像形式的資料，可以直接從影像中區分受測者現在是做什麼樣的手勢。

圖 27.取測試集中一筆檔案來做視覺化，已知此資料依序為手勢 2、10、5、9，然而即便知道這個測試資料中依序含有四個不同的手勢，也沒有辦法輕易將波形分割成對應到各自手勢的區段，因為我們無法單純從波形的變化，就能推得實際上它是某類手勢的一部分。

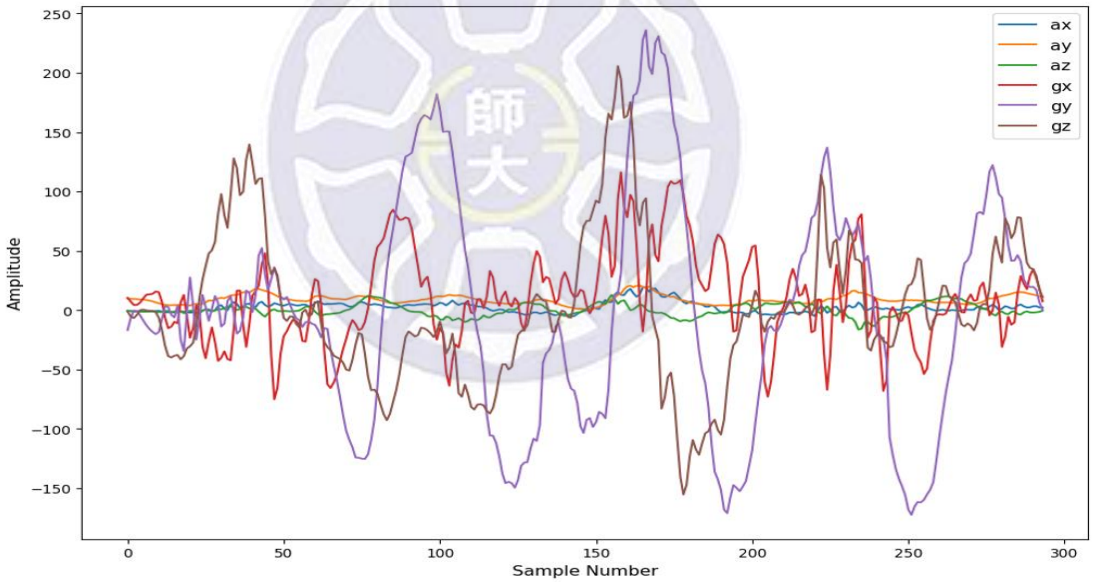

圖 27. 單筆測試集資料視覺化。

經過 3-6 中的演算法，將手勢資料傳進深度學習模型進行辨識，便可將圖 27. 中的測試資料切割成對應的手勢。如圖 28.中所示，只要中間有出現任何其他手勢就中斷然後計算該手勢長度，例如：手勢 5 與手勢 9 之間就有一小段被辨識成手勢 10，這時便會計算綠色框框部分當作手勢 5 的長度。後處理的目標就是找出連續長度最長的 K 個手勢，K 則是由該筆測試資料的 Label 包含多少個手勢決定，也就是實驗中 K 的大小是已知的。

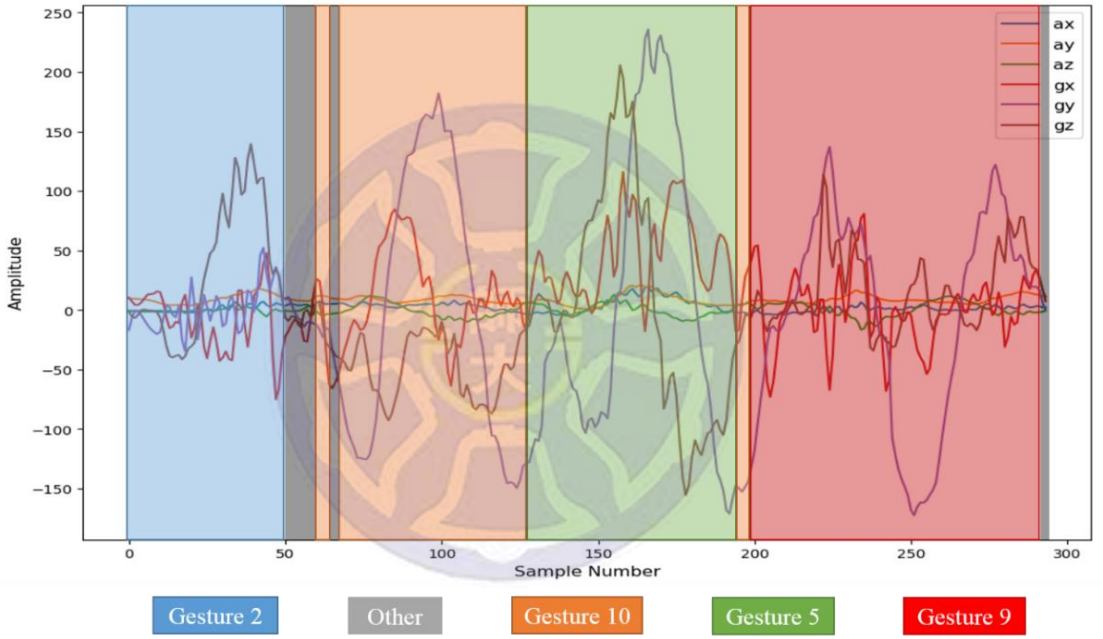

圖 28. 手勢辨識結果之視覺化。

在得到辨識結果後透過下列式 1.來計算，比較辨識與實際上的差距，用以評估模型對這筆資料的準確率 (Hit Rate)。

$$\text{Hit Rate} = \frac{\text{正確辨識出的手勢個數}}{\text{資料中含有的手勢個數}} \quad (1)$$

4-3 實驗採用的模型

本實驗使用第二、三章中提出的深度學習模型－PairNet、CNN、LSTM、Bi-LSTM、ResPairNet、IncePairNet 和 Ince-ResPairNet，總計七個模型，過程中使用 4-1 提到的資料集，來對模型做訓練及測試，所有的實驗都是在相同的環境下進行。

4-4 實驗結果分析

將測試集中的資料送進已經訓練好的 7 個模型中，透過 4-2 中提出的後處理，並經由式 1.計算辨識準確率，便可得到表 1.中的統計結果。

表 1. 實驗模型對測試集之辨識率

	PairNet	CNN	LSTM	Bi-LSTM	ResPairNet	IncePairNet	Ince-ResPairNet
辨識率	92.36%	90.42%	88.60%	90.19%	95.30%	93.63%	94.10%

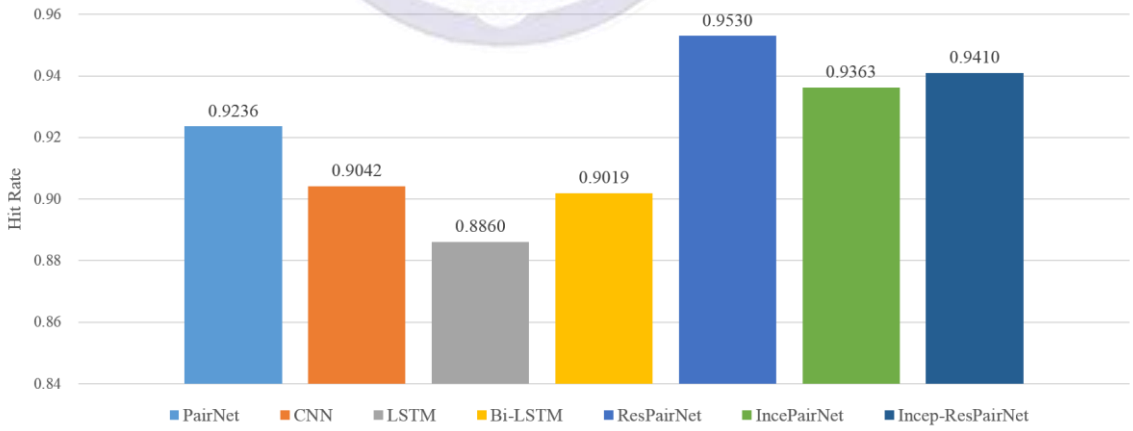

圖 29. 實驗模型對測試集之辨識率。

根據圖 29.可以觀察到，CNN 相較於 LSTM 的辨識率更高，即便是以 Bi-LSTM 做為基準，兩者的辨識率都是 90%左右，而 PairNet 在實驗結果上取得高於他們的 92.36%，從這結果可以推得—FNN 能夠處理時間序列資料，甚至比 RNN 有更好的辨識能力。

接著比較基於 PairNet 結構中「卷積核大小與步伐大小一致的非交疊摺積運算」與「使用 Global Average Pooling 整合輸出」的設計，加入 ResNet、GoogLeNet 與 Inception-ResNet 設計出的模型。

在圖 29.中可以觀察到 ResPairNet 的辨識率 95.30%位居所有模型之冠，而 IncePairNet 與 Ince-ResPairNet 也取得比 PairNet 高出 1%以上的改善，不過由於層數少所以還無法完整發揮這兩者的能力。不過藉由這樣結果可以推得原本為了影像問題中物件辨識設計的結構，在時間序列資料上也能夠提高卷積層學習能力，即便實驗使用的模型深度只有 7-8 層，相較於 ResNet 最簡單的 18 層或 GoogLeNet 的 22 層少非常多，但在處理連續手勢這種時間序列資料上，不僅穩定處理傳統上切割點(Spotting)的問題，辨識率也有顯著的提升。

透過 Keras 中 Model.summary()得到的參數總量，可做為評估模型複雜度的標準。雖然 RNN 在辨識能力上遜於 FNN，但由於其「遞迴」的設計－共用權重，從表 2.中可以看到 LSTM 與 Bi-LSTM 的參數量遠比其模型低數十倍。

表 2. 實驗模型之參數總量

	PairNet	CNN	LSTM	Bi-LSTM	ResPairNet	IncePairNet	Ince-ResPairNet
參數量	271,116	426,764	5,388	10,764	304,396	200,844	244,876

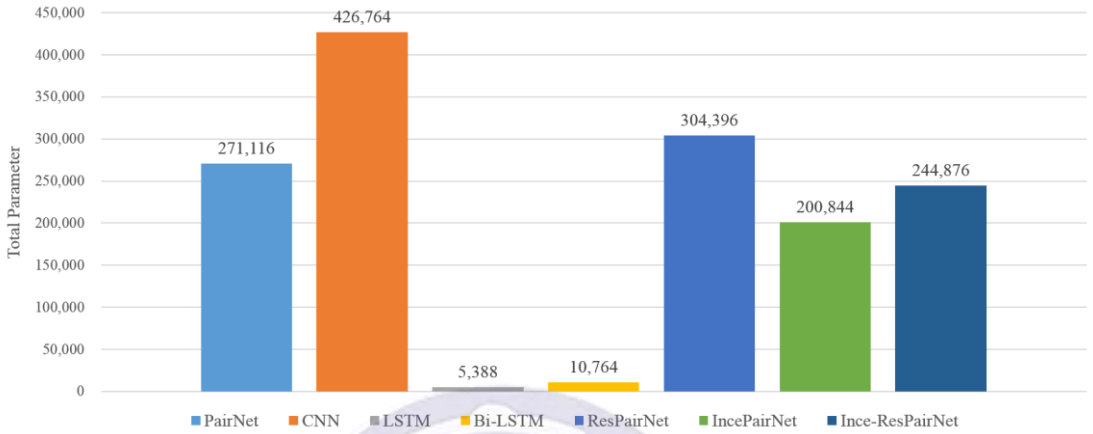

圖 30. 實驗模型之參數總量。

另外，基於 PairNet 改進的模型在參數量上即便比 PairNet 高，但與 CNN 比起來仍舊有至少 30%的降低，尤其結合 Inception 設計中多尺度卷積核與 1x1 卷積層的 IncePairNet，參數量比 PairNet 降低了 25%。比較少量的參數量，代表模型需要的運算量比較低，當我們想要複製這樣成功的結果到其他平台，例如：嵌入式系統上時就有著莫大的優勢。

從整體辨識率上能夠觀察到模型之間的優劣，但為什麼能夠得到比較好的辨識結果，這部分可以透過比較混淆矩陣 (Confusion Matrix) 來理解。

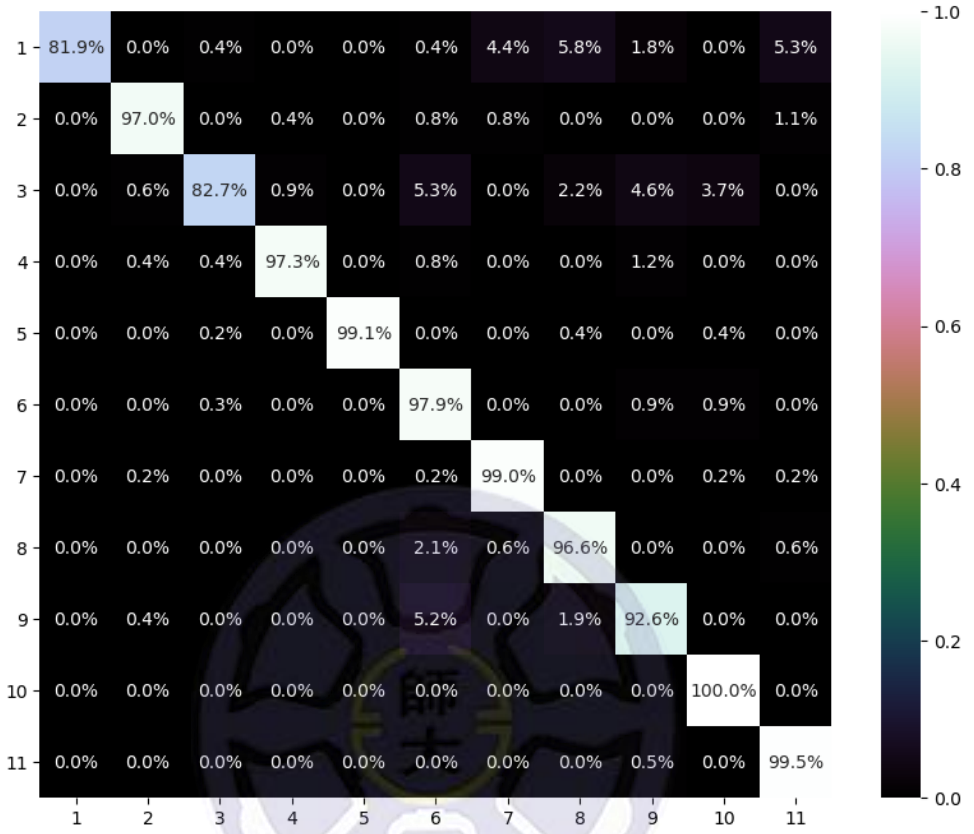

圖 31. 辨識結果之混淆矩陣視覺化。

這裡以總體辨識率最高的 ResPairNet 來做舉例，圖 31.中的混淆矩陣直向的 1~11 表示實際的種類，橫向的 1~11 表示辨識出的種類，Confusion Matrix 中[1, 1] 表示「手勢 1 被辨識成 手勢 1」的機率有 81.90%。

從圖 31.中 Confusion Matrix 的斜對角線上，可以觀察到實際上辨識時各手勢的辨識率，除此之外還能了解更詳盡的訊息，例如：手勢 1 之所以辨識率只有 81.9%，是因為有一部分的資料被模型當做手勢 7、8 或 11，透過 Confusion Matrix 能更深入研究各模型的問題所在。

根據圖 29.的圖表了解各模型對測試集的辨識率，但也透過圖 31.能觀察到因為某部分手勢，影響模型整體的辨識率，究竟這樣因素有多大的影響，能從表 3.中模型對單一種類手勢的辨識率來觀察。

表 3. 各類手勢辨識率統計表

	PairNet	CNN	LSTM	Bi-LSTM	ResPairNet	IncePairNet	Ince-ResPairNet
Gesture 1	70.35%	69.03%	66.81%	76.11%	81.86%	75.66%	77.88%
Gesture 2	90.19%	74.72%	76.23%	81.51%	96.98%	91.70%	95.47%
Gesture 3	72.14%	70.90%	68.11%	74.61%	82.66%	75.85%	75.23%
Gesture 4	86.43%	85.66%	79.07%	84.11%	97.29%	91.09%	92.25%
Gesture 5	99.12%	99.12%	98.07%	98.77%	99.12%	99.12%	98.95%
Gesture 6	97.93%	97.34%	96.75%	99.70%	97.93%	99.11%	98.82%
Gesture 7	98.23%	96.97%	95.96%	92.17%	98.99%	97.73%	98.99%
Gesture 8	98.16%	96.93%	93.56%	93.25%	96.63%	99.69%	97.55%
Gesture 9	94.05%	92.94%	89.96%	85.50%	92.57%	92.57%	94.05%
Gesture 10	100.00%	99.07%	98.61%	96.76%	100.00%	100.00%	100.00%
Gesture 11	99.54%	99.54%	98.15%	99.54%	99.54%	99.54%	99.54%

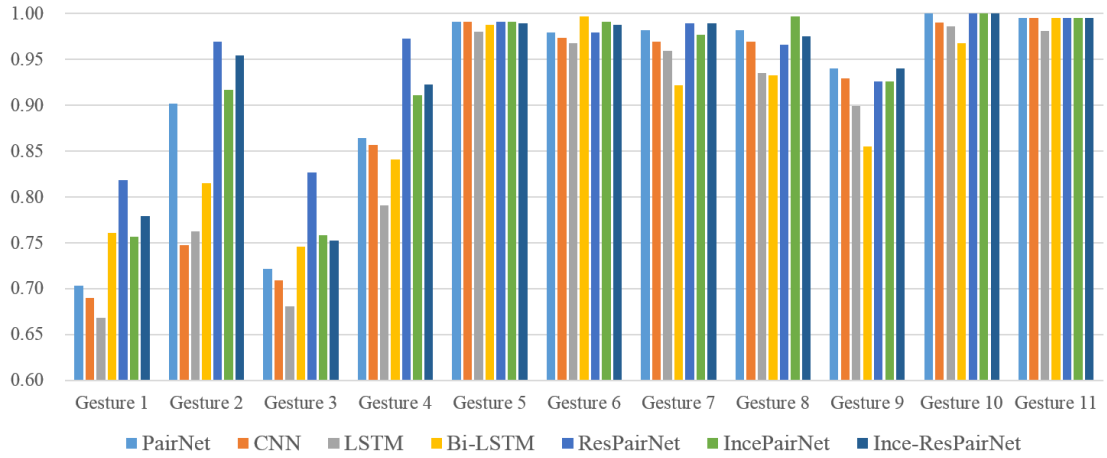

圖 32. 各類手勢辨識率統計圖。

圖 32.中能看到與圖 31.類似的狀況，同樣在前四個手勢有比較大的差距，經推測應該是由於 1~4 號手勢是比較簡單的動作(上下左右)，在辨識時很容易被當作比較複雜手勢的一部分，進而使前四類手勢的辨識結果比較不理想，其他種類的手勢相較之下辨識率差異不大，而前饋式神經網路正因能夠有效將前四類手勢區分出來，才使得辨識率能比遞迴神經網路更高，這些數值上的分布能夠從表 3.中觀察到。

另外，雖然實驗是採用三軸陀螺儀與三軸加速度計組成的六維資料，在做手勢的過程中陀螺儀的變化幅度會比較大，那加速度計是否有需要？表 4.中根據輸入資料維度不同，比較辨識率的變化。

表 4. 輸入資料維度比較統計表。

	Both	**3-Axis Gyroscope**	**3-Axis Accelerometer**
PairNet	92.36%	90.19%	80.43%
CNN	90.42%	86.84%	77.41%
LSTM	88.60%	84.67%	72.39%
Bi-LSTM	90.19%	87.57%	72.56%
ResPairNet	95.30%	90.28%	79.05%
IncePairNet	93.63%	90.36%	81.96%
Ince-ResPairNet	94.10%	90.54%	82.20%

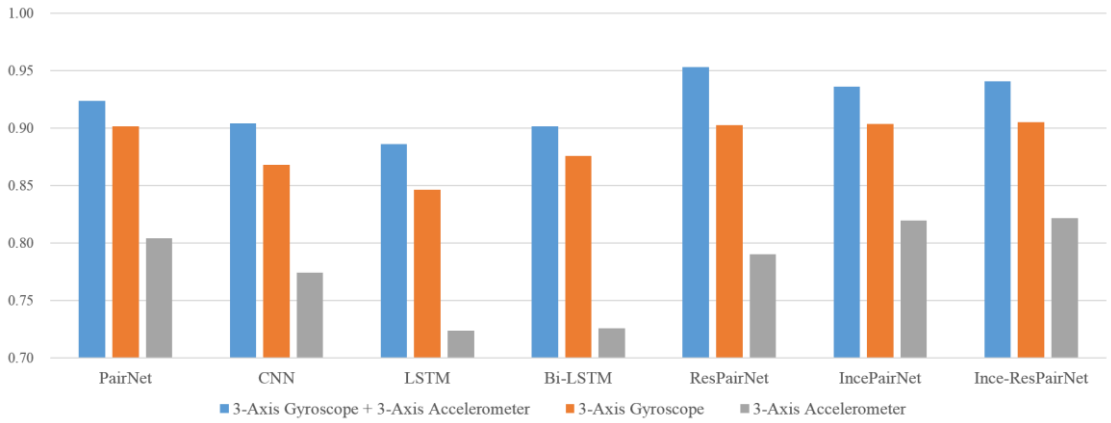

圖 33. 輸入資料維度比較統計圖。

從圖 33.中可觀察到不論哪一個類神經網路模型，只使用陀螺儀的資料比只使用加速度計的資料能夠得到更高的辨識率，代表著陀螺儀比較能表達手勢特徵。不過整合兩者的六軸資料還是能得到比較好的結果，證明實驗中採用三軸陀螺儀和三軸加速度計做為輸入資料，比只使用單一感測器組成手勢資料時，對連續手勢辨識更有效。

第五章　結論與未來方向

本論文提出基於感測器的連續手勢辨識系統，透過智慧型手機上配備的三軸陀螺儀與三軸加速度計產生的訊號組成手勢。透過深度學習我們不再需要先找出連續手勢之間的 Spotting，就可以直接去作辨識，並且透過實驗證明 CNN 這個主要使用於影像處理及辨識上的架構，也能夠在時間序列資料上有很強的學習能力，例如：PairNet 比 LSTM 辨識率高出將近 4%，而透過整合 ResNet、GoogLeNet 結構，也幫助前饋式神經網路更準確找到連續手勢之間的切割點(Spotting)，區分不同手勢種類之間資料上的差異，例如：ResPairNet 在辨識率上比起原型的 PairNet 有 3%左右的增幅，再次驗證類神經網路模型於實際應用上可行性。

在辨識能力上當然還有可以改善的空間，但是若要能更讓連續手勢辨識系統實作在現實生活中，還有許多必須解決的問題。目前收集手勢的系統中，一開始的地方還是需要按下啟始按鈕，相較於控制家電是由使用者自主決定，為保障安全設計的 AA 則會面臨到問題，例如：無法每隔一段時間就彈出視窗要使用者按下開始扭，輸入驗證用的連續手勢，而手機這端要如何辨識做出手勢的人是合法的使用者，這是一個需要考量的問題；另外，若使用者做了非事先決定好的動作，也會被辨識成已知手勢，如何區隔「背景手勢」與已知手勢，以及是否能辨識未知數量的連續手勢，通過解決這些問題才能夠奠定這套系統於日常生活中應用的可能性，這也是連續手勢辨識上未來需要面對的挑戰。

參考文獻

[1] S. Mitra and T. Acharya, "Gesture Recognition: A Survey" in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311-324, May 2007.

[2] Rung-Huei Liang and Ming Ouhyoung, "A real-time continuous gesture recognition system for sign language," *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, 1998, pp. 558-567.

[3] T. Starner, J. Weaver and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371-1375, Dec. 1998

[4] A. Agarwal and M. K. Thakur, "Sign language recognition using Microsoft Kinect," *2013 Sixth International Conference on Contemporary Computing (IC3)*, Noida, 2013, pp. 181-185.

[5] E. Ohn-Bar and M. M. Trivedi, "Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2368-2377, Dec. 2014.

[6] Hyun Kang, Chang Woo Lee, and Keechul Jung. 2004. Recognition-based gesture spotting in video games. *Pattern Recognition Letters*. 25, 15 (November 2004), 1701-1714

[7] G. D. Clark and J. Lindqvist, "Engineering Gesture-Based Authentication Systems," in *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 18-25, Jan.-Mar. 2015.

[8] Oza, P., & Patel, V.M. (2019). Active Authentication using an Autoencoder regularized CNN-based One-Class Classifier. *CoRR, abs/1903.01031*.

[9] A. Pozo, J. Fierrez, M. Martinez-Diaz, J. Galbally, and A. Morales, “Exploring a statistical method for touchscreen swipe biometrics,” in *Security Technology (ICCST), 2017 International Carnahan Conference on*. IEEE, 2017, pp. 1–4.

[10] P. Perera and V. M. Patel, “Extreme value analysis for mobile active user authentication,” in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 346–353.

[11] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, “Touch gesture based active user authentication using dictionaries,” in *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2015, pp. 207–214.

[12] P. Perera and V. M. Patel, “Towards multiple user active authentication in mobile devices,” in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*. IEEE, 2017, pp. 354–361.

[13] Rautaray, Siddharth S., and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: a survey." *Artificial Intelligence Review* 43.1 (2015): 1-54.

[14] Murthy, G. R. S., and R. S. Jadon. "A review of vision based hand gestures recognition." *International Journal of Information Technology and Knowledge Management* 2.2 (2009): 405-410.

[15] M. Elmezain, A. Al-Hamadi, J. Appenrodt and B. Michaelis, "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," *2008 19th International Conference on Pattern Recognition*, Tampa, FL, 2008, pp. 1-4.

[16] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," *Proceedings of International Symposium on Computer Vision - ISCV*, Coral Gables, FL, USA, 1995, pp. 265-270.

[17] Malima, Ozgur and Cetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control," *2006 IEEE 14th Signal Processing and Communications Applications*, Antalya, 2006, pp. 1-4.

[18] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta and K. Sharma, "A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors," in *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6425-6432, Aug.15, 2016.

[19] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang and J. Yang, "A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors," in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064-1076, Nov. 2011.

[20] T. Tai, Y. Jhang, Z. Liao, K. Teng and W. Hwang, "Sensor-Based Continuous Hand Gesture Recognition by Long Short-Term Memory," in *IEEE Sensors Letters*, vol. 2, no. 3, pp. 1-4, Sept. 2018, Art no. 6000704.

[21] L. Yun and Z. Peng, "An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs," *2009 Second International Workshop on Computer Science and Engineering*, Qingdao, 2009, pp. 72-76.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 248-255.

[24] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[25] Svozil, Daniel & Kvasnicka, Vladimir & Pospíchal, Jiří. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*. 39. 43-62.

[26] J. Ilonen, J.K. Kamarainen, J. Lampinen, Differential evolution training algorithm for feed-forward neural networks, *Neural Processing Letters* 17 (2003) 93–105.

[27] Hochreiter, Sepp. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998): 107-116.

[28] Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." *In Proc. 30th International Conference on Machine Learning* 1310–1318 (2013).

[29] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

[30] Gers, F. A., Schmidhuber, J., & Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

[31] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[32] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, ECCV, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014.

[33] 張筠婕，《基於 PairNet 的連續手勢辨識》，國立臺灣師範大學資訊工程研究所

[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778.

[35] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016

[36] C. Szegedy et al., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9.

[37] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR Workshop*, 2016.