2021 年臺灣國際科學展覽會
優勝作品專輯

作品編號　**190018**

參展科別　電腦科學與資訊工程

作品名稱　語音情緒辨識之研究

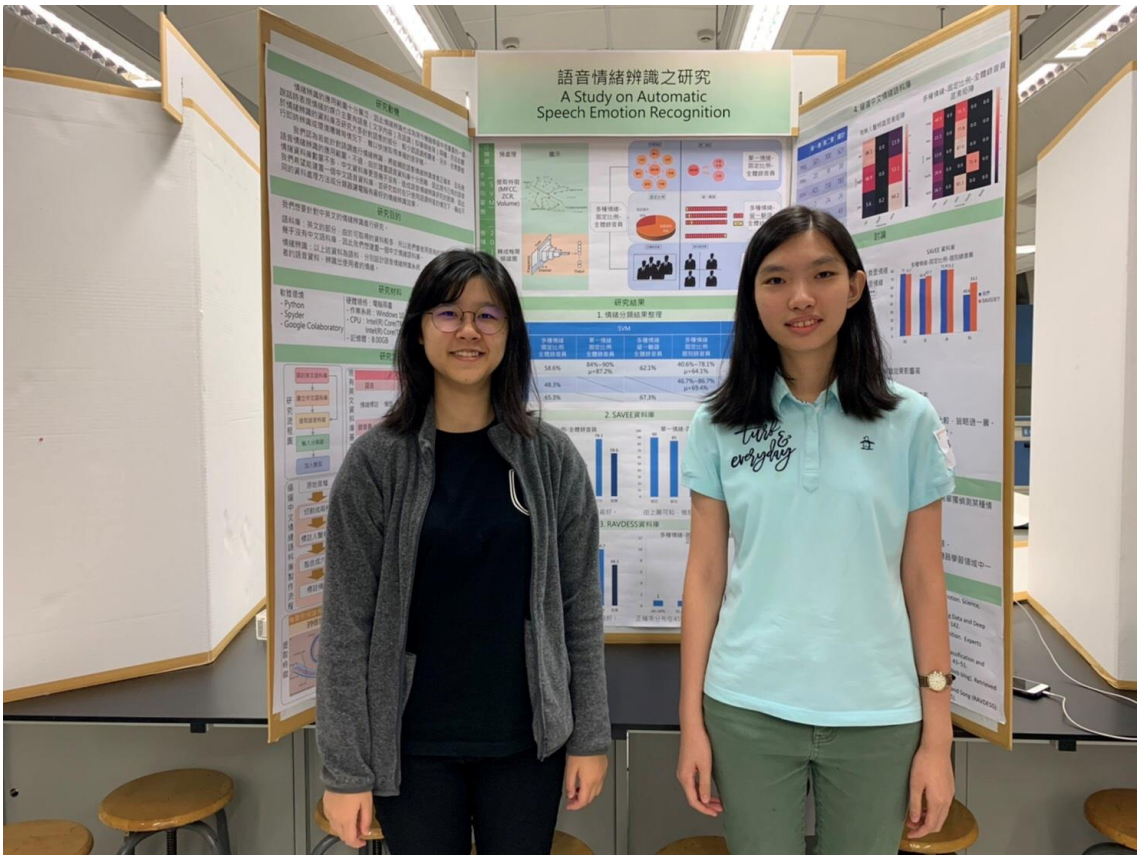
就讀學校　臺北市立第一女子高級中學

指導教師　許永真、黃芳蘭

作者姓名　劉又慈、張齊恩

關鍵詞　情緒辨識、監督式機器學習、卷積神經網路

作者簡介



我是劉又慈，今年就讀北一女三年級。很幸運能找到自己想研究的專題，並獲得了教授、老師、學姊、父母等人的幫助和支持，讓我和齊恩能盡情的投入到專題研究及國際科展之中。

我是張齊恩，今年就讀北一女三年級。很高興有機會跟又慈研究一個有趣的題目，也感謝過程中有許多人的幫助，讓我們能順利完成報告，並且從中學到許多。

摘要

情緒辨識是增進人際溝通的重要能力。如生命線、電話客服等應用情境缺乏表情、肢體語言等輔助時，單以語音進行情緒辨識有極高的實用價值。

本研究探討比較支持向量機（SVM）及卷積神經網路（CNN）兩種機器學習方法於訓練「AI 語音情緒辨識」分類器模型的表現。我們採用 SAVEE 和 RAVDESS 兩個英文語音資料庫，並自行製作與標註「逼逼中文情緒語料庫」。研究結果顯示 SVM 對 SAVEE 資料庫單一情緒的辨識正確率達 84~94%，個別錄音員正確率達 75%，超越官網紀錄的 73.7%。同時，實驗顯示深度學習的模型在訓練資料不足的狀況下，反而相對遜色。

Abstract

Emotion recognition is an essential skill to interpersonal communication. Emotion recognition by speech alone has great practical value in scenarios without the support of facial expressions, body language, and so on. Some examples of these situations include the Life Line and call centers.

Our study compares the performance of two machine learning methods, Support Vector Machine (SVM) and Convolutional Neural Network (CNN), on training 'AI speech emotion recognition' classifying models. We used two English databases, SAVEE and RAVDESS, and we also created and labelled the 'BB Chinese Emotional Speech Database'. Results show that the accuracy of SVM on SAVEE is 84~94% in the single emotion category, and can get to 75% in the speaker-dependent category, which is higher than the result on the official website, which is 73.7%. Also, the results from the experiments demonstrated that deep learning models are less effective in conditions with insufficient training data.

壹、前言

一、 研究動機：

每當生命線的輔導員接起電話，便要根據諮詢者的狀況判斷要花多少時間輔導該諮詢者，才能將有限的資源提供給最需要的人。如果能以電腦輔助判斷，預警哪些諮詢者情緒特別不穩定，將能減輕輔導員的負擔並讓諮詢者得到最大的幫助。除此之外，進行人機互動時，電腦若能辨視用戶當下的情緒，將可依此改變與人互動的語氣或方式，提供更人性化及差異化的服務，進而提升用戶的使用經驗。由上述例子可見，情緒辨識的應用範圍十分廣泛，因此情緒辨識也成為現今機器學習中很重要的一環。

說話時表現情緒的媒介主要有語意（文字內容）及語調（抑揚頓挫等）兩種，而目前關於情緒辨識的資料庫及研究大多針對語意的部分，較少跟語調相關者。另外，在環境嘈雜或需即時辨識等情況下，難以快速取得精準的逐字稿。我們認為若能針對語調進行情緒辨識，將能輔助語意情緒辨識增進正確率，並拓展語音情緒辨識的應用範圍。不過，由於建置語音資料庫十分困難，因此現有的語音情緒資料庫不多，而中文資料庫更是幾乎沒有，造成語音情緒辨識研究的困境。因此我們希望能建置一個中文語音資料庫，並研究如何在只使用語調特徵的情況下，藉由不同的資料處理方法或分類器讓電腦有最好的情緒辨識效果。

二、 研究目的

我們針對中英文的情緒辨識進行研究，研究主軸為建立中文語料庫及情緒辨識系統。

(一) 比較以支持向量機（Support Vector Machine, SVM）和卷積神經網路（Convolutional Neural Network, CNN）作為分類器的情緒分類正確率。

(二) 尋找特定條件下可增進正確率的方法：

1. 單一情緒實驗：某些情境下可單獨判斷特定情緒，以提升該情緒的正確率。
2. 留一驗證實驗：增加訓練資料的資料數量。
3. 個別錄音員實驗：探討錄音員間的差異對情緒辨識的影響。

(三) 研究如何降低建置語料庫的成本，並製作「逼逼中文情緒語料庫」，再以之實作中文語音的情緒辨識。

貳、研究方法或過程

一、 研究設備及器材

軟體環境：

(一) Python：Python 是物件導向、直譯式的高階程式語言，強調程式語言的簡潔、清晰，並以減少開發及維護成本的觀念發展。方便使用，可以完成各種難度的應用，並可在多數的系統中運行。我們選擇使用 Python 是因為它在機器學習方面有較多函式庫可使用。

(二) Spyder：Spyder 是一個使用 Python 語言的開放原始碼跨平台科學運算整合開發環境（Integrated Development Environment, IDE）。

(三) Google Colaboratory：Google Colaboratory 是一個免費的 Jupyter 筆記本環境，不需要進行任何設置就可以使用，且完全在雲端運行。借助 Colaboratory 可以編寫和執行代碼、保存和共享分析結果，以及利用 Google 強大的計算資源。由於神經網路需要較強的運算能力，我們使用 Colaboratory 的 GPU 資源測試神經網路的部分。

硬體規格：

(一) 作業系統：Windows 10

(二) CPU：Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz

Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz

(三) 記憶體：8.00GB

二、 文獻探討

心理學上（Ekman, Sorenson, & Friesen 1969）[1]將情緒分為離散的六種，分別是快樂（happiness）、傷心（sadness）、憤怒（anger）、恐懼（fear）、驚喜（surprise）、厭惡（disgust）。而語音情緒辨識這個分類問題就是研究如何讓電腦正確判斷一段語音屬於上述六種情緒的哪一種。

Zhao、Ye 與 Wan（2018）在論文[2]中將語音情緒分類的方法略分成三種，分別是：

傳統語音情緒辨識：

先由人工選取要使用的語音特徵，接著以語音處理工具提取特徵，最後丟入傳統分類器中進行情緒分類。

如 Ooi、Seng、Ang 與 Chew（2014）[3]即是使用傳統語音情緒辨識的方法進行辨識。他們選取的語音特徵有過零率（Zero Crossing Rate, ZCR）、音量（Volume）、梅爾頻率倒譜係數（Mel-Frequency Cepstral Coefficients, MFCC）、音高（Pitch）、Teager 能量運算子（Teager Energy Operator, TEO）。

神經網路語音情緒辨識：

特徵亦由人工選取，但提取後使用神經網路進行分類，優點是經過較多非線性轉換，因此模型可保留更多提取的特徵，達到更好的分類效果。

端點對端點（end-to-end）：

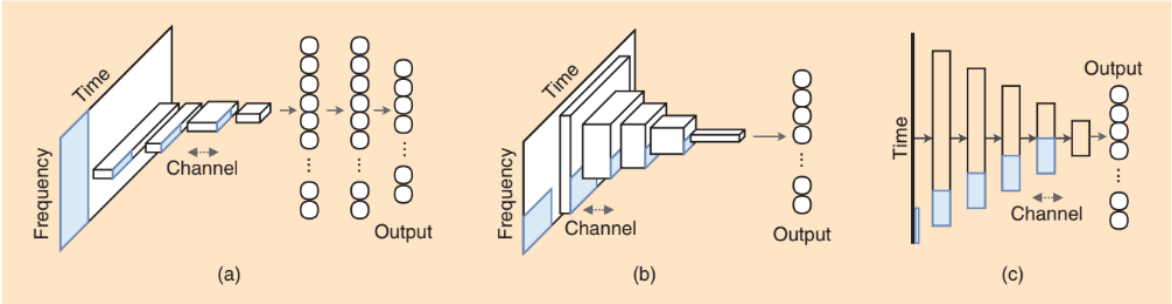
Nam、Choi、Lee、Chou 與 Yang（2019）的研究[4]整理了三種音訊分類常用的 CNN 模型，分別為：

(一) 一維卷積神經網路（1-D CNNs）

(二) 二維卷積神經網路（2-D CNNs）

(三) 取樣點級別卷積神經網路（Sample-Level CNNs）

其中 1D-CNN 及 2D-CNN 的輸入是語音檔轉換成的頻譜圖，而 Sample-Level CNN 的輸入則是原始訊號強度，也就是波形圖。



圖一、三種 CNN 模型示意圖（圖擷取自論文[4]）

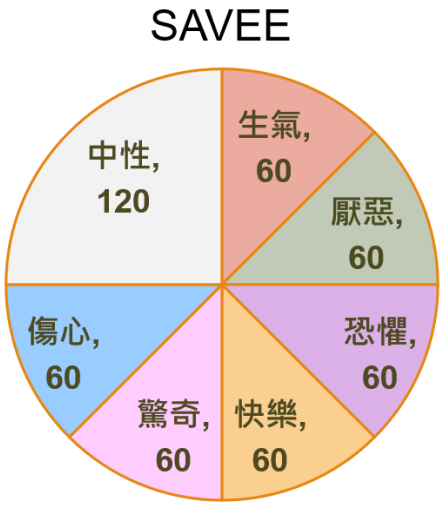
我們的研究選用了傳統語音情緒辨識中的 SVM 和端點對端點中的 CNN 兩種方法實作。

三、 現有語料庫探討

(一) Surrey Audio-Visual Expressed Emotion（SAVEE）Database（Jackson & Haq 2009）[5]

薩里大學錄製的語音-影像資料庫（本次僅使用到語音資料庫）。

1. 錄製語言：英文。
2. 情緒分類：七種情緒，分別為：憤怒（anger）、厭惡（disgust）、恐懼（fear）、快樂（happiness）、驚奇（surprise）、傷心（sadness）、中性（neutral）。這七種分別為心理學家 Ekman 等人（1969）提出的六種情緒分類加上做為比較基準的中性情緒[1]。
3. 錄製人員：四名該校的研究生（非專業演員），皆為男性，年齡範圍為 27-31 歲。
4. 資料筆數：除中性情緒，每種情緒由每位錄音員各錄製 15 筆資料，中性則是每人各 30 筆。四人加總後共有 480 筆資料，分為音檔及影像檔。
5. 資料內容：每筆資料均為一個句子的錄音或錄影。
6. 發表年分：2009 年。

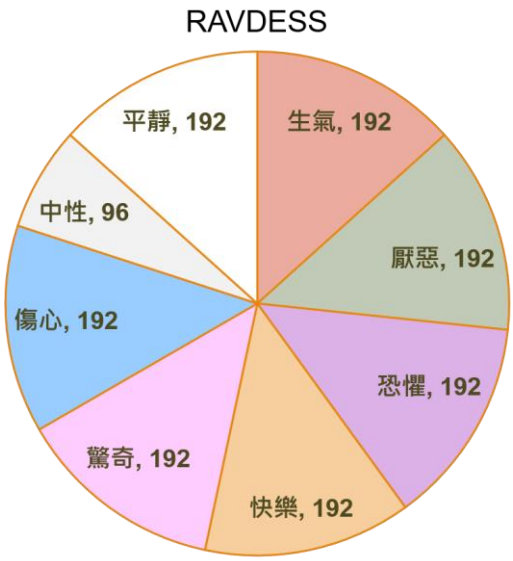


圖二、SAVEE 資料庫情緒筆數比例圖（單位：筆數）

(二) Ryerson Audio-Visual Database of Emotional Speech and Song（RAVDESS）（Livingstone & Russo 2018）[6]

這是懷雅遜大學錄製的語音-影像資料庫（本次僅使用到語音資料庫），這個資料庫除了說話外，還有歌唱的資料，但我們這次並沒有使用歌唱的部分。

1. 錄製語言：英文。
2. 情緒分類：八種情緒，分別為：快樂（happy）、傷心（sad）、 憤怒（angry）、恐懼（fearful）、驚奇（surprise）、厭惡（disgust）、 平靜（calm）、中性（neutral）。這八種包含與 SAVEE 資料庫相同的七種情緒以及 RAVDESS 資料庫新增的平靜項目。平靜是中性外的另一情緒基準，增設的目的是改善其他資料庫的中性資料易被誤判為負面情緒的缺失。
3. 錄製人員：二十四名專業演員，男女各半，年齡為 21-33 歲。
4. 資料筆數：每種情緒皆由每位錄音員分別以正常及強烈兩種強度錄製兩句話（中性沒有分強度），所有錄音員的資料加總後共有 1,440 筆錄音。
5. 資料內容：每筆資料均為一個句子的錄音或錄影。
6. 發表年分：2017 年。



圖三、RAVDESS 資料庫情緒筆數比例圖（單位：筆數）

表一、SAVEE 及 RAVDESS 資料庫比較

	SAVEE 資料庫	RAVDESS 資料庫
語言	英文	英文
情緒標註	7 種 憤怒、厭惡、恐懼、快樂、 驚奇、傷心、中性	8 種 憤怒、厭惡、恐懼、快樂、 驚奇、傷心、中性、冷靜
錄音員人數	4 位	24 位
資料筆數	480 筆	1440 筆
資料內容	句子	句子
發表年分	2009	2017

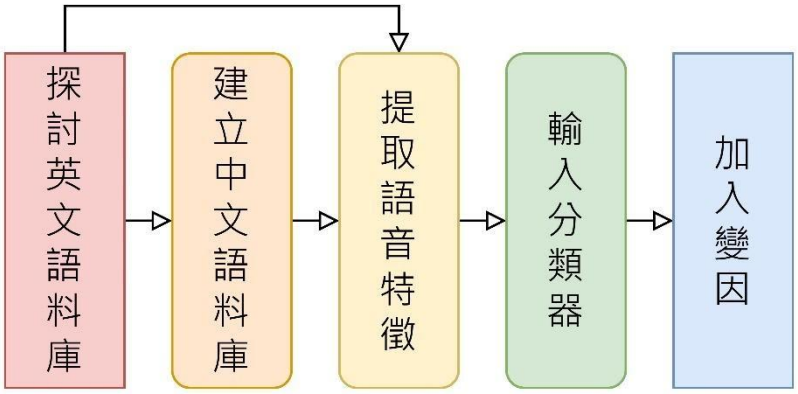
(三) 台灣地區華人情緒與相關心理生理資料庫—基本情緒聲調（黃世琤、李明純、李麗雯、詹雅婷、蔡鑫廷，民 103）[7]

這是國立中正大學心理學系暨認知科學研究中心錄製的語音-影像資料庫。

1. 錄製語言：中文。
2. 情緒分類：七種情緒，分別為：快樂（happiness）、悲傷 （sadness）、生氣（anger）、厭惡（disgust）、害怕 （fear）、驚訝（surprise）和輕蔑（contempt）。
3. 錄製人員：八十名未曾參與過廣播戲劇等相關訓練或表演之參與者，男女各半，年齡為 19-28 歲。
4. 資料筆數：經研究單位挑選後共收錄 424 段音檔。
5. 資料內容：每筆資料均為一個句子的錄音或錄影。

這個資料庫是目前唯一的中文語音情緒資料庫，然而其資料量並不多，且部分資料標註有錯誤，因此我們想要自己另外再建立一個中文語音情緒資料庫。

四、 研究方法



圖四、研究流程圖

(一) 建立中文語料庫

目前中文語音情緒辨識的研究並不多，也找不到已公開釋出的完整中文情緒語料庫，因此我們想嘗試自己建立一個中文語料庫。

1. 資料選擇

開始建立資料庫時，我們考慮了幾種語料來源，包括使用網路上的影音資源或自己錄製中文語料。

表二、不同資料來源比較

	自行錄製音檔	上網蒐集語音資源
音檔源	我們二人、同學	影音平台
情緒標註	原錄音時指示的情緒	本研究作者標註
句子設計	參考英文語料庫的語句， 再自己設計適合的中文語句	選擇有明確情緒的語句， 每筆資料語句內容皆不同
遇到難題	錄音品質不穩定 （硬體設備、錄音員技術差異）	耗費人力

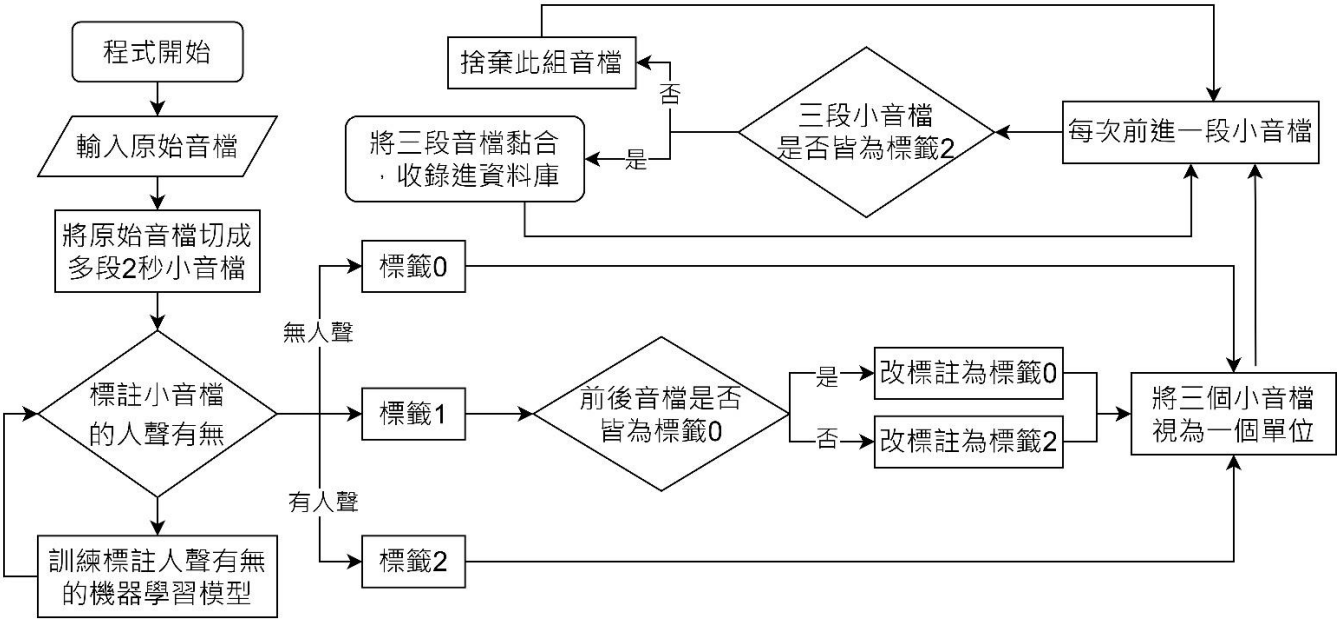
一開始我們選擇使用自行錄製音檔，但預錄時我們發現難度太高。遇到的困難包括硬體設備不夠精密，導致錄音品質低、錄音員非專業演員，因此情緒表現不清楚以及資料蒐集太慢等，所以最後改蒐集現有網路上的語音資源。

表三、不同網路資料來源比較

	卡通	綜藝節目、球賽轉播	電視劇
優點	發音清楚明確	情緒真實	語音品質高 正負面情緒皆具
缺點	語句短促 負面情緒較少 對話交替速度過快	背景雜音較多 語音較不清楚 負面情緒較少	有背景音樂

我們比較了不同類型網路資源的優缺點，最終因為後宮甄嬛傳在網路上可取得的資料量較大，所以我們選擇使用該電視劇。

2. 資料處理



圖五、資料處理流程圖

原始音檔

我們使用上述的電視劇作為音訊資料來源，且將完整一集電視劇的音檔（約 30~40 分鐘）稱為一筆原始音檔。這次研究中，共使用了三筆原始音檔。

切割成兩秒與標註人聲有無

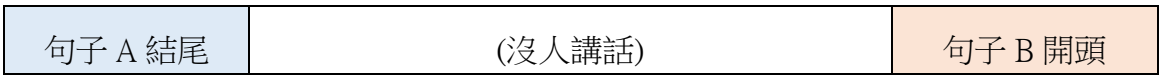
電視劇的原始音檔中，能以「是否有人聲」分為兩類。由於語料庫只採用有人聲的部分，因此為方便後續資料標註，需先將沒有人聲的段落（背景音樂或噪音）去除。

作法是將原始音檔切割成等長的段落（以下將每個段落稱做小音檔），並分別將其標註為以下三類：

表四、三種人聲類別

標籤	內容
0	無人聲
1	不到一個完整的字發音
2	有人聲

一開始，小音檔的長度被設定為六秒，但實際聽過後，發現六秒時間太長，時常無法用一個標籤精準地表達音檔的狀況。例如：若小音檔開頭有人說了幾個字，結尾又有人說了一小段話，但中間數秒沒有人聲，如圖六。這種音檔時常有一半以上沒有人聲，標註為類別 2 會造成後續情緒判讀時收錄太多背景音樂等雜訊。但標為類別 0，則之後的情緒標註就會損失許多資料。若一律標註為類別 1，則除了與標註為類別 2 相同的雜訊問題，也會造成類別 1 小音檔中有說話部分長度參差不齊。因此將其分在上述三類中任何一類都不適合。



圖六、六秒音檔太長示意圖

於是我們改嘗試以兩秒作為小音檔的長度，發現這個長度較為合適，小音檔大多可以明確分在類別 0 或 2，少數類別 1 的小音檔也不會收錄過多雜訊。

黏合成六秒及標註情緒

因為兩秒的長度無法明確判斷情緒，所以我們接著要將小音檔連接成較長的音檔以進行情緒的標註。因為之前在標註六秒音檔是否有人聲時發現六秒的音檔雖然不適合標註人聲有無，但可以判斷情緒，因此此步驟就決定以六秒為情緒標註音檔的長度。

以下是黏合的步驟，為方便說明，會以下方這些音檔做舉例，其中每個格子代表一筆小音檔，數字是人聲有無的標註結果：

0	2	2	2	2	1	0	0	1	0	2	2	2	1

圖七、一排小音檔範例

(1) 處理類別 1：

若一筆類別 1 的音檔的前或後一筆音檔是類別 2，那此段音檔可能是長句子的前端尾端，故將此段改標註為類別 2。

反之，若一筆類別 1 的音檔前後皆是類別 0，則表示此段音檔可能僅含有小於 1 秒鐘的句子，而小於 1 秒的句子也較不易　判斷明確情緒，故可將此段音檔直接捨棄，改標註為類別 0。

如下方這兩排音檔中，第一排是原本的標註，第二排則是調整過的結果，紅色部分為經過這個步驟改動後的標註結果。

處理前	0	2	2	2	2	1	0	0	1	0	2	2	2	1
處理後	0	2	2	2	2	2	0	0	0	0	2	2	2	2

圖八、處理類別 1 前後示意圖

(2) 黏合音檔：

從頭搜尋，每次檢查連續的三筆音檔，如果三筆都是類別 2，我們就會將它們黏合並收錄進資料庫中。下方示意圖中，每組橘色的音檔串會成為資料庫中的一筆資料。

接著往前移一個音檔長（2 秒），並重複上述檢查，直到達到結尾。

0	2	2	2	2	2	0	0	0	0	2	2	2	2
0	2	2	2	2	2	0	0	0	0	2	2	2	2
0	2	2	2	2	2	0	0	0	0	2	2	2	2
0	2	2	2	2	2	0	0	0	0	2	2	2	2
0	2	2	2	2	2	0	0	0	0	2	2	2	2

圖九、黏合音檔示意圖

(3) 資料標註：

我們預計將經過上述處理的六秒音檔整理成資料庫並進行標註。我們設計了五種標籤，如下表所示：

表五、三種情緒標籤

標籤	內容
negative / neg.	負面情緒
neutral / neu.	中性/沒有情緒
positive / pos.	正面情緒
multiple / multi.	多人同時講話
both	正負面情緒皆有

因為中性情緒無法精準定義，一段音檔是否標註為 neu.較難判斷，所以我們盡量避免將一般對話標註在 neu.，會標註為 neu.的情況主要是純粹轉達訊息時（如：太監宣讀聖旨）。

另外，multi.及 both 的設置是因應一些較不容易分類的狀況。multi.是指有多人同時說話，這種情況下無法判別出一位主要能代表情緒的發言者。而 both 則是指一段音檔中正負面情緒都有出現且長度差不多，以至於無法分在 neg.或 pos.。

(二) 提取語音特徵

1. 梅爾頻率倒譜係數（MFCC）

MFCC 是一種在音訊上廣泛使用的聲學特徵，其特色為考慮到人耳在不同頻率的聽覺特性。提取的步驟如下：

(1) 預強調（Pre-emphasis）：將語音訊號通過高通濾波器（High-pass filter），目的是消除聲帶和嘴唇的影響。

(2) 音框化（Frame blocking）：將 N 個（N 為一固定正整數，通常取 2048）取樣點集合成一個音框單位，相鄰音框間有重疊，以保持音框的連續性。

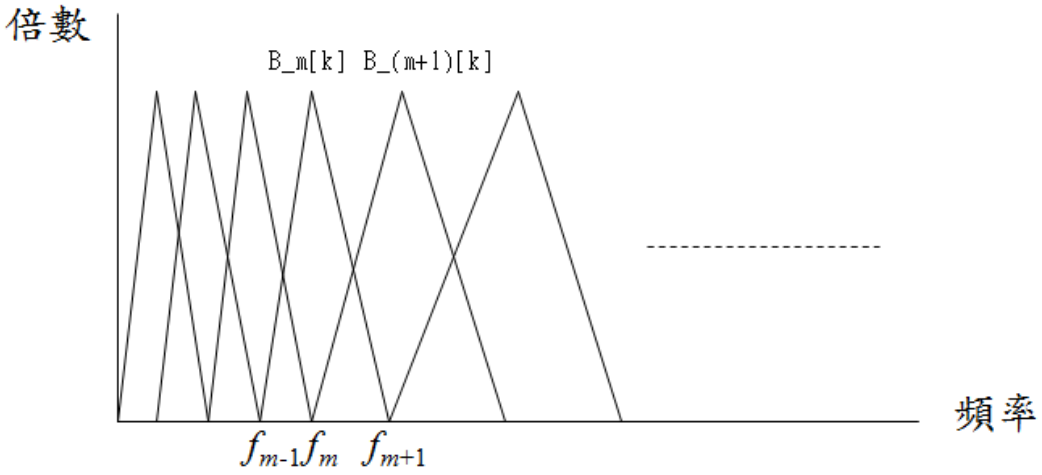
(3) 漢明窗（Hamming window）：將每個音框乘上漢明窗，可強調音框中間的部分，以增加左右音框的連續性。

(4) 快速傅利葉轉換（Fast Fourier Transform, FFT）：將音訊從時域（Time Dormain）轉至頻域（Frequency Dormain），以觀察各頻帶的能量分布。

(5) 梅爾頻譜濾波器（Mel-frequency filter bank）：以梅爾三角帶通濾波器，

得到梅爾刻度，以模仿人耳對頻率的感受。梅爾刻度是一個根據人耳對不同頻率敏感度不同（頻率越低人耳越能辨別頻率不同）而建立的非線性頻率刻度。梅爾刻度（m）與赫茲（f）換算的公式如下：

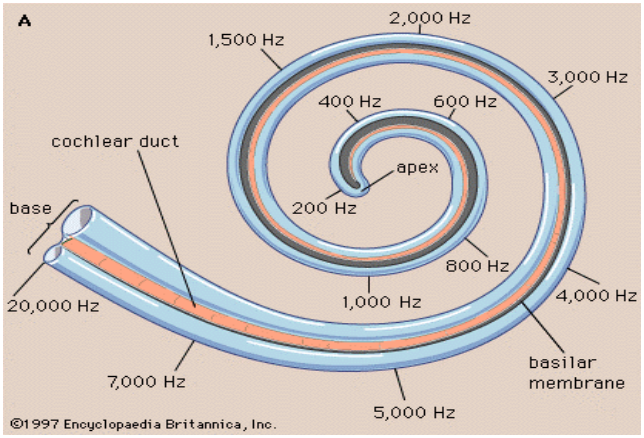
$$m = 2595 \log_{10}(1 + \frac{f}{700})$$



圖十、梅爾三角帶通濾波器示意圖

(6) 對數能量（Logarithm energy）：透過對數運算將音量壓縮，除去語音訊號在相位上的變化。

(7) 離散餘弦轉換（Discrete Cosine Transform, DCT）：將訊號轉換為倒頻譜係數，用意在於減少維度間的關係。

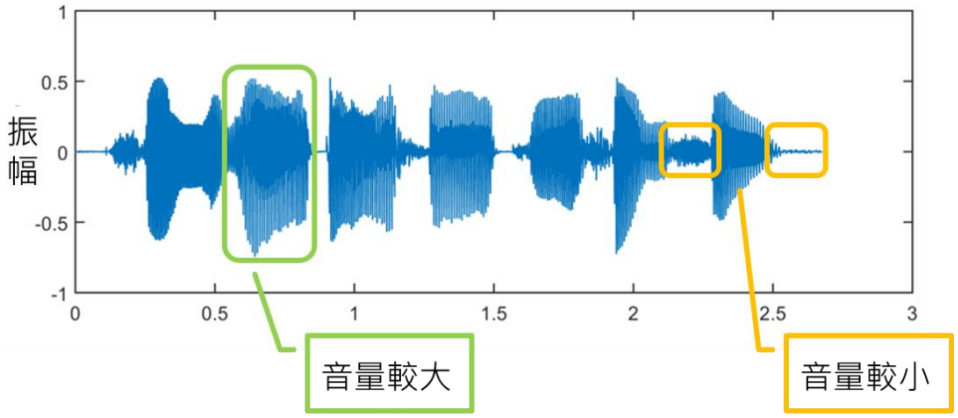


圖十一、人耳對不同頻率感知示意圖[5]

我們使用 Librosa 函式庫的 librosa.feature.mfcc 提取出前 13 個 MFCC 係數，並依照普遍處理 MFCC 的方法，對其做一次微分、二次微分，最後共得到 39 個特徵參數。

2. 音量（Volume）

音量是音訊能量的對數值，以分貝（dB）為單位。



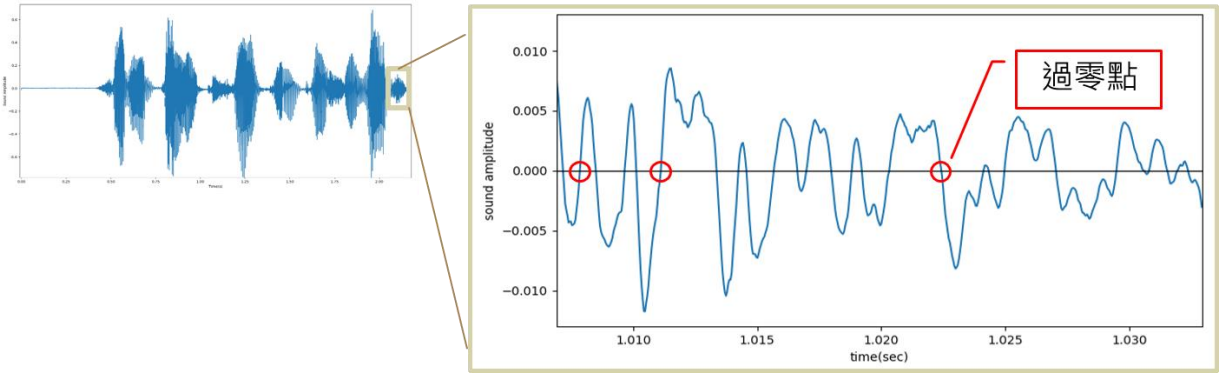
圖十二、音量示意圖

我們使用 Librosa 的 librosa.core.amplitude_to_db 提取出音檔的音量，再對其取平均、中位數及標準差，共得到三個特徵參數。

3. 過零率（ZCR）

過零點是一個音框中，音訊通過波型圖零點的次數。

過零率則為一音框中的過零點次數總和除以音框長度。



圖十三、過零點示意圖（圓圈內為其中三個過零點）

我們使用 Librosa 的 librosa.feature.zero_crossing_rate 提取出來音檔的 ZCR，再對其取平均、中位數及標準差，共得到三個特徵參數。

Librosa 是一個用來分析音樂和音頻檔案的 Python 函式庫，功能除了提取上述特徵外，還有計算短時距傅立葉轉換、提取音高等許多功能。

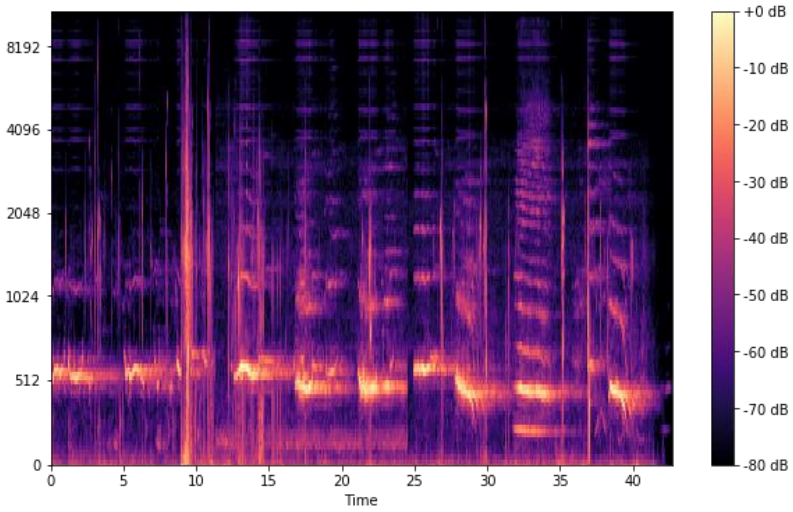
合併以上三種特徵，每筆音檔共可得 45 個參數。我們將它們合併為一個 45 項的陣列，再將所有音檔的特徵陣列連接為一個「音檔數*45」的矩陣，輸入 SVM。

4. 梅爾頻譜圖（Melspectrogram）

頻譜圖為音訊經過快速傅立葉轉換後所得的頻率-時間關係圖，梅爾頻譜圖則是以梅爾刻度取代原本的赫茲作為單位，這麼做是為了更貼近人耳對聲音的感受。圖中每一格的資料是一個數字，代表該頻率區間在該時間的音量大小。

我們以 Librosa 的 librosa.feature.melspectrogram 提取梅爾頻譜圖。

提取出梅爾頻譜圖後，會得到一個「梅爾刻度（128）*時間」的矩陣。合併所有音檔的梅爾頻譜圖則成為一個「音檔數*梅爾刻度（128）*時間」。我們將它當作 CNN 的輸入。



圖十四、梅爾頻譜圖（顏色為電腦套色，顏色越淺代表音量越大）

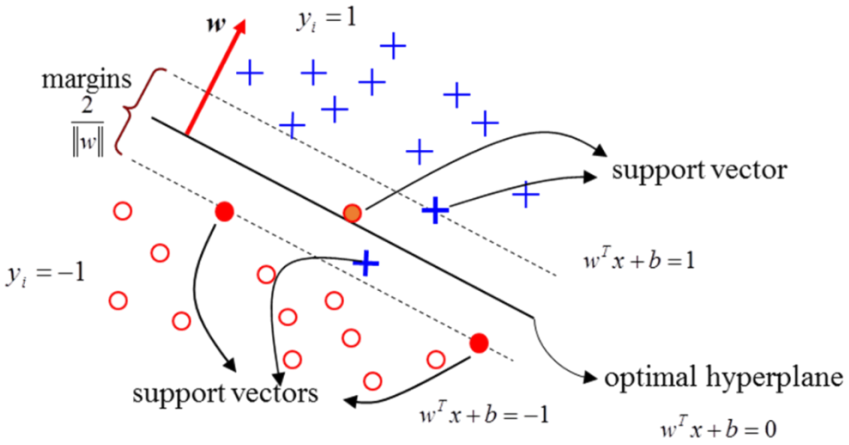
(三) 選擇分類器

1. 支持向量機（SVM）

很多研究（如 Chenchah & Lachiri 2015 [7]）指出，在聲學判斷方面，SVM 是個十分好用的分類器。

SVM 是一種監督式的學習方法，可以根據給定的資料特徵決定一個高維度的超平面，以將不同類別的資料分開，而這個超平面距離不同類別的邊界會是最大的。

我們使用 Scikit-learn（Sklearn）當中的 SVC 函式來完成 SVM 模型訓練及測試。Sklearn 是適用於 Python 的機器學習工具庫，有許多常見的機器學習模型。



圖十五、支持向量機

2. 卷積神經網路（CNN）

近年來使用神經網路做機器學習十分盛行，加上許多研究（如 Mustaqeem & Kwon 2019 [8]）指出，在聲學判斷方面，CNN 也是個不錯的分類器，因此我們決定除了 SVM，也用 CNN 做測試。

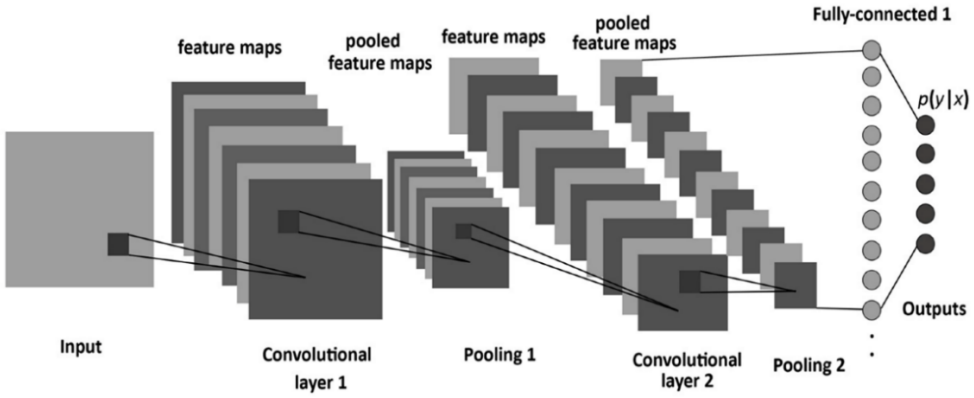
CNN 是神經網路的一種變化，由許多卷積層（Convolution Layers）和池化層（Pooling Layers）排列重疊組成，最後用全連接層（Fully-Connected Layers）對學習到的特徵向量進行最後的整合。

(1) 卷積層：具備如濾鏡的功能，可使某些特徵更加明顯。通過卷積層輸出的圖片稱為特徵圖。1D-CNN 及 2D-CNN 的差別就在於卷積層是一維或二維。

(2) 池化層：為了避免因為部分神經元太過活躍，使得特定區域的計算量過大，所以加入池化層，找出局部的特殊值。池化層有平均池化層與最大池化層等，一般選用最大池化層。

(3) 全連接層：在全連接層中的每一個神經元都會和上一層的所有神經元連結，故計算量較大。不同於卷積層，經過全連接層計算的值不會受到學習到的特徵所在位置的影響。

我們這次研究中使用 Python 的 Keras 和 Tensorflow 套件達成 1D 及 2D 的 CNN 模型訓練及測試。



圖十六、卷積神經網路

表六、CNN、SVM 比較

	特徵	需要的資料量	訓練時間
SVM	MFCC、ZCR、Vol	較少	較短
CNN	由模型自行學習	較多	較長

(四) 不同資料處理方式

1. 標註方式

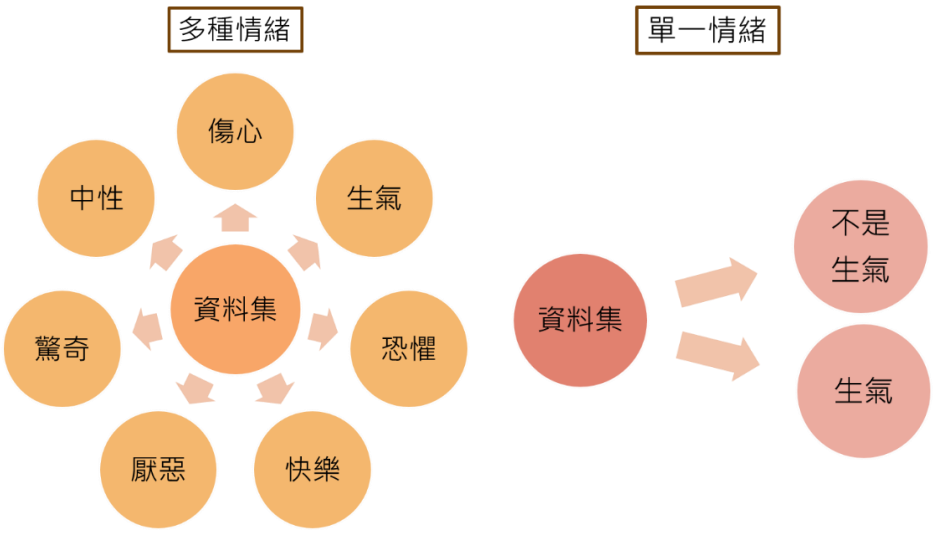
多種情緒

使用資料庫中的原始標註，訓練分類器一次分辨所有情緒。

單一情緒

針對每個情緒輪流將所有情緒的標籤改為「是某一情緒」及「非某一情緒」，以計算出分類器針對某單一情緒的分類正確率。由於中性情緒是用來對比其他情緒的對照組，所以在單一情緒項目中，我們並沒有做中性情緒的分類辨識。

大部分的語音情緒辨識研究使用的是「多種情緒」標註。但是有些情況並不需要一次判斷那麼多種情緒（如：生命線的輔導員在和諮詢者進行對話時，電腦可及時提供輔導員諮詢者當下的情緒，有利於輔導員進行引導，此時情緒的偵測可著重於判斷負面情緒）。另外，我們也想知道，如果分類器只需要分兩類，得出的正確率是否會有提升。



圖十七、標註方式示意圖

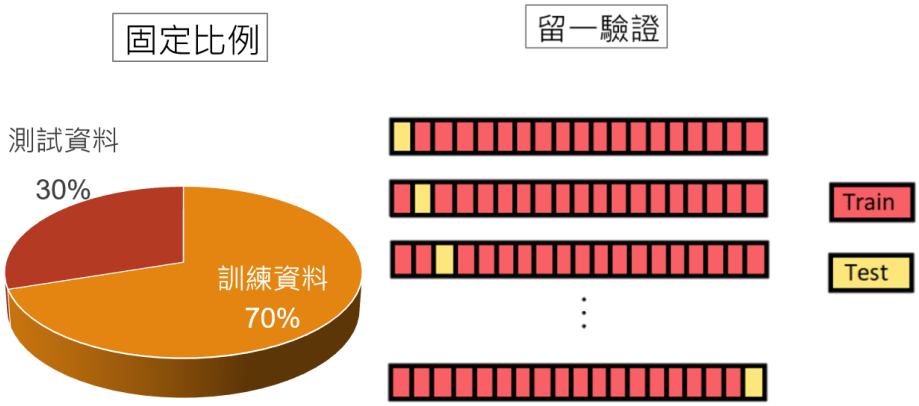
2. 訓練方式

固定比例

預先設定好訓練資料和測試資料筆數的比例，再隨機抽選固定筆數的資料做為測試資料。

留一驗證（leave-one-out）

將每筆資料輪流當作測試資料，其餘做為訓練資料，再統計分類器對於每筆測試資料的預測結果，以得到分類器的分類正確率。



圖十八、訓練方式示意圖

大部分的語音情緒辨識研究使用的是「固定比例」方式。但我們擔心資料庫的資料筆數過少，無法得出好結果，所以想出使用「留一驗證」的方式彌補資料及大小不足的問題。

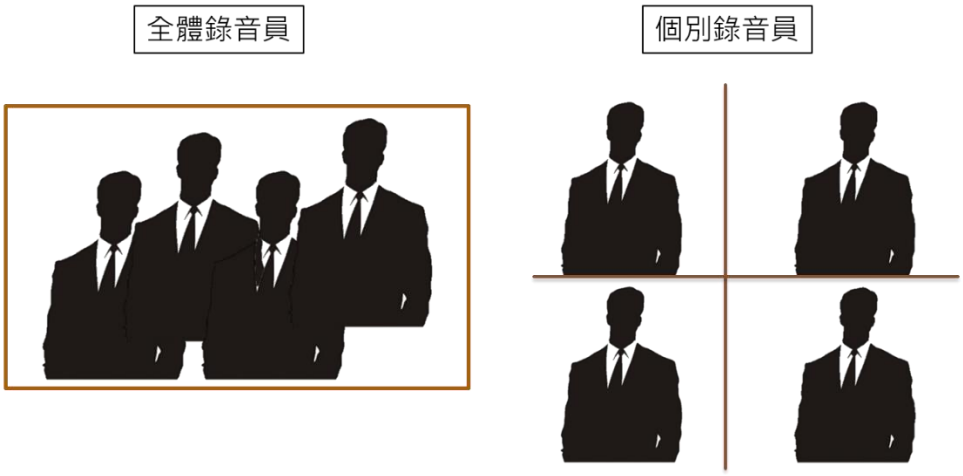
3. 選用資料

全體錄音員

即以整個資料庫做為此組實驗的資料。

個別錄音員

即輪流選用某一資料庫的某一錄音員做為此組實驗的資料。



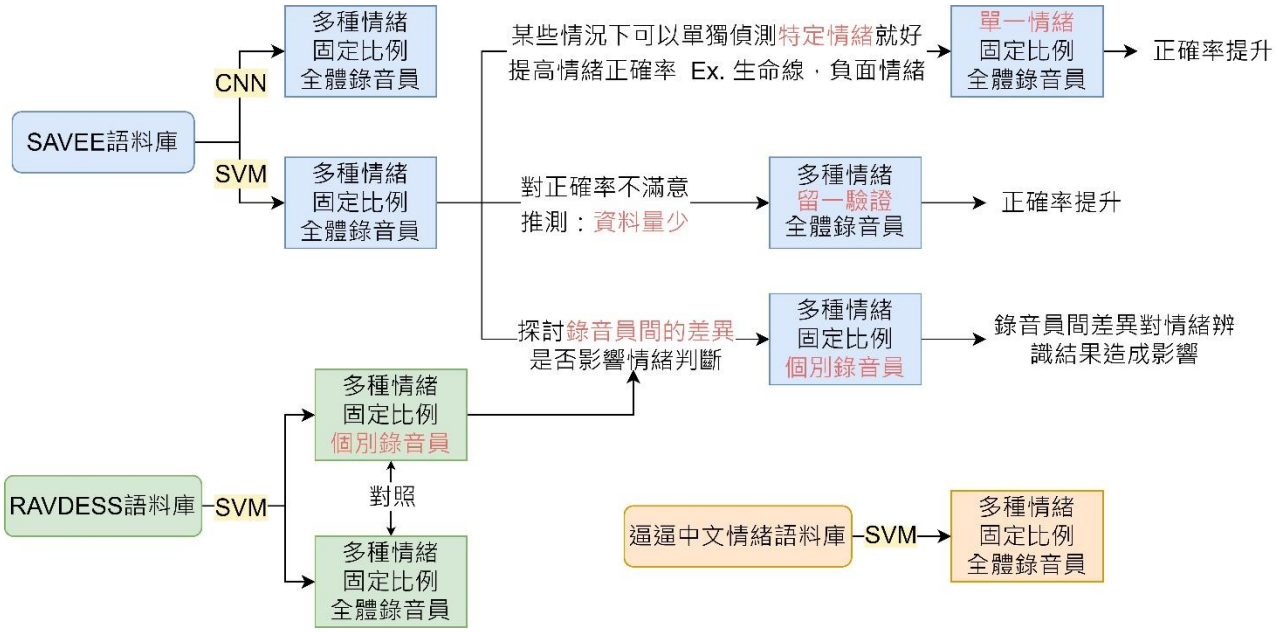
圖十九、選用資料示意圖

我們想知道錄音員間的個別差異是否會影響情緒判斷的正確率，所以設計出選用資料的這項實驗。

(五) 計算正確率、畫出結果圖

正確率 $= \frac{\text{測試結果為正確的資料筆數}}{\text{總資料筆數}} \times 100\%$。其中，測試結果為正確的資料為：訓練模型時給定的標籤和模型預測結果相同者。

參、研究結果與討論



圖二十、研究結果架構圖

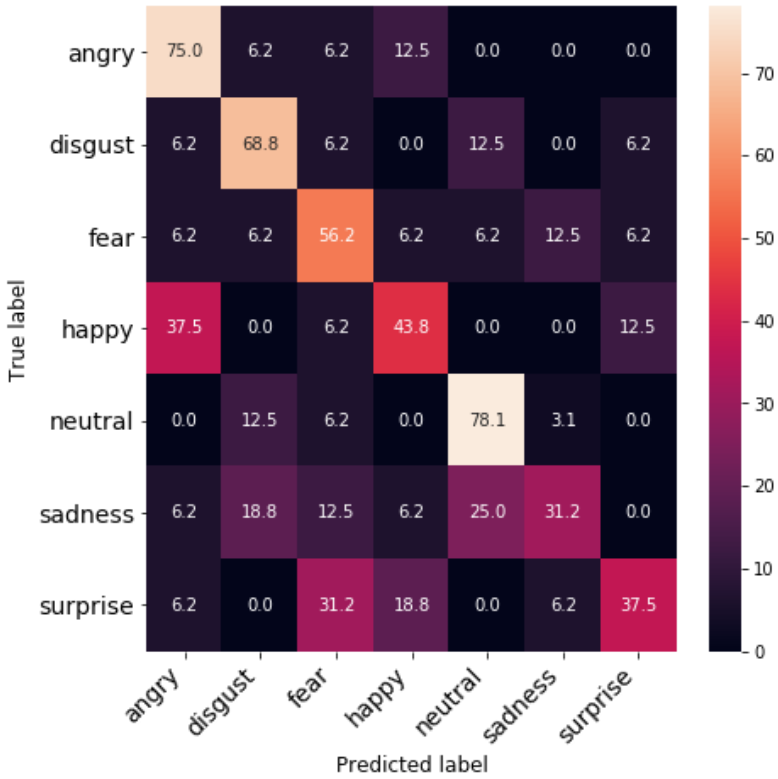
表七、中英文情緒標籤對照表

中文情緒標籤	SAVEE 資料庫 英文情緒標籤	RAVDESS 資料庫 英文情緒標籤
憤怒	anger	angry
厭惡	disgust	disgust
恐懼	fear	fearful
快樂	happiness	happy
傷心	sadness	sad
驚奇	surprise	surprise
中性	neutral	neutral
平靜	---	calm

一、SAVEE 資料庫結果

(一) 使用 SVM 作為分類器

多種情緒-固定比例-全體錄音員

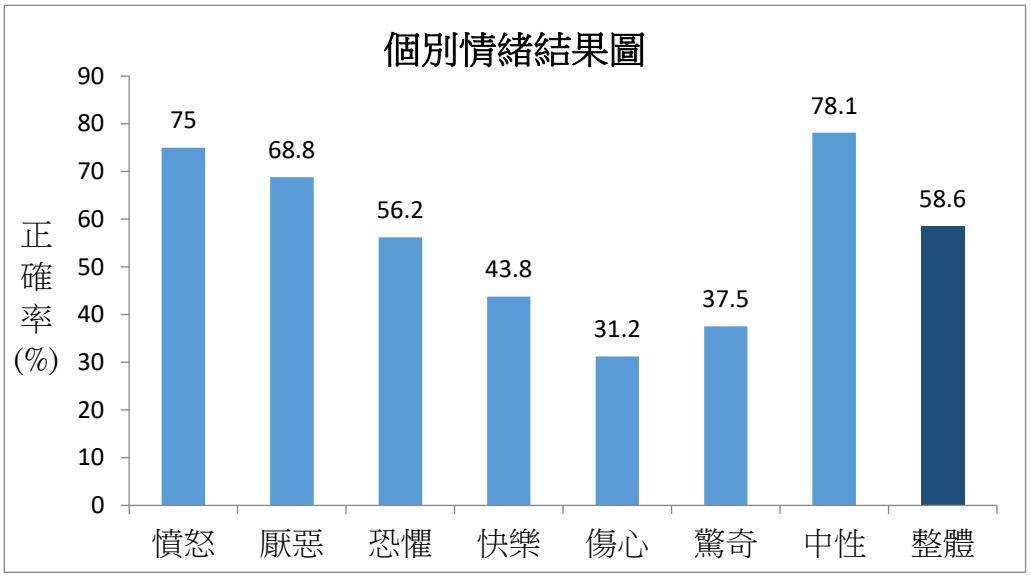


圖二十一、多種情緒-固定比例-全體錄音員混淆矩陣

（單位：%）（顏色越淺者正確率越高）

使用此組合（測試資料約佔總資料 26.67%）的訓練方法，正確率約為 58.6%。

個別情緒的正確率：（四捨五入到小數點第一位）



圖二十二、多種情緒-固定比例-全體錄音員個別情緒結果圖

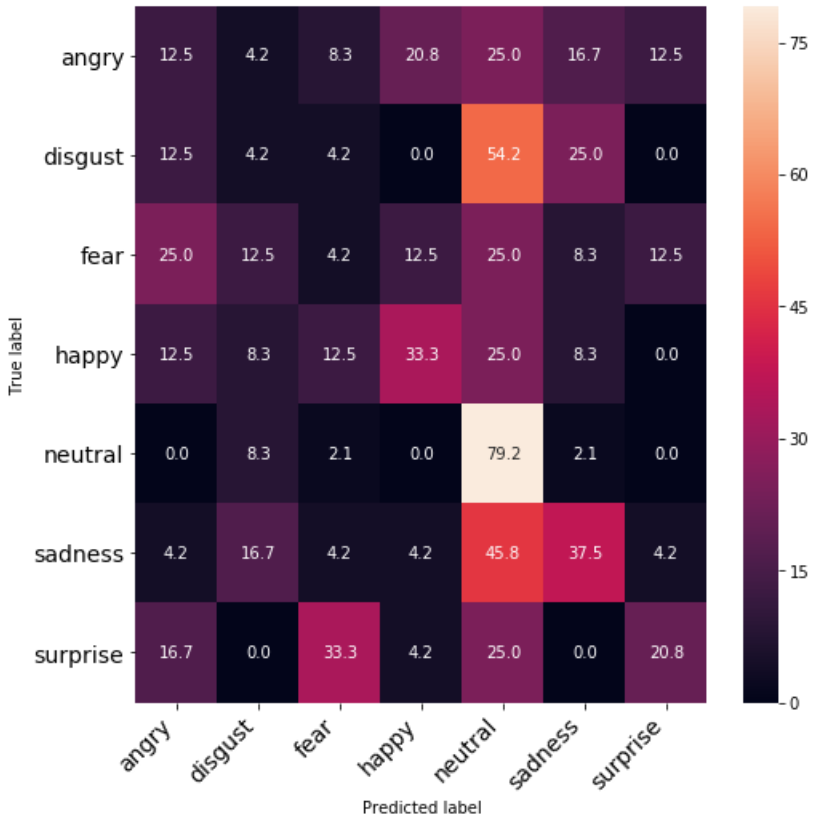
由上圖可知模型大致能夠辨別各個情緒（隨機猜測的正確率約為15.6%）。分析個別情緒，又以憤怒及中性的分類結果最好。由圖四則可發現被誤判成中性的資料都是厭惡、恐懼、傷心這三種負面情緒，分類錯誤的中性資料也都來自上述三種負面情緒。

(二) 使用 CNN 作為分類器

1. 比較 1D 及 2D 卷積神經網路

[4]中有提到 1D-CNN 易因相同圖案出現在不同頻率位置而產生十分不同的結果，導致正確率下降，2D-CNN 則較能避免此問題。因此，2D-CNN 的效果一般較 1D-CNN 為佳，而從我們自己的測試結果也發現 2D-CNN 效果的確優於 1D-CNN。除此之外，我們考量了測試資料數量、硬體效能等因素，綜合考量之下最後決定使用 2D-CNN 作為實驗主要使用的卷積神經網路。

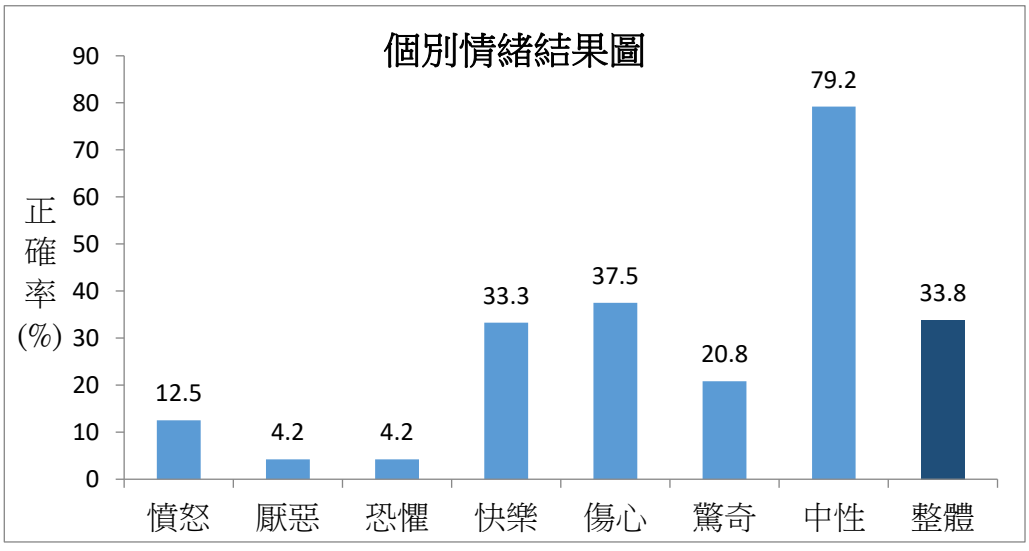
2. 多種情緒-固定比例-全體錄音員



圖二十三、多種情緒-固定比例-全體錄音員混淆矩陣（單位：%）

使用此組合（測試資料約佔總資料 26.67%）的訓練方法，正確率約為 33.8%。

個別情緒的正確率：（四捨五入到小數點第一位）

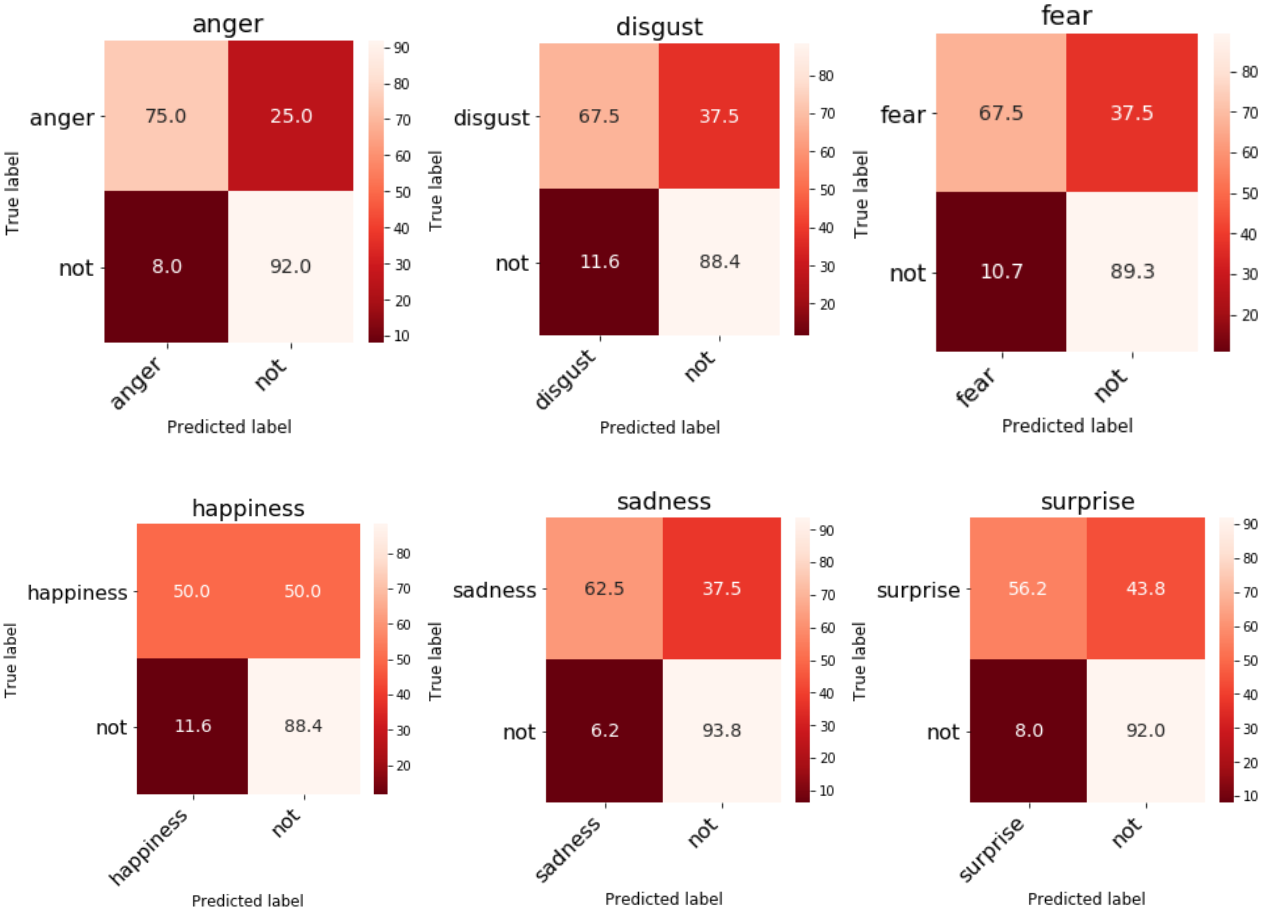


圖二十四、多種情緒-固定比例-全體錄音員個別情緒結果圖

由上圖可得知正確率最高的情緒是中性，不過從圖二十可以發現分類器將許多非中性的資料也預測為中性，因此中性的正確率較高並非因為中性情緒容易辨識，而是分類器傾向將一筆新資料預測為訓練資料中筆數最多的中性情緒。

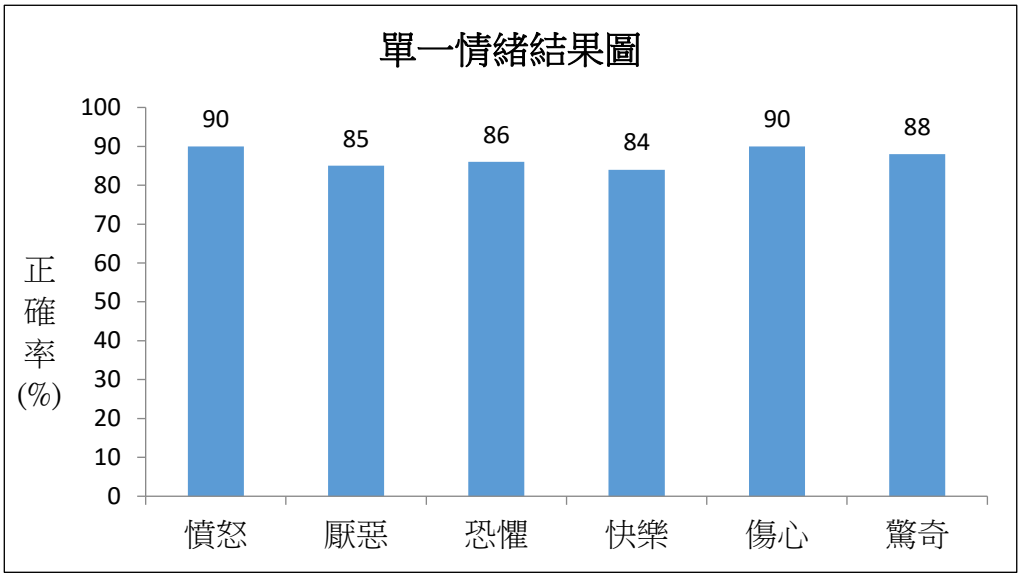
由上述結果可知，SVM 的結果表現明顯優於 CNN。因此，在之後的研究，我們皆使用 SVM 作為分類器。並且，為了更提升正確率，我們又設計了以下三種實驗：

(一) 單一情緒-固定比例-全體錄音員



圖二十五、單一情緒-固定比例-全體錄音員混淆矩陣（單位：%）

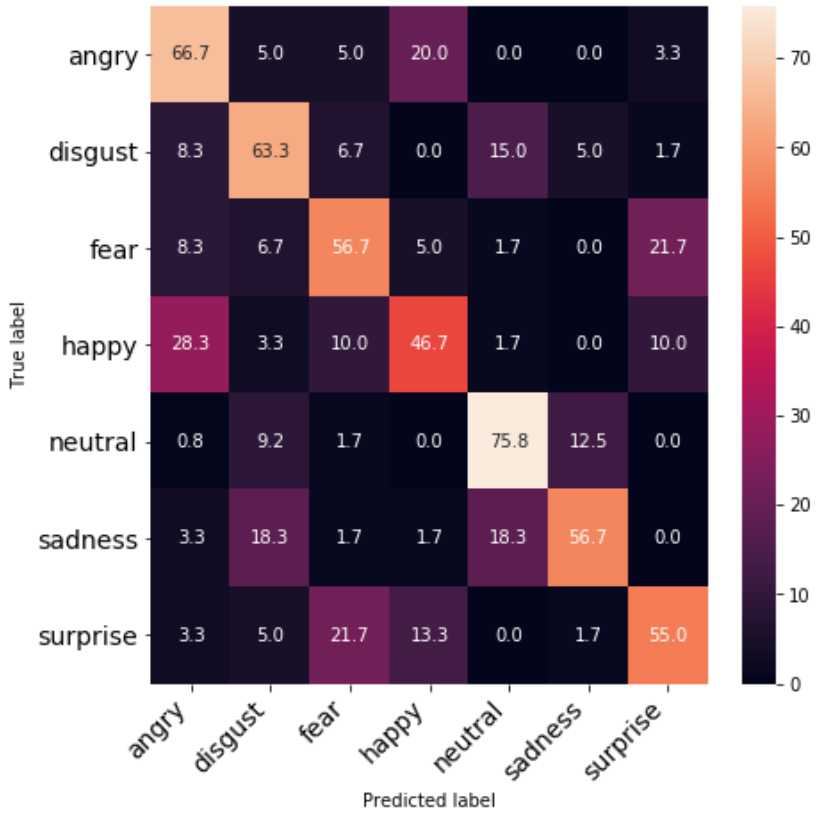
使用此組合的訓練方式，得到個別情緒的正確率分別為：（四捨五入到整數位）



圖二十六、單一情緒-固定比例-全體錄音員結果圖

由上圖可看到此實驗的正確率相比於多種情緒的正確率有所提升。

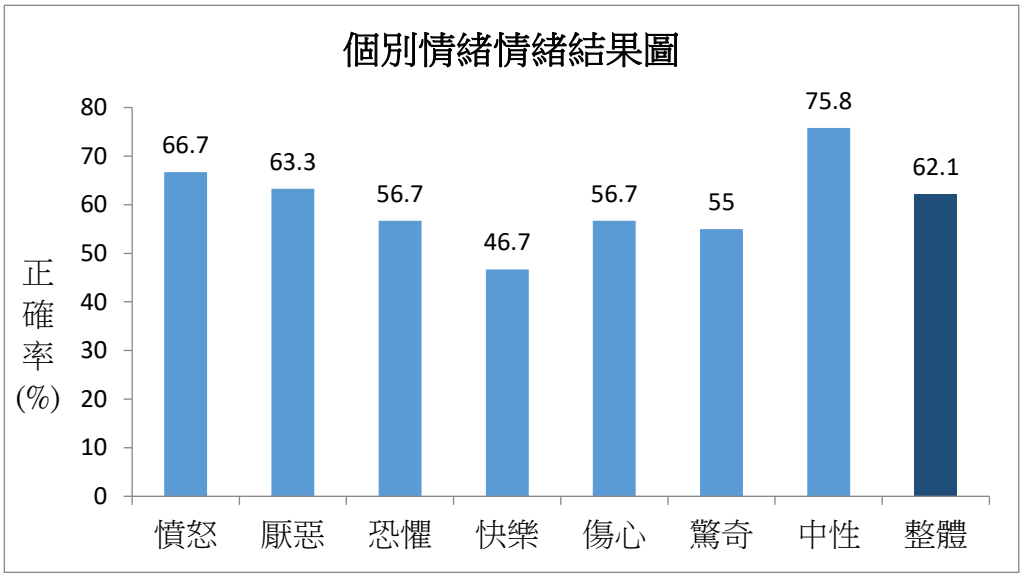
(二) 多種情緒-留一驗證-全體錄音員



圖二十七、多種情緒-留一驗證-全體錄音員混淆矩陣（單位：%）

使用此組合的訓練方式，測試 480 次後的每筆結果，得到總正確率為 62.1%。

個別情緒的正確率：（四捨五入到小數點第一位）

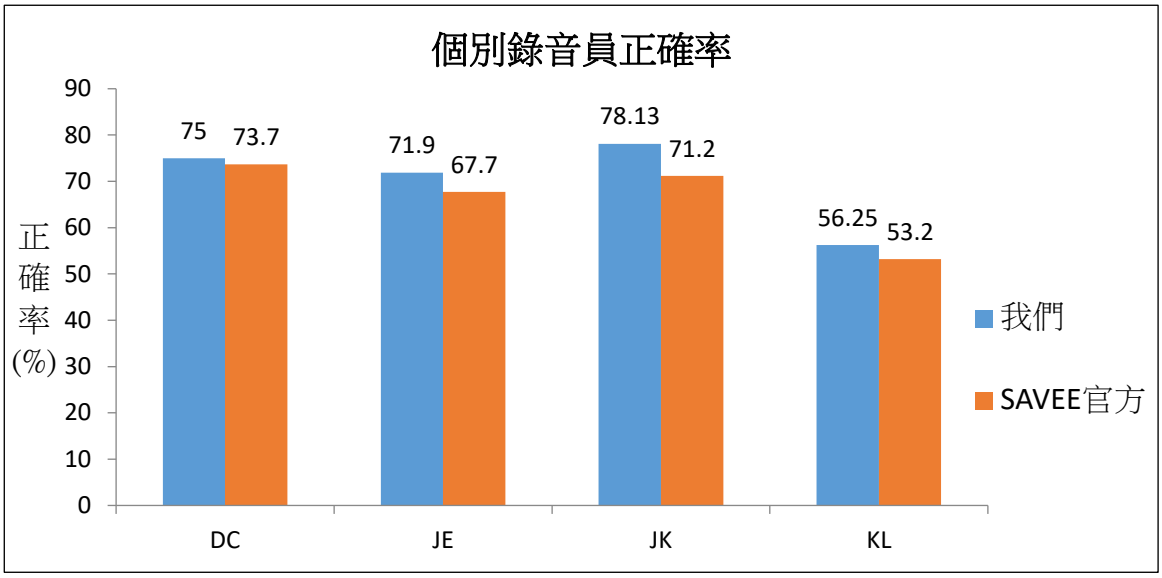


圖二十八、多種情緒-留一驗證-全體錄音員個別情緒結果圖

由上圖分析各情緒可知依然是憤怒、中性的分類結果最好。除此之外，整體的正確率相比於固定比例也有所提升。

(三) 多種情緒-固定比例-個別錄音員

除了將資料庫所有錄音員的錄音一起訓練（480 筆）外，我們還將不同錄音員所錄的資料分開訓練。SAVEE 資料庫中共有四位錄音員，其代號分別是 DC、JE、JK、KL，每人有 120 筆資料。我們將四人的錄音以固定比例的方式，分別以不同的參數訓練及測試，找到最適合每個人的參數組合。另外，我們也與 SAVEE 官網上人工判斷情緒的正確率做比較，作圖如下：



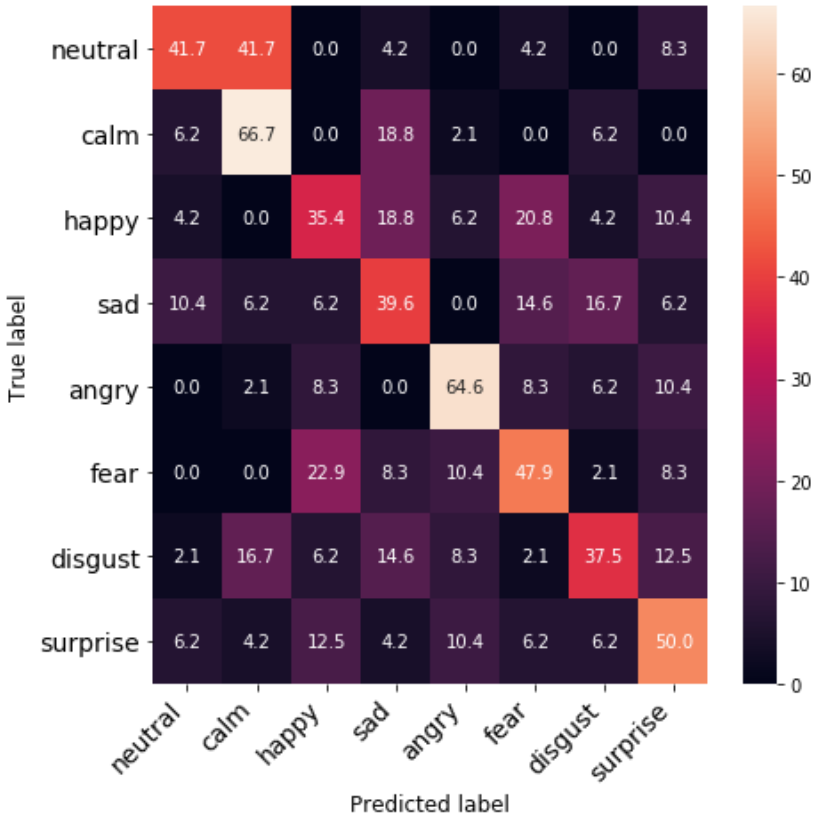
圖二十九、多種情緒-固定比例-個別錄音員正確率

二、RAVDESS 資料庫結果

為了驗證上述個別錄音員實驗的結果：「個別錄音員的正確率普遍高於全體錄音員、個別錄音員間的差異頗大」，我們又找了另一個錄音員數目較多的語料庫—RAVDESS，預期能得出相同的結果。

(一) 多種情緒-固定比例-全體錄音員

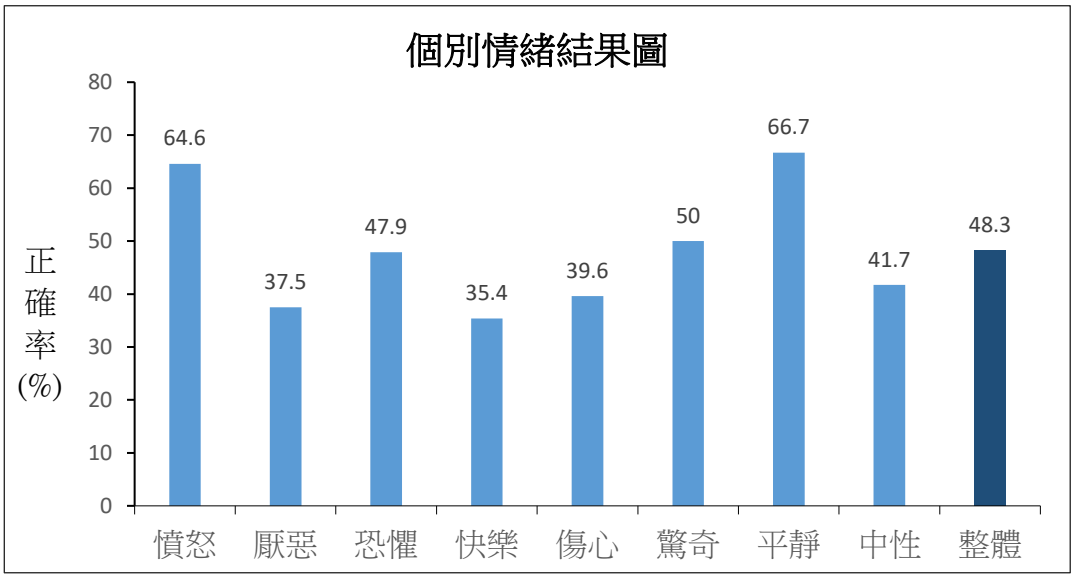
我們先做了全體錄音員實驗，當作比較的基準。



圖三十、多種情緒-固定比例-全體錄音員混淆矩陣（單位：%）

使用此組合（測試資料約佔總資料 26.67%）的訓練方法，正確率約為 48.3%。

個別情緒的正確率：



圖三十一、多種情緒-固定比例-全體錄音員個別情緒結果圖

總正確率為 48.3%（隨機猜的正確率：12.5%），由混淆矩陣得知，結果最好的情緒是冷靜，生氣情緒次之。

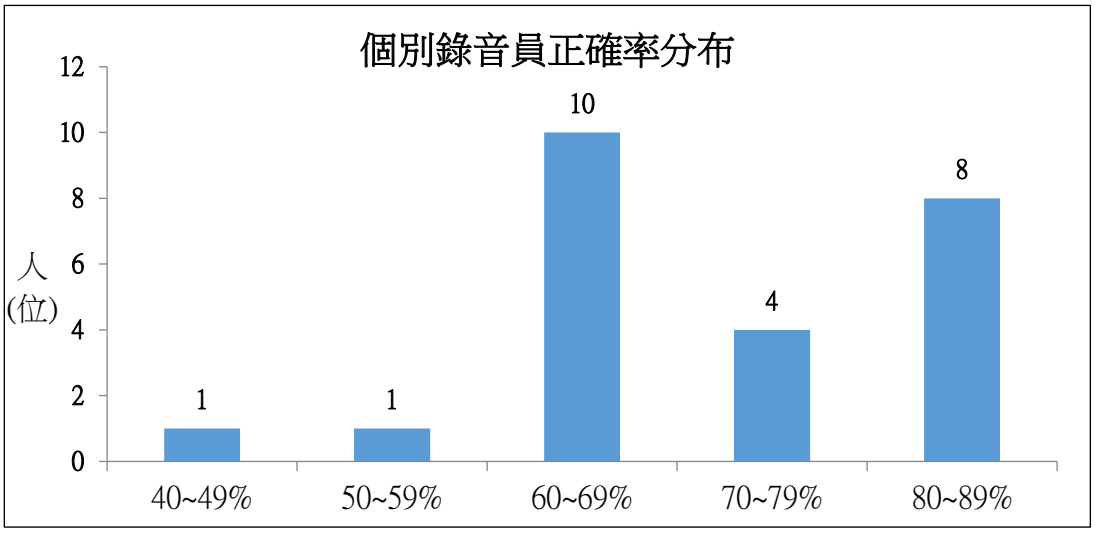
(二) 多種情緒-固定比例-個別錄音員

接著是個別錄音員的實驗結果（共有 24 位錄音員，每人錄有 60 筆資料），結果如下：

表八、RAVDESS 資料庫各錄音員正確率

錄音員編號（男）	正確率（%）	錄音員編號（女）	正確率（%）
1	86.7	2	80
3	53.3	4	73.3
5	73.3	6	80
7	60	8	73.3
9	80	10	66.7
11	66.7	12	60
13	73.3	14	60
15	60	16	46.7
17	80	18	80
19	60	20	80
21	86.7	22	60
23	66.7	24	60

下圖整理了二十四人的正確率分佈：



圖三十二、個別錄音員正確率分布圖（單位：位）

由表八及圖十六可知，RAVDESS 個別錄音員的正確率分布在 45%至 85%間，平均正確率則是 69.4%。

三、逼逼中文情緒語料庫結果

(一) 語料庫製作

1. 切割成兩秒與標註人聲有無

我們針對三筆原始音檔標註了人聲有無，數量如下表。

表九、小音檔標註結果統計表（單位：筆數）

	第一集	第二集	第三集	合計
類別 0	201	296	319	816(23.10%)
類別 1	30	32	70	132(3.74%)
類別 2	934	870	781	2585(73.17%)
合計	1165	1198	1170	3533

目前語料庫缺乏的主因是製作流程太麻煩，因此我們在製作逼逼中文情緒語料庫時有思考如何降低處理語料的時間。我們認為「標註人聲有無」部分對於機器學習來說並不困難，因此有望將這個步驟交給機器來分類。我們建立完成的機器學習模型正確率可以達到 89.6%，可以正確地代替人工標註一段小音檔是否有人聲。

2. 黏合成六秒與標註情緒

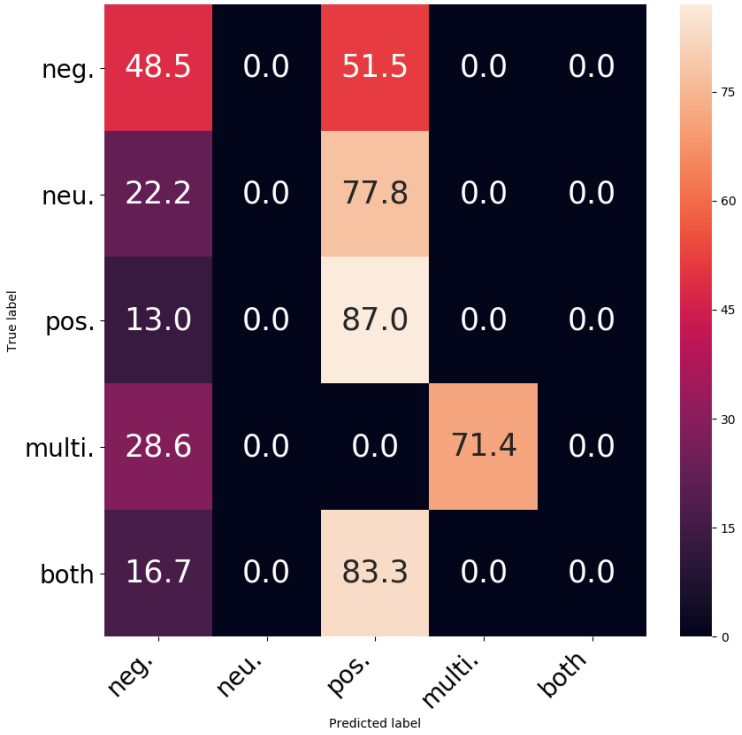
表十、六秒音檔標註結果統計表（單位：筆數）

	neg.	neu.	pos.	multi.	both	總計
第一集	321	20	425	2	23	791
第二集	300	18	358	24	20	720
總計	621	38	783	26	43	1511

上表即為人工標註情緒後的統計表。

(二) 以支持向量機分類情緒

多種情緒-固定比例-全體錄音員



圖三十三、多種情緒-固定比例-全體錄音員混淆矩陣（單位：%）

使用此組合（測試資料佔 20%）的訓練方法，正確率約為 65.33%。

單獨分析正面情緒，會發現正面情緒正確率頗高。而負面情緒則約有一半被分成正面情緒。

四、討論

(一) SVM 和 CNN 結果比較

以 CNN 作為分類器的測試，其結果與以 SVM 分類的結果比較略遜一籌，我們推測是因為資料量不夠，因此結構較複雜的 CNN 無法抓到資料的規律。

(二) 使用 SVM 作為分類器結果比較

1. 多種情緒、單一情緒比較：類別減少有助於提升正確率

單一情緒實驗在簡化標籤後，六種情緒正確率都明顯提升且不同情緒間差異不大。我們希望這可以應用在需要著重偵測某一情緒的情況。

2. 固定比例、留一驗證比較：訓練資料比例增加有助於提升正確率

由 SAVEE 資料庫的固定比例與留一驗證兩種模式比較，可發現留一驗證的結果較為準確，這符合訓練資料越多，訓練效果越好的普遍認知。

3. 全體錄音員、個別錄音員比較：錄音員對辨識效果影響高

從 SAVEE 資料庫的試驗結果，我們發現不同錄音員的錄音在 SVM 上較容易被辨識出的情緒都不太一樣：DC 的錄音資料中開心、厭惡情緒較容易判斷；JE 則是生氣情緒；JK 正確率較高的情緒是厭惡和恐懼；KL 則是開心。我們還發現不同錄音員的錄音資料對於 SVM 預測準確率會產生影響，並將自己做出的結果和原 SAVEE 資料庫的平台所做的結果作了比較。（請見圖二十九）

儘管個別錄音員測試時的訓練資料較少，但正確率都高於四位錄音員資料一起訓練的正確率。

由 RAVDESS 資料庫的結果也可發現相同狀況：單獨使用個別錄音員的資料作訓練時，大部分錄音員的個別訓練結果皆明顯高於一次使用所有錄音員的總正確率。（只有一位錄音員的正確率 46.7%低於總正確率 48.3%，其餘 23 位正確率皆高於 48.3%）

錄音員間表現情緒方法的差異使得 SVM 判斷情緒時較易被混淆，才會造成一次使用全部錄音員的錄音訓練時，雖然訓練資料較多，測試結果反倒低於只使用單獨錄音員的資料。

(三) 和 SAVEE 資料庫官網的模型正確率比較

SAVEE 資料庫製作官方也曾對 SAVEE 資料庫做情緒辨識，正確率為 61%，低於本研究多種情緒留一驗證的最高正確率 62.1%。

(四) 逼逼中文情緒語料庫

本資料庫為少數有足夠數量音檔，可供機器學習使用的中文語料庫。另外，我們也提出用程式輔助將長的音訊資源截取成適合辨識情緒的長度之方法，降低資料處理所需之時間成本。

肆、結論與應用

一、 結論

(一) 單一情緒時，正確率達 84%至 90%

本研究在單一情緒時可以做到 84%至 90%的準確度，可應用於只需單獨偵測某種情緒的情況。如生命線的輔導員在和諮詢者進行對話時，電腦可及時判斷諮詢者當下的情緒，有利於輔導員進行會話引導。在此情況中，情緒的偵測即可著重於判斷負面情緒。

(二) 資料庫數據的缺乏是語音情緒辨識的主要困境

由於資料庫的製作十分費時，因此可取得的已標註資料庫不多，資料量通常也不大。在情緒辨識的領域，語音的資料庫尤其稀少，這導致語音情緒的辨識正確率受限於資料量不足。

(三) 提出處理語音資料、製作語音資料庫的方法

本研究提出了，使用現存音訊資源，再利用機器學習挑選有人聲的部分並擷取成適當的長度。如此只有標註情緒的部分需要人工處理，可以大大縮短處理原始資料的時間，降低製作資料庫的成本。

(四) 證明中文語音情緒辨識可用英文情緒辨識流程達到分類

從我們的研究結果可以發現中文語音使用與英文相同的特徵及分類器可以達到不錯的分類結果，證實中文也可以藉由語音進行辨識，且這些特徵及分類器不只適用單一語言。

(五) 情緒表現不一，且判讀主觀，辨識情緒仍十分困難

由於每個人說話時不同情緒所表現的特徵因人而異，加上情緒的判讀十分主觀，目前辨識情緒仍是機器學習領域中一項困難的挑戰。尤其是在單使用語音的狀況下，少了面部表情變化，使得語音情緒辨識更加不容易。SAVEE 的官網也提到，與音頻相比，視覺特徵更清晰，表明面部表情對於情緒辨識仍十分重要。

二、 未來展望

(一) 整合語音情緒辨識和語意情緒辨識

如同研究動機中提及的，我們希望能用語音情緒辨識來輔助語意情緒辨識，以增進情緒辨識的準確率。未來希望能製作整合文字及語音的資料庫，同時利用兩種情緒表現方式進行情緒辨識的訓練。

(二) 建立個人化語料庫

在個別錄音員實驗中，我們發現錄音員間的差異是會影響情緒判斷的，個性的不同導致了每個人有不同的情緒表達方式。所以客製化的情緒辨識模型比統一的模型能有更高的正確率。然而目前仍只能透過手動調整的方式替每個人挑選適合的模型，未來若有自動化的調整模式，將能夠為更多人服務。

(三) 合併多種資料庫進行跨語言研究

我們想合併多種語言的資料庫，甚至是內容沒有明確語句的資料庫（有些語料庫的錄音內容是只藉由母音或狀聲詞表現情緒），來進行情緒辨識，以增加情緒辨識系統的通用性

伍、參考文獻

一、參考資料

[1] Ekman, P., Sorenson, E. R., & Friesen W. V. (1969). Pan-cultural elements in facial displays of emotion. *Science, 164*(3875), 86-88.

[2] Zhao, H., Ye, N., & Wang, R. (2018). A Survey on Automatic Emotion Recognition Using Audio Big Data and Deep Learning Architectures. *2018 4th IEEE International Conference on Big Data Security on Cloud*, 139- 142.

[3] Ooi, C. S., Seng, K. P., Ang, L.-M., & Chew, L. W. (2014). A new approach of audio emotion recognition. *Experts Systems with Applications, 41*, 5858–5869.

[4] Nam, J., Choi, K., Lee, J., Chou, S.-Y., & Yang, Y.-H. (2019). Deep learning for audio-based music classification and tagging: Teaching computers to distinguish Rock from Bach. *IEEE Signal Processing Magazine, 36*(1), 41–51.

[5] Jackson, P., & Haq, S. (2015, April 2). Surrey Audio-Visual Expressed Emotion (SAVEE) Database [Web blog]. Retrieved from http://kahlan.eps.surrey.ac.uk/savee/

[6] Livingstone, S. R., & Russo F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE, 13(5).

[7] 黃世琤、李明純、李麗雯、詹雅婷、蔡鑫廷（民 103）。台灣地區華人情緒與相關心理生理資料庫—基本情緒聲調。中華心理學刊，56 卷，4 期，437-452。

二、附錄：程式碼

```
#%%  import
import numpy as np
from glob import glob
import librosa as lr
import sklearn as sk
from sklearn import svm
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#%% read files
data_dir=r'C:\Users\Audio_SAVEE'
audio_files=glob(data_dir+'/*.wav')
n_file=len(audio_files)

#%% extract features
fea_all=np.zeros((n_file,45), dtype=float)
for i in range(0,n_file,1):
    audio, sfreq=lr.load(audio_files[i])
    #mfcc
    mf=lr.feature.mfcc(y=audio, sr= sfreq, n_mfcc=13)
    fea_all[i,0:13]=mf.mean(axis=1)
    fea_all[i,13:26]=lr.feature.delta(mf).mean(axis=1)
    fea_all[i,26:39]=lr.feature.delta(mf,order=2).mean(axis=1)
    #volume
    vol=lr.core.amplitude_to_db(audio,ref=0)
    fea_all[i,39]=vol.mean()
    fea_all[i,40]=np.median(vol)
    fea_all[i,41]=vol.std()
    #zcr
    zcr=lr.feature.zero_crossing_rate(audio,frame_length=2018, hop_length=1024, center=True)
    fea_all[i,42]=zcr.mean()
    fea_all[i,43]=np.median(zcr)
    fea_all[i,44]=zcr.std()

#%% check data
labels=emotion_labels
x=fea_all
y=raw_labels

#%% set test/train
index_test=np.array([0,4,9,11,18,20,26,27,33,39,40,43,45,54,56,59,61,64,70,76,82,84,87,88,91,96,98,99,108,114,
x_train = np.zeros( (n_file-len(index_test)+1 , 45) )
y_train = np.zeros( (n_file-len(index_test)+1) )
x_test  = np.zeros( (len(index_test)-1 , 45))
y_test  = np.zeros( (len(index_test)-1) )

test=train=0
for file in range(0, n_file, 1):
    if file!=index_test[test]:#train
        x_train[train, ::]=x[file, ::]
        y_train[train]=y[file]
        train+=1
    else:
        x_test[test, ::]=x[file, ::]
        y_test[test]=y[file]
        test+=1

#%% svm
svm_talk = svm.SVC(C=6500, kernel='poly', degree=3, gamma='scale')
svm_talk.fit (x_train, y_train)

y_pred = svm_savee.predict(x_train)
count=0
for i in range(len(y_train)):
    if y_pred[i]==y_train[i]:
        count+=1
acc_train=count/len(y_train)*100

y_pred = svm_savee.predict(x_test)

Code analysis
undefined name 'svm_savee'

acc_test=count/len(y_test)*100
```

```
#%% testing Cs
acc=np.zeros((20,2),dtype=float)
def try_c(min_c, max_c, gap, kernel, degree, gamma):
    global acc
    max_c=int(max_c)
    degree=int(degree)
    for j in range(min_c, max_c+gap, gap):
        print(j)
        svm_talk = svm.SVC(C=j,kernel=kernel,degree=degree,gamma=gamma)
        svm_talk.fit(x_train, y_train)
        y_pred_train=svm_talk.predict(x_train)
        y_pred_test=svm_talk.predict(x_test)
        count=0
        for i in range(len(y_test)):
            if y_pred_test[i]==y_test[i]:
                count+=1
        acc[(j-min_c)//gap,1]=count/len(y_test)*100
        count=0
        for i in range(len(y_train)):
            if y_pred_train[i]==y_train[i]:
                count+=1
        acc[(j-min_c)//gap,0]=count/len(y_train)*100

#%% confusion matrix
def print_confusion_matrix(confusion_matrix, class_names, figsize=(7,7), fontsize=16):
    df_cm = pd.DataFrame(confusion_matrix, index=class_names, columns=class_names)
    fig=plt.figure(figsize=figsize)
    try:
        heatmap = sns.heatmap(df_cm,  annot=True, annot_kws={"size": fontsize}, fmt=".1f")
    except ValueError:
        raise ValueError("Confusion matrix values must be integers.")
    heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, ha='right', fontsize=fontsize)
    heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=45, ha='right', fontsize=fontsize)
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.title('C=6500,kernel=poly,degree=3,gamma=scale\nCorrect Rate: '+str(acc_test))

#%% use confusion_matrix
c =sk.metrics.confusion_matrix(y_test, y_pred)
print_confusion_matrix(c, class_names=labels)
```

【評語】190018

本研究主題清楚且聚焦，且可用科學方法檢驗研究成果。實驗設計可以更清楚的呈現方法與資料的分析。建議對於不同相關研究成果進行比較，以進一步提出改進的方向與構思。