國立成功大學工程科學系

專題成果報告

Department of Engineering Science

National Cheng Kung University of Engineering

Practicum of Engineering Science

基於深度學習之聲紋辨識技術–實作智慧型影片字幕系統

學生：陳錦麟、梁哲嘉

指導教授：陳牧言 博士

中華民國 111 年 9 月

目 錄

第一章 緒論

第一節 研究動機

在深度學習這塊領域當中，其中一項熱門的應用便是將這項技術應用在聲紋辨識上。當我們在說話時，電腦能夠採集我們的聲音，將其音檔轉換成頻譜並進行各種分析，去學習辨識每個人發聲的特徵，進而完成一個只要聽一小段句子就能夠分辨出發聲者的模型。其精準性之高已足以應用於各種身分識別當中，例如：電信公司的繳款服務、智慧家電的聲控、甚至是手機的解鎖功能。

在這次的專題研究中，本專題希望能夠利用這項技術，實作出一個能夠準確分辨出影片中的說話者，且能分析並顯示其說話內容的辨識系統。其發想是源自於 YouTube 的「CC 字幕協作功能」，當 YouTube 的上傳影片者未提供影片字幕時，可以開放給觀看者進行字幕的協作，利用本次的專題作品，便能高效率地製作影片字幕，並附上說話者及對話時間軸。此外，亦能夠協助聽障者輕鬆地理解影片的內容。

第二節 研究目的

本專題將結合「聲紋辨識之深度學習訓練」、「利用餘弦相似性將模型應用在非訓練樣本上」、「引用訓練好之開源語音辨識模型」、「設計 GUI 介面」四項技術，製作出一個能夠在影片中，嵌入辨識之說話者名稱及說話內容的字幕系統。

並利用上述技術，研究以下主題：

- 訓練出能夠分析出發聲者之聲音特徵的學習模型：

  究聲音數據之處理過程、訓練模型結構和訓練過程。

- 使用訓練好的模型去辨認影片中的發聲者為何人：

  利用餘弦相似性做向量比對，避免模型應用範圍侷限在訓練樣本標籤內，而導致每新增一人就要再次收集樣本與訓練模型浪費時間。

- 利用語音辨識技術分析發聲者之說話內容：

  學習引用已開發健全的模型，結合自身研究應用於現實議題上。

- 將以上成果實作出智慧型影片字幕系統：

  實作應用程式，增加研究結果可讀性。包裝程式，針對使用者做考量。

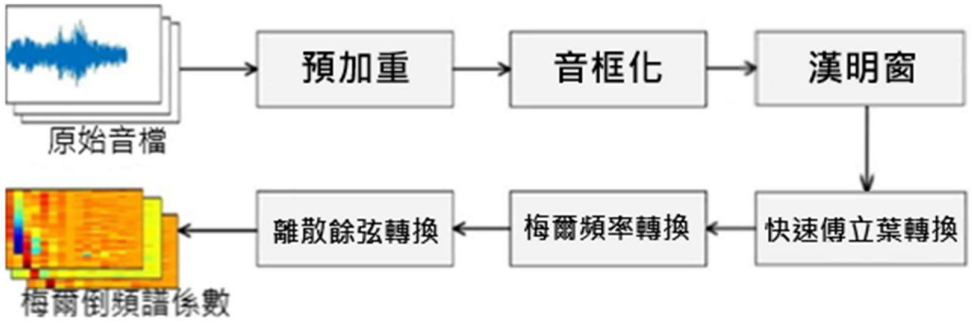
第二章 文獻探討

這次的作品主要會使用到三項技術，分別是 MFCCs、ResNet-50、餘弦相似性。

第一節 MFCCs(梅爾倒頻譜係數):

梅爾倒頻譜係數(Mel-Frequency Cepstral Coefficients, MFCCs)[1]是組成梅爾倒頻譜(Mel-Frequency Cepstrum, MFC)的相關係數。

倒頻譜(Cepstrum)是將頻譜(Spectrum)取對數後，做逆傅立葉轉換而得的訊號處理後數據，便於提取訊號中所關心之頻率組成。其中，梅爾倒頻譜在做逆傅立葉轉換之前，多加入一段梅爾頻率轉換，使得處理後數據更貼近人類聽覺系統，其詳細如下:



圖一、MFCCs 之處理步驟

(1) 預加重:使原始訊號通過一個高通濾波器，目的是為了去除發聲時聲帶與嘴唇的效應，以補償被壓抑的高頻訊號。

(2) 音框化:以 n 個連續取樣點為一個觀測單位(音框)，n 通常為 256 或 512，鄰近的音框有一段重疊，避免相鄰音框變化幅度過大。音框化的目的是保證後續快速傅立葉轉換時，能保留原始訊號的時間特徵。

(3) 漢明窗:將得到的音框乘上漢明窗，增加相鄰音框間的連續性。

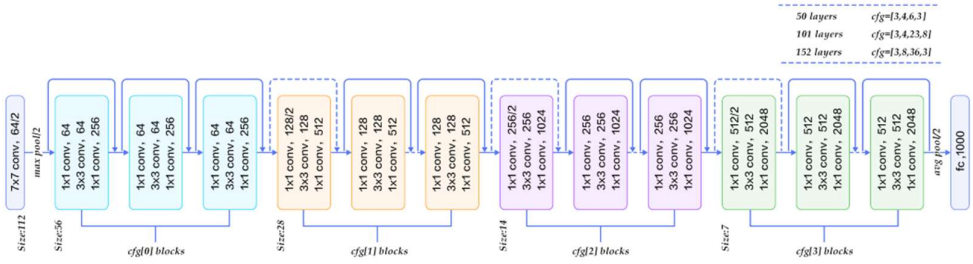
(4) 快速傅立葉轉換(FFT):將音框中的訊號由時域轉換成頻域，利用頻譜觀察其能量分佈。

(5) 三角帶通濾波器:將能量頻譜的能量乘上多個三角帶通濾波器，求得每一個濾波器輸出的對數能量，轉換成以人耳的感受度為基準的梅爾頻率。

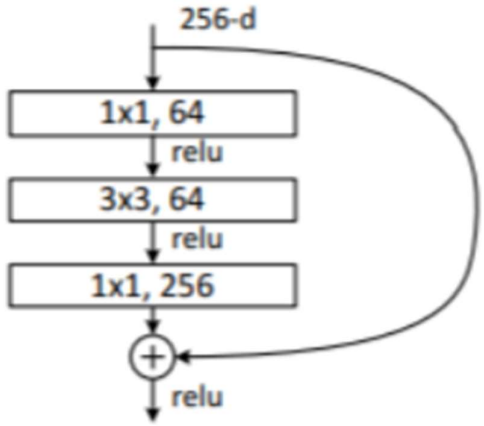
(6) DCT:將能量做離散餘弦轉換，使頻域圖轉換回類時域圖。

第二節 ResNet-50:

這次用來學習的模型是 ResNet-50[2]，這是在 2015 年所問世的訓練模型。其全名為 Deep Residual Neural Network。ResNet-50 的架構大致可以分為 6 層，第一層為 7*7 之卷積核，第二至五層的結構分別是由 3、4、6、3 個 block 組成，每個 block 都是由(1*1,3*3,1*1)的卷積核堆疊而成，其詳細結構如圖三。第六層作為輸出層則是由第六層作為輸出層則是由 pooling+全連接層+softmax 組合而成。



圖二、ResNet-50 之模型架構



圖三、卷積核的 block 之組成結構

與其他學習模型不同的是，它還利用恆等映射的方式，在較深層的網路中，

額外加入了淺層網路的輸出數值(如圖三)，再訓練層數加深的同時，也保留了原始資料的特性，避免資料的失真。這樣的方法有效地改善了加深神經網路的深度時，所獲得的準確度不增反減的問題，因而被廣泛應用於深度學習當中。

第三節 餘弦相似性

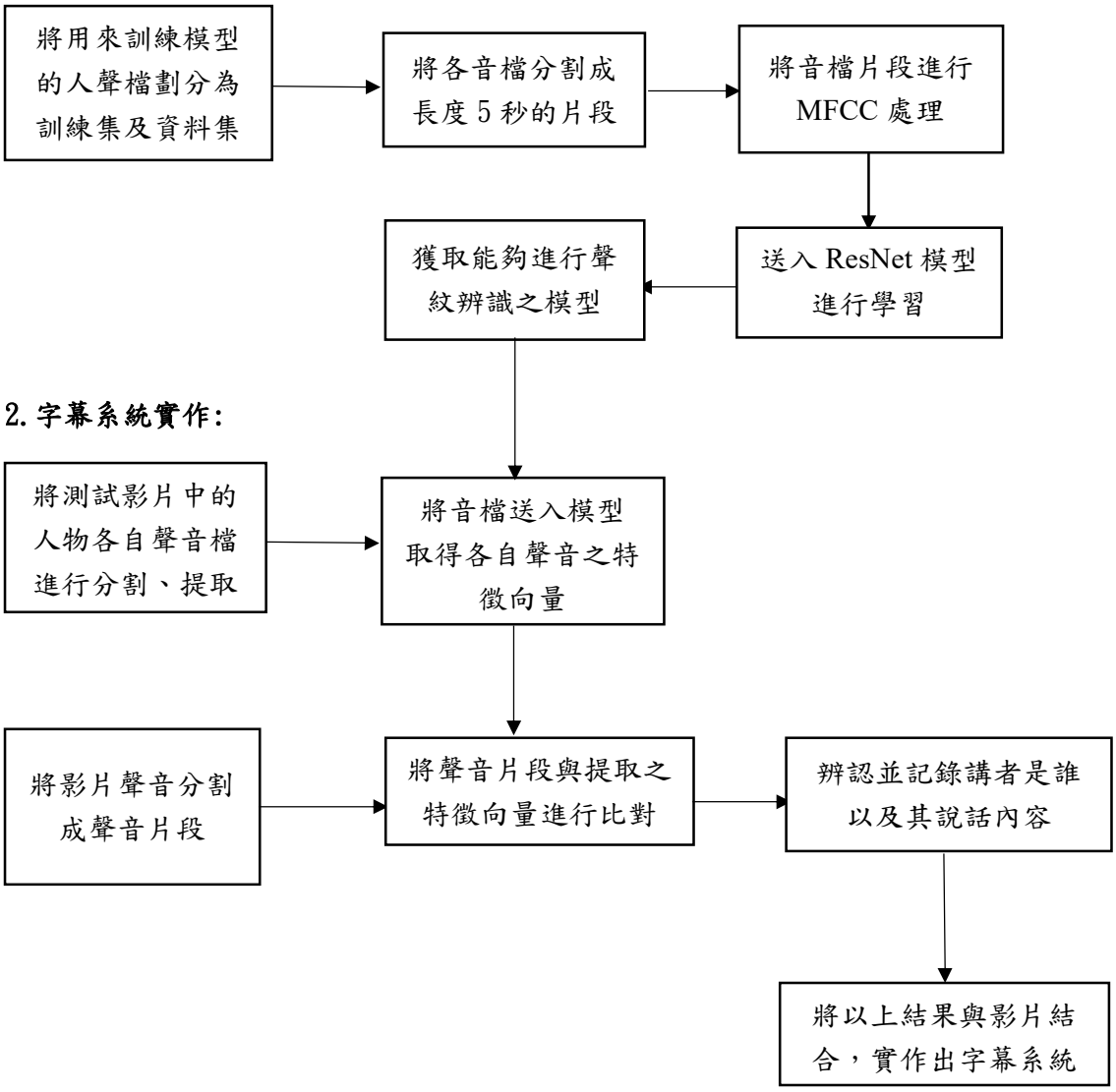
本專題使用餘弦相似性來比較目標樣本與發聲者之特徵向量之間的相似性。其方法是透過測量兩個向量之間的餘弦值來度量他們之間的相似性，當餘弦值為 -1 時，就代表兩者之間相差最大，而餘弦值為 1 時，則代表他們之間的相似度最大。詳細公式如下:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

第三章 研究方法

下圖為本次專題之實作流程:

1. 模型訓練:



圖四、本次專題之研究流程

1. 模型訓練:

資料取自 Kaggle 數據平台的 Speaker Recognition Audio Dataset[3]提供的 51 人無雜音環境下的錄音數據，每個人有 20-120 不等的一分鐘錄音檔。將每個人

的錄音檔，以 9:1 的比例劃分成訓練集以及測試集。

將音檔無聲片段去除後，以每 5 秒為一片段，經由 librosa 函式庫的 librosa.feature.mfcc 函式處理成梅爾倒頻譜之後，放入模型內進行學習[4]，並採用即時停損來避免過擬合。

訓練模型的部分，這次的作品主要是利用 Python 程式語言在 Google Colab 中編寫程式。模型使用 Keras 提供的 Resnet 未訓練模型依序加上 Dropout 層、Global Max Pooling 層與全連接層製成。目的是訓練模型辨識資料中 51 個人的聲音，並去掉最後的全連接層，得到生成「可以辨識他人聲音的特徵向量」的生成模型。

2. 字幕系統實作:

模型訓練完畢之後，進行影片測試的部分。本專題從 YouTube 影音平台下載 So Very Wrong About Games 頻道的《Stroganov Review by So Very Wrong About Game》[5]，2 名成年男子 Mark 和 Michael 英語對話影片。

首先從影片中擷取兩人聲音，削除噪音之後，送入訓練好的生成模型內，得到各自聲音之特徵向量，並標上標籤。此特徵向量為辨識影片各時段講者的預測憑據，目標向量。

之後將影片之聲音切割成聲音片段，分別輸入至講者辨識和語音辨識函式中：講者辨識函式將聲音片段處理成梅爾倒頻譜後，送入生成模型生成特徵向量，並使用餘弦相似性分析，找出與特徵向量最相似之目標向量，預測聲音片段之講者為該目標向量的標籤人物。語音辨識函式引用 Speech Recognition 函式庫中的 Google 語音辨識模型，得到聲音片段之語音內容。

將預測的標籤人物與語音內容儲存到字幕檔中。製作一個影片控制介面將影片之畫面與生成的字幕檔進行整合，製作出智慧型的影片字幕系統。觀察電腦所分析出的內容是否與實際影片內容一致。

第四章 實驗設計和績效評估

第一節 實驗環境

本專題之實驗環境主要是在 Google Colab 之開發環境下進行，用來訓練模型之 GPU 為 Tesla T4，虛擬機之 RAM 為 12.68GB，磁碟空間為 78.19GB。詳細硬體規格如表一。

硬體名稱	硬體規格
GPU	Tesla T4
RAM	12.68GB
Disk	78.19GB

表一、本專題使用之硬體規格

程式撰寫的部分，本專題所使用之語言為 Python，訓練用的模型所採用之框架為 Keras 提供的 ResNet-50，相關函式庫為 TensorFlow。聲紋及語音內容辨識的部份分別使用到了 Librosa 及 SpeechRecognition，UI 介面則是使用 Pyqt5。詳細之套件版本如表二。

套件名稱	套件版本
Python	3.7.13
TensorFlow	2.8.2
Librosa	0.8.1
Pyqt5	5.15.7
SpeechRecognition	3.8.1

表二、本專題使用之套件版本

而這次的研究所用到之技術的函式來源部份，用來獲取梅爾倒頻譜係數的函式為 Librosa，實作 ResNet 模型所使用之函式為 TensorFlow.keras，並採用 SpeechRecognition 函式進行語音辨識。詳細來源如表三。

技術名稱	函式來源
MFCCs	librosa.feature.mfcc
ResNet-50	tensorflow.keras.application.ResNet50V2
語音辨識	Speech_recognition.Recognizer().recognition_google

表三、本專題使用之技術的函式來源

第二節 資料集說明

本次專題所用來訓練聲紋辨識模型的資料集來源是取自 Kaggle 平台的 Speaker Recognition Audio Dataset [6]。此資料集包含了 50 名講者無雜音錄音數據，每個人有 20-120 個不等的一分鐘錄音檔，所使用之語言均為英文，且腔調多有不同。資料集分割依據試誤法，以每人 9:1 的比例劃分成訓練集及資料集。

而字幕系統之資料集是取自 YouTube 網站之影片《Stroganov Review by So Very Wrong About Games》[6]，首先提取兩位講者: Mark Bigney 及 Michael Walker 的聲音之特徵向量。並使用兩人之對話影片作為展示本專題之實驗成果之影片。

第三節 參數設定

本專題之參數設定依據試誤法和文獻[4]。

librosa.feature.mfcc 函數參數設定如下表：

sr (sample rate)	22050
n_mels	128
hop_length	256

表四、梅爾倒頻譜函式參數設定

ResNet 模型之參數設定如下表:

Layer 名稱	參數設定
ResNet50V2	include_top = False, weight=None, input_shape = (128, None, 1)
Activity Regularization	L2 = 0.5
Dropout	0.5
Dense	units = 50, activation = tf.nn.softmax
Optimizer	Adam (Rate:1e-4)

表五、模型參數設定

模型結構如下表：

```
Layer (type)                 Output Shape              Param #
=================================================================
resnet50v2 (Functional)      (None, 4, None, 2048)     23558528

activity_regularization (Ac  (None, 4, None, 2048)     0
tivityRegularization)

dropout (Dropout)            (None, 4, None, 2048)     0

global_max_pooling (GlobalM  (None, 2048)              0
axPooling2D)

dense (Dense)                (None, 50)                102450

=================================================================
Total params: 23,660,978
Trainable params: 23,615,538
Non-trainable params: 45,440
_________________________________________________________________
```

表六、模型結構

模型訓練參數入下表：

batch size	32
loss	sparse_categorical_crossentropy

表七、模型訓練過程參數

模型訓練時採用即時停損技術，當損失值連續上升超過 100 次且最低損失值小於 0.00005 時，或損失值連續上升 300 次且最低損失值小於 0.5 時，訓練提前終止。

影片方面，採用高於 40 分貝連續片段為一個聲音片段，如果聲音片段大於 4 秒，則分割成每 4 秒一段。減少 2 人對話無縫銜接的可能性。

第四節　實驗結果

將 50 位人物的聲音進行學習時，此次實驗設定每經過 20 個 epoch 就顯示一次學習的成果，包含 loss 及 accuracy，同時設定每經過 100 個 epoch 就將測試集內的資料放進模型內進行測試，用以觀察學習成效。下圖為生成之 log 檔:

```
Batch 0, Loss 3.083894, Accuracy 0.125000
Batch 20, Loss 2.707834, Accuracy 0.218750
Batch 40, Loss 3.874828, Accuracy 0.125000
Batch 60, Loss 3.616269, Accuracy 0.062500
Batch 80, Loss 3.310273, Accuracy 0.125000
Batch 100, Loss 2.880338, Accuracy 0.125000
================================================
Test, Loss 3.755861, Accuracy 0.148438
================================================
Batch 120, Loss 3.320084, Accuracy 0.218750
Batch 140, Loss 2.967261, Accuracy 0.281250
Batch 160, Loss 2.439378, Accuracy 0.375000
Batch 180, Loss 2.620418, Accuracy 0.218750
Batch 200, Loss 3.470358, Accuracy 0.125000
================================================
Test, Loss 2.845028, Accuracy 0.203125
================================================
Batch 220, Loss 1.981255, Accuracy 0.468750
Batch 240, Loss 1.949966, Accuracy 0.312500
Batch 260, Loss 1.136397, Accuracy 0.625000
Batch 280, Loss 2.062478, Accuracy 0.343750
Batch 300, Loss 0.892787, Accuracy 0.750000
================================================
Test, Loss 1.334791, Accuracy 0.591146
================================================
Batch 320, Loss 0.949709, Accuracy 0.593750
Batch 340, Loss 1.195066, Accuracy 0.593750
Batch 360, Loss 1.642031, Accuracy 0.500000
Batch 380, Loss 1.101279, Accuracy 0.656250
Batch 400, Loss 1.020908, Accuracy 0.625000
================================================
Test, Loss 2.117514, Accuracy 0.473958
================================================
Batch 420, Loss 0.631980, Accuracy 0.812500
Batch 440, Loss 0.584844, Accuracy 0.875000
Batch 460, Loss 0.602471, Accuracy 0.781250
Batch 480, Loss 0.341835, Accuracy 0.843750
Batch 500, Loss 0.495987, Accuracy 0.750000
================================================
Test, Loss 1.008033, Accuracy 0.721354
------------------------------------------------
```

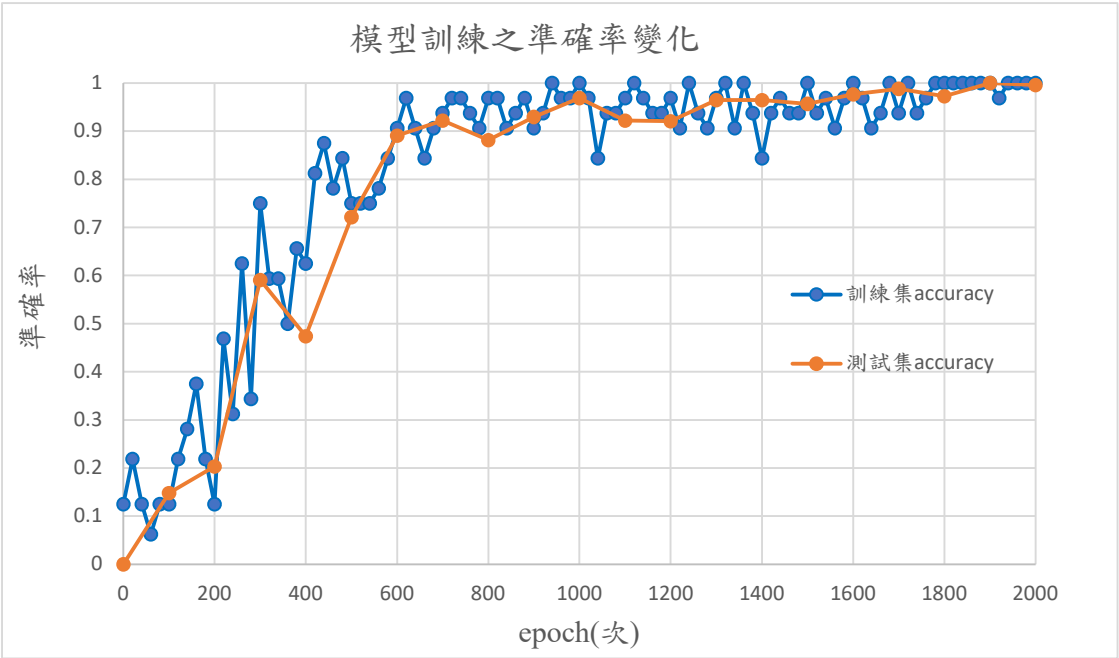
圖五、剛開始學習時之成效

```
Batch 1420, Loss 0.092779, Accuracy 0.937500
Batch 1440, Loss 0.037117, Accuracy 0.968750
Batch 1460, Loss 0.168175, Accuracy 0.937500
Batch 1480, Loss 0.148017, Accuracy 0.937500
Batch 1500, Loss 0.032422, Accuracy 1.000000
================================================
Test, Loss 0.124093, Accuracy 0.957031
================================================
Batch 1520, Loss 0.157505, Accuracy 0.937500
Batch 1540, Loss 0.143895, Accuracy 0.968750
Batch 1560, Loss 0.204123, Accuracy 0.906250
Batch 1580, Loss 0.068202, Accuracy 0.968750
Batch 1600, Loss 0.017036, Accuracy 1.000000
================================================
Test, Loss 0.081529, Accuracy 0.976562
================================================
Batch 1620, Loss 0.086019, Accuracy 0.968750
Batch 1640, Loss 0.655353, Accuracy 0.906250
Batch 1660, Loss 0.095313, Accuracy 0.937500
Batch 1680, Loss 0.011634, Accuracy 1.000000
Batch 1700, Loss 0.210465, Accuracy 0.937500
================================================
Test, Loss 0.040055, Accuracy 0.988281
================================================
Batch 1720, Loss 0.015951, Accuracy 1.000000
Batch 1740, Loss 0.131448, Accuracy 0.937500
Batch 1760, Loss 0.164914, Accuracy 0.968750
Batch 1780, Loss 0.007069, Accuracy 1.000000
Batch 1800, Loss 0.003253, Accuracy 1.000000
================================================
Test, Loss 0.049777, Accuracy 0.972656
================================================
Batch 1820, Loss 0.015313, Accuracy 1.000000
Batch 1840, Loss 0.004200, Accuracy 1.000000
Batch 1860, Loss 0.009637, Accuracy 1.000000
Batch 1880, Loss 0.011852, Accuracy 1.000000
Batch 1900, Loss 0.003268, Accuracy 1.000000
================================================
Test, Loss 0.009678, Accuracy 1.000000
================================================
```

圖六、測試正確率達 100%時之成效

從圖四中可以看到，經過 500 個 epoch 之後，測試之正確率從一開始的 14.8%快速上升到 72.1%，由此可知，電腦確實能夠從每個人的聲音當中，獲得其各自之特徵向量。

而在圖五中，可以觀察到經過 1900 個 epoch 後，不論是訓練集或測試集，其辨識精準度竟然達到了 100%，這也就表示，此時電腦已能夠非常準確地辨識發聲人是誰了。



圖七、模型訓練之準確率變化

接下來這次的實驗使用 YouTube 影片《Stroganov Review by So Very Wrong About Games》[6]作為測試影片。首先將影片中的 2 人: Mark Bigney 以及 Michael Walker 各自的影片經過人聲提取後，放入模型內獲取其各自聲音之特徵向量。再利用先前所訓練的模型，將測試影片放入程式內，每隔 4 秒程式便會辨識當前的說話者是誰，同時利用 Google 的開源模型 API 辨識其對話內容，最後生成字幕檔。下圖為字幕檔之生成結果:

1. 時間軸 2. 辨識出之說話者名稱 3. 辨識出之說話內容

```
[00:03:29] Mark : almost never make this recommendation when learning this gave the first
[00:03:33] Mark : spread the board out
[00:03:35] Mark : while you're reading the rule book because otherwise you're not going to realize
[00:03:38] Mark : little you have to remember going into
[00:03:40] Michael : two types of actions basic action
[00:03:42] Michael : Advanced auction
[00:03:45] Michael : Ford is covered with either
[00:03:46] Michael : silver rings are gold rings
[00:03:48] Michael : those two different transactions we know
[00:03:51] Michael : what the actions are and what type of action it is
[00:03:54] Mark : there's a lot of different ways
[00:03:58] Mark : there are these favor of the Tsar cards motive which give you special
[00:04:01] Mark : our summer scoring
[00:04:03] Mark : there are a whole bunch of one-time effects to give you points
[00:04:06] Mark : points are going to come from buying these landscape tiles
[00:04:09] Mark : in turn is influenced by the economy of hunting them so there's
[00:04:13] Mark : play there especially in terms of not wanting to give you
[00:04:16] Michael : phone is good deals
[00:04:17] Mark : flash snapping up the deals
[00:04:19] Mark : left for you
[00:04:20] Mark : honestly the trade-offs involved are
[00:04:22] Mark : some of the standard ones you're going to find another your games but I think the
[00:04:25] Mark : working out does it
```

圖八、影片之字幕辨識結果

從影片分析完畢之後所生成之 log 檔中，可以觀察到電腦已經學會分辨清楚每一段話究竟是由 Mark 還是 Michael 所說的，然而由於對話語速、語調、同時說話等等的因素干擾，造成辨識出的對話內容準確率有些許下降的趨勢。

最後使用 PyQt5 將測試影片之影像與得到的字幕檔結合，實作出能夠自動標上說話者名稱及說話內容的字幕軟體，其成果如下圖：

圖九、智慧型字幕系統展示(1)

圖十、智慧型字幕系統展示(2)

使用者將影片檔及先前所生成之字幕檔放入本專題所製作之軟體後，便能開始觀賞影片，使用者除了能夠觀看影片及其字幕外，也能同時看見電腦辨識出的說話者名稱。

第五章 結論

第一節 研究貢獻

- 針對成年男子英語聲音辨識之訓練模型，本專題之研究成果精確度高，對聲音數據之處理、模型結構以及訓練過程可以給予參考。
- 避免了訓練資料限制，利用同為說英文的成年男子，卻非訓練資料的 2 人影片進行講者辨識，辨識度 7 到 8 成，驗證了餘弦相似性定理在辨識模型中應用的可行性。
- 結合 Google 語音辨識，得到對話中重要的兩大訊息：說話者和說話內容，投入到影片中進行實際應用。
- 實作字幕與影片同步系統，未來社會大眾若在網路上看到沒有附上字幕的影片，或者無法明確得知說話者是誰的時候，便可以使用本專題所製作之智慧型字幕系統，不僅能夠得到外加的字幕，也能夠得知說話者之資訊。抑或是當電腦沒有喇叭或無法播放音訊時，無須依靠聲音，這個軟體也能夠讓大家輕鬆地觀賞影片。
- 除了觀看者以外，影片的上傳者也能利用本專題所製作之軟體，快速地生成出字幕檔，只要將影片稍加修改，就可以簡易後製出影片，無須再耗費大量心力於字幕的製作上，省時又省力。
- 對於聽力障礙者也是得力的助手，即使無法聽清楚影片聲音，在本專題之軟體的輔助之下，亦能夠輕易地理解影片內容。

第二節 研究限制

這次的專題研究當中，對於辨識發聲者的部分算是相當成功，但是在使用雙人對話影片進行測試時，由於對話情境有許多因素干擾，以及用來訓練的資料集人數不足，使得學習出來之模型仍無法百分百準確辨識出說話的內容等等之原因，

因此認為這部分還有能夠改善的空間，像是尋找更有利於訓練之資料集、最佳化提取人聲的演算法、研究能夠更精準辨識對話內容的演算法等等。希望未來能夠實現實時翻譯影片內容的程式，為社群協作盡一份心力。

參考文獻

[1] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi (March 2010), Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques, *Journal of Computing*,2(3), 136-143

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (Jul 2016), Identity Mappings in Deep Residual Networks, Microsoft Research

[3]Vibhor Jain, Speaker Recognition Audio Dataset (2019), 檢自 https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset?fbclid=IwAR1totcNgBHAMMOku-LFQPlQUF2OaaEKCAhuFGQi96-XjbWHp1vXJ4n3vbQ (2019)

[4] 使用 Tensorflow 实现声纹识别 (2020), 檢自 https://blog.csdn.net/qq_33200967/article/details/105915781 (May.04, 2020)

[5]《Stroganov Review by So Very Wrong About Games》(2022), 影片來源: https://www.youtube.com/watch?v=gp8tlRyekmM&ab_channel=SoVeryWrongAboutGames