中華民國第 63 屆中小學科學展覽會

作品說明書

高中組　電腦與資訊學科

團隊合作獎

052507

影片情境化字幕實現探討

學校名稱：國立花蓮高級中學

作者：	指導老師：
高二 陳恩泓 高二 關以諾 高二 顧懷允	趙義雄

關鍵詞：聲音辨識、人臉辨識、影片自動化處理

摘要

本研究旨在改善聽障人士無法完整接收影音類型資訊的狀況，探討各種影片處理技術，尋找、嘗試並比較各種方法，整合出最適合的系統自動替影片嵌入情境化字幕——用視覺的方式呈現影片聽覺訊息，讓聽障人士便於理解各種類型的影片內容與資訊。

為此，我們呈現的情境化字幕有主要幾個特點：

1、將聲音對話轉為字幕標記在說話者旁，透過畫面中語句位置就可以了解跟語者的對應關係。

2、畫面中字幕會以漸漸上飄消失的泡泡字幕來呈現，使觀影者有充足時間閱讀字幕理解內容。

3、將環境音效如電話聲、雷聲與貓叫聲等各種能傳達資訊的聽覺訊息標示在畫面中。

藉由這些處理使畫面呈現更豐富的影片資訊，最終達到改善聽障人士資訊接收權益不平等的目標。

壹、 前言

一、研究動機

平時我們接觸的網路媒體多數都是影音的形式，如直播、新聞、娛樂電影、政治節目與政策說明會等等，然而並非所有內容都會加上字幕，聽障人士也就無法完整的理解這些資訊，因此為了保障他們的權益，大部分的政見說明會都會有手語協助他們了解內容，美國電影院更是推出了隱藏式字幕來讓所有人都能享受影音樂趣，但是以上這些服務成本較高且難以普及。

而聯合國於 2005 年宣布了「2030 永續發展目標」(Sustainable Development Goals, SDGs)，當中十七項核心目標中，第十項目標「消除不平等」強調要保障身心障礙人士的權益。因此，我們就想實現一套系統，可以自動化將影片聽覺訊息視覺化呈現的情境化字幕，保障這類族群的媒體接收、識讀的權利；同時研究當中我們也針對商業化的影片字幕運作效果做一些比較，如 Netflix 影片的描述性字幕，探討我們系統可以改進的方向。

二、研究目的

(一)、透過語音辨識將原影片音檔自動轉換為文字

(二)、透過語者分群辨識來判斷影片中的說話者

(三)、結合人臉分群辨識來達到將字幕標示於說話者旁

(四)、能將環境音效標示於畫面中

(五)、將上述功能整合成自動化流程，為一部影片嵌入情境化字幕

貳、 研究設備與器材

一、硬體

(一)、筆記型電腦 ASUS X509 處理器 Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz *3

二、軟體

(一)、Anaconda 環境下的 Python 編譯器 Spyder (Python 3.8)

(二)、VScode-Python(Python 3.8)

(三)、Google Colab(Python 3.10)

(四)、重要 Python 套件

1、OpenCV

2、scikit-image

3、MediaPipe Face Mesh

4、Dlib

5、Whisper AI (Whisper-timestamped)

6、spleeter

7、MediaPipe audio classification

8、SpeechRecognition

參、 研究過程或方法

一、文獻探討

(一)、聽障人士的需求

現今的電視節目如果沒有字幕，將會嚴重影響聽障人士接收資訊的權利，例如晚間時段的新聞邀請政治家來討論、批判政府政策、報導最新型疫情的現狀或是政見公聽會，若沒有字幕或提供手語指示，聽障人士就無法接收完整資訊。

《殘疾人士權利國際公約》第 9 條關於無障礙的條文提到，殘疾人士有權生活於一個無障礙的社會，享有無障礙的資訊、通信和其他服務（包括電子服務和緊急服務），因此政府有責任確保各管業機構在其向公眾開放的建築物內，為殘疾人士提供無障礙的環境及設施，如引路徑、凸字標誌和易讀易懂標誌、手語翻譯、緊急訊息電子通告板等。

針對影片廣告等影片字幕的設計，還需要將不同說話者的字幕分開標示，以及增加畫面情境的提示字幕，才能讓聽障人士更好理解影片傳達的資訊。以美國電影院的「隱藏式字幕」為例，會將字幕標示於說話者附近，也會把背景聲音作為提示字卡顯示於畫面中。

以上許多資料都顯示了我們現在社會中還存在的不平等問題，而聽障人士也需要以下資源：

1、影音媒體加入字幕保障聽障人士獲得資訊的權力

2、用顏色或位置將不同說話者跟對應的語句標示出來

3、加入背景音效提示字卡協助理解畫面

此外，我們也參考了如「聽覺障礙者使用同步聽打服務經驗之探究」[1]這篇論文以及 TikTok 與 Netflix 等影音平台推出專為聽障人士提供字幕的政策，目的都是在為這個族群提供更完善的服務，保障他們資訊接收的權利，但是以上內容不管是同步聽打服務還是影音平台的字幕皆需要人力來協助這些功能，時間與勞力也勢必是提供服務者要負擔的成本。

因此，本研究著重於設計一套自動化系統，將各種類型的影音資訊加上字幕以及情境提示字，讓情境化字幕實現過程以電腦可自動化的程式來取代，方便影片發布者增添字幕的同時也保障聽障人士獲得資訊的權利。

(二)、MediaPipe 開源專案[2]

為了在影片中的說話者旁標示出對應的字幕，我們需要標示出每一幀圖片中說話者的座標位置，於是我們選擇使用 MediaPipe 框架，會選擇使用 MediaPipe 是因為它有內建人臉辨識的模組，可以讓我們更方便的提取人臉中的特徵點進行進一步的分析。

1、人臉網格 (Face Mesh)

MediaPipe 的 Face Mesh 可以將人臉轉換為幾何網格模型，經由機器學習判斷人臉的表面和深度，再透過 468 個臉部標記畫出 3D 的人臉網格。

2、人臉網格座標

在 MediaPipe 專案裡面，可以找到人臉網格標註的圖片，如圖 1 所示：

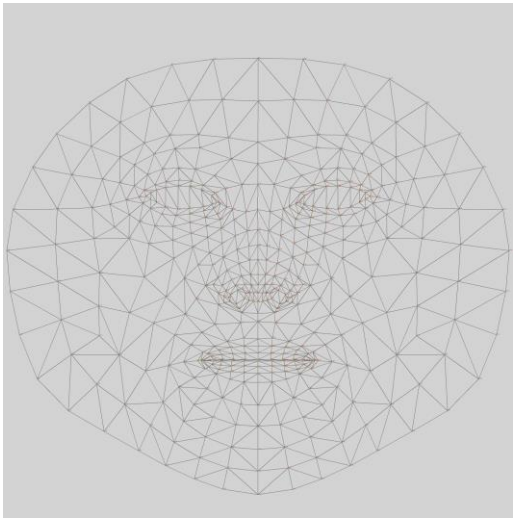


圖1、人臉網格標註圖片

圖片上每個標註點都有各自的編號，這個編號為每個標註點的名稱，我們可以透過在程式中輸入編號，獲取圖片上對應的位置座標。

(三)、K-Means 演算法(K-Means Clustering)[3]

K-Means 為非監督式學習的演算法，將一群資料分成 k 群 (cluster)，原理上是透過計算資料間的距離來作為分群的依據，較相近的資料會成形成一群並透過加權計算或簡單平均可以找出中心點，透過多次反覆計算與更新各群中心點後，就能找出代表該群的中心點，之後便可以透過與中心點的距離來判定測試資料屬於哪一分群，或可進一步被用來資料壓縮，代表特定類別資料，以達到降低雜訊或填空值等議題。此為分

割式分群法(partitional clustering)中的一種，藉由反覆運算，逐次降低誤差目標值，直到目標函式值不再變化或更低，就達到分群的最後結果。

分割式分群法目的是希望盡量減少每個分群中，每一資料點與群中心的距離平方差 (square error)，假設一組包含 c 個群聚的資料，其中第 k個群聚可用集合 G_k 表示，而 G_k 包含n_k 筆資料$\{x_1, x_2, x_3, \dots, x_{nk}\}$，此群聚中心為 y_k，則該群聚的平方誤差e_k為

$$e_k = \sum_i |x_i - y_i|^2$$

其中x_i 是屬於第k 群的資料點。

而這 c 個群聚的總合平方誤差E 便是每個群聚的平方誤差總合，可稱為分群的誤差函數 (error function) 或失真度 (distortion)。

$$E = \sum_{k=1}^{c} e_k$$

故分群方法就變成一個最佳化問題，也就是說要如何選取 c 個群聚及其相關群中心，可促使 E的值最小。

若用目標函式來說明，則假設給定一組n點資料 $X = \{x_1, x_2, x_3, \dots, x_n\}$，每一資料點有$d$維，K-Means 分群為找到一組 m 代表點 $Y = \{y_1, y_2, y_3, \dots, y_m\}$，每個點亦是 d 維，促使下方目標函數越小越好：

$$J(X, Y, U) = \sum_{i=1}^{n} |x_i - y_k|^2$$

K-Means 在測試資料具有代表性或資料趨近於常態分布時有相當好的結果，但當訓練資料過少或不具代表性時，K-Means 的分群結果相當的差，且會因訓練資料問題造成 k 值判定易出現過適應問題(overfitting)，通常 K-Means 的 k 值定義在專業知識的判斷下較容易有好的分群結果；但對於未知的資料時，則可以透過 k 的循序遞增或遞減等，查看資料間的分布差異，便可以了解 k 值為何可能為最佳，也就是接下來要提到的輪廓係數法(Silhouette Coefficient)。

輪廓係數法的概念是「找出相同群凝聚度越小、不同群分離度越高」的值，也就是滿足 Cluster 一開始的目標。其算法如下：

$$S = \frac{b - a}{max(a, b)}$$

其中，凝聚度（a）是指與相同群內的其他點的平均距離；分離度（b）是指與不同群的其他點的平均距離。S是指以一個點作為計算的值，輪廓係數法則是將所有的點都計算S後再總和。S值越大，表示效果越好，適合作為 k。

(四)、Speech Bubbles

在「SpeechBubbles: Enhancing Captioning Experiences for Deaf and Hard-of-Hearing People in Group Conversations」[4]這篇論文的研究目的中有提到要解決聽障人士進行團體談話時所遭遇到的困難，包括多個說話者同時發聲以及說話者不在視線範圍內等。這和我們「消除聽障人士不平等」的目標不謀而合。

文中提到相較於傳統字幕，聽障人士較喜歡「泡泡字幕」的呈現方式。泡泡字幕指的是隨時間流逝而上飄的字幕，這不只能幫助聽障人士釐清每句話的說話者為何，逐漸上飄的字幕在畫面中也會有更長的停留時間，讓聽障人士有更多的時間閱讀及理解字幕，因此我們決定在研究中使用泡泡字幕來取代傳統字幕。

(五)、語者辨識

1、實現語者辨識

標記字幕時需要有語句對應的語者資料，但是在一部新的影片輸入時程式並不知道影片中的語者數量與特徵資訊，因此我們採用語者分群，將影片中所有語者的聲音統一提取特徵後分類為數個語者。此種方法的優點是可以不用仰賴先前標籤好的資訊，對於一群未知的音訊便可以起到分類的效果。

(1)、特徵提取(Feature Extraction)

選擇從語音訊號中提取哪些特徵將會是語音分群中最重要的部分，會直接關乎到結果優劣。 目前有一些流行的特徵是：MFCC、LPC、過零率等。這份研究中，我們主要使用 MFCC 和 LPC 兩種特徵來比較與實驗。

(2)、MFCC(Mel-Frequency Cepstral Coefficients)

人類的聽力本質上不是隨頻率線性變化，而是成對數關係，這代表我們的耳朵是一種過濾器，而 MFCC 就是基於已知的人耳臨界帶寬隨頻率的變化轉換而來。這個濾波器在低頻呈線性分佈，在高頻呈對數分佈，常用於各種音訊處理。

(3)、LPC(Linear Prediction Coefficients)

LPC 是一種線性預測係數，它運用了語音自回歸模型以及對每個語音幀做標準化處理，每幀的結果都來自於先前時段的線性預測結果，而這個預測值是一個矩陣，透過一連串的線性轉換得到這個特徵。

要了解 LPC，我們必須首先了解語音的自回歸模型。 語音可以建模為 p 階 AR 過程，其中每個樣本由以下公式給出：

$$x(n) = -\sum_{k=1}^{p} \alpha_k x(n-k) + u(n)$$

上面公式中第 n 個時刻的每個樣本都取決於 p 個先前樣本，並添加了高斯噪音 $u(x)$。該模型假設語音信號是由管末端的蜂鳴器（濁音）產生的。LPC 係數由 α 給出。為了估計係數，我們使用 Yule-Walker 方程來推導線性相關係數。它使用自相關函數 $R(l)$。相臨時間段(lag) l 處的自相關性由下式給出：

$$R(l) = \sum_{n=1}^{N} x(n)\, x(n-l)$$

計算出來 Yule-Walker 方程的最終形式為：

$$\sum_{k=1}^{p} \alpha_k R(l-k) = -R(l)$$

α 解由下式給出：

$$\alpha = -R^{-1} r$$

在這種情況下，我們已對估計的 LPC 係數進行標準化處理，使其位於 [-1,1] 之間，可以提供更準確的結果。

2、特徵壓縮與配對(Feature Matching)

原始提取出來的語音訊號是多維且巨大的數值，為了有較好的分群與辨識效率，我們在處理資料前須先將資料做一定的壓縮，而本研究使用的特徵壓縮演算法是向量量化(Vector quantization, VQ)。VQ 是將向量從大向量空間映射到該空間中有限數量的區域的過程。每個區域稱為一個簇，可以由其特徵碼字(code)的中心表示。所有碼字的集合稱為碼本(codebook)。在訓練期間，LBG 演算法通過最小化簇中每個向量與碼字之間的失真來為每個簇選擇一個碼字當作代表。碼字的集合形成每個說話者唯一的特定碼本。

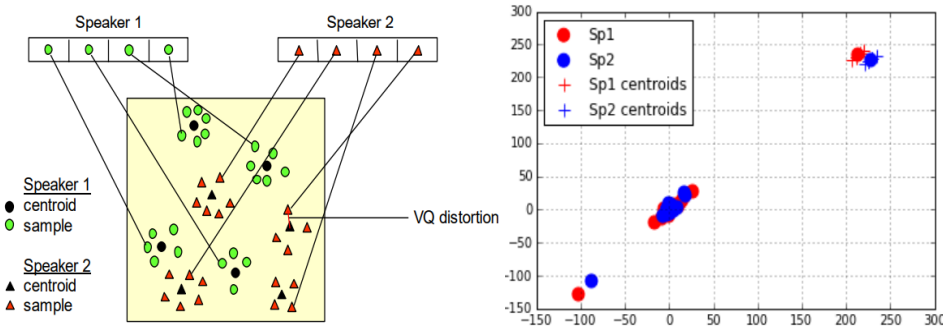
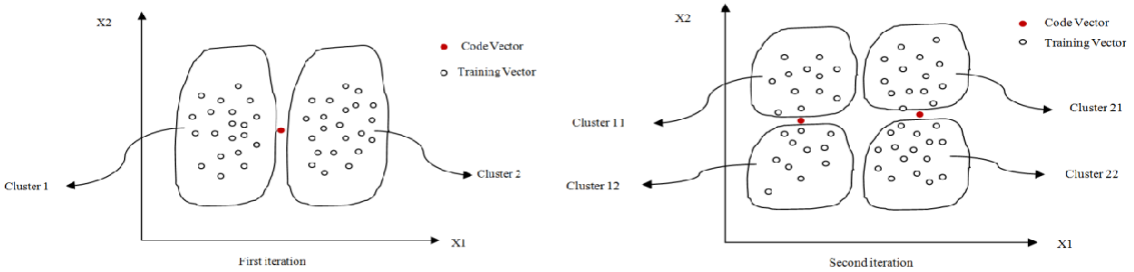


圖2、VQ 示意圖

圖 2 中左邊就是聲音特徵向量映射至一個二維平面的示意圖，綠色圓點代表語者一的所有特徵向量，經過運算後就會取距離該群體最接近的那筆資料作為代表特徵代替整個群，如此一來便可以將多數的資料化簡為四個主要向量，達到特徵壓縮的目的；而圖 2 中右邊就是我們在程式執行時分析出來的聲音特徵分布圖。

上文有提到每個說話人的碼本由 LBG 算法確定，用來識別說話人，計算說話人特徵與所有訓練碼本的距離（或失真）。

LBG 算法是一個迭代過程，基本思想是劃分訓練向量組並使用它從一組中找到最具代表性的向量。來自每組的這些代表性向量被收集起來以形成碼本。



a) b)

圖3、LBG 演算法示意圖

圖 3 中顯示了 LBG 演算法將質心分裂擴大與向量依照最鄰近碼字分群的概念。在 LBG 演算法中，左邊圖 a 中的質心向量先分裂為兩個相近向量，之後每個向量就依照最近質心分群得到圖中兩個向量群，而這兩個向量群的質心被更新進碼本後就是圖 b 中的碼向量（碼字），按照這個步驟拆分下去直到找出目標特徵數 M。

3、資料分群

有了壓縮後的資料（碼本），語音分群處理便可以進入最後階段，也就是開始執行分群演算法，此處使用的是前面介紹過的 K-Means。

(六)、語音辨識

在我們爬梳的文獻中有兩種語音辨識的方法，一種是使用 SpeechRecognition 模組，另一種則是 Whisper 套件。

1、SpeechRecognition

Python 有一個 SpeechRecognition 的模組，可以用來進行語音識別。這個模組可以訪問多種語音識別引擎（包括 Google Speech Recognition、IBM Speech to Text、CMU Sphinx）。SpeechRecognition 模組可以辨識許多種語言，只要在程式中修改語言的代碼，就能輸入不同語言的音檔。

2、OpenAI Whisper[5]

2022 年 9 月 21 日 OpenAI 發表「Whisper」神經網路。Whisper 是一種使用編碼器-解碼器架構的 Transformer 模型，也稱為序列到序列模型。它是一種自動語音識別（ASR）系統，使用從網路收集的 68 萬小時半監督學習標記的語音數據進行訓練。此外，它還支持多種語言的轉錄，以及將這些語言翻譯成英語。Whisper 共有五種模型尺寸（其中 large 和 large-v2 的架構完全一致，但性能較好），如表 1 所示，模型參數越大，所需要的顯存越大，而相對速度就越慢。除了尺寸最大的模型，其他四個尺寸皆有接受純英文訓練，訓練出的模型在速度和準確性上會比多語種的模型更好。

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39M	tiny.en	tiny	~1 GB	~32×
base	74M	base.en	base	~1 GB	~16×
small	244M	small.en	small	~2 GB	~6×
medium	769M	medium.en	medium	~5 GB	~2×
large	1550M	N/A	large (large-v2)	~10 GB	1×

表1、Whisper 模型相關資訊

3、Whisper-timestamped[6]

Whisper 模型被訓練來預測語音片段的近似時間戳（大多數時間準確度為 1 秒），但最初無法預測單詞時間戳，所以我們使用了 Whisper-timestamped 這個 Whisper 擴展儲存庫來實現預測單詞時間戳，而該儲存庫是基於動態時間校正 (DTW) 演算法。

4、動態時間校正(Dynamic Time Warping)[7]

DTW 是一種用於比較時間序列的方法，它不只可以應用於視頻、音頻和圖形數據的時間序列，任何可以變成線性序列的數據都可以進行分析。DTW 的主要概念是將兩個時間序列對齊，基本步驟是先計算兩個時間序列之間的距離矩陣，再根據距離矩陣，計算出一個最佳的對齊路徑，使得兩個時間序列之間的距離最小化。

(七)、環境音效辨識

1、MediaPipe audio classification

MediaPipe 在 2023 年初發表了一個新的音效分析工具，提供了偵測音訊類型的功能。這個音效分類器是透過預訓練分類好的音訊集來偵測音檔含有什麼樣類別的聲音，以每 0.975 秒為一個單位來切割音訊並分析，最後給出各個音效的預測分數，不過以上功能尚未完整整合進 Windows 系統內，暫時只能使用 Google Colab 環境執行。

2、Yamnet 模型

Yamnet 模型是與 YouTube 合作的大型音訊資料集，擁有許多種類的音訊資料[8]，為 MediaPipe audio classification 工具當中的音訊模型之一。而此模型供音訊模型各種音訊資料，就能分析輸入音訊中含有的聲音種類。本研究中將會使用這個模型作為後續實驗使用。

二、研究架構圖

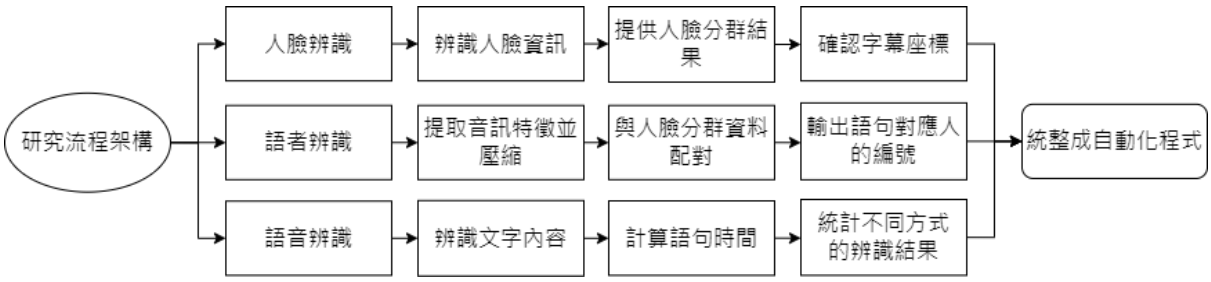


圖4、研究架構圖

如圖 4 所示在我們研究的一開始會先將我們希望達成的情境化字幕分為三個部分研究與實驗，分別就是研究架構圖中的人臉辨識、語者辨識和語音辨識，進行完各種實驗後，會將效果最好的方式統整起來，成為一個自動化的程式，實現為影片自動嵌入情境化字幕的目標。

三、偵測說話者:

要將字幕標示在說話者旁，我們就要先知道畫面中所有人臉位置，並判斷哪位人物是當前的說話者。本研究使用的 MediaPipe Face Mesh 同時包含人臉偵測和標記人臉特徵點的功能，故可得知人臉位置。而判斷說話者的方式就是觀察畫面中人物「嘴巴張合」的情形，以圖 5 為例：

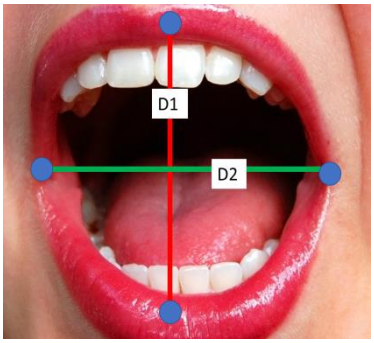


圖5、D1 和 D2 示意圖

圖 5 中的 D1 和 D2 分別是上下唇間距離和左右嘴角間距離，我們利用 D1 和 D2 間的比例關係來判斷當前畫面人物的嘴是張或合 ($\frac{D2}{D1} \leq \alpha$為張嘴)。但由於說話時嘴巴會有張有合，且笑的人雖嘴巴一直張著但並不是說話者，於是我們取當前畫面前 n 幀同人物的嘴巴張合數據做紀錄，若前 n 幀人物嘴巴有張有合則判斷此人物為說話者，否則若前 n 幀嘴巴皆合或皆張(可能在張嘴笑)，則判斷此人物非說話者。

圖 6 為偵測說話者的流程圖：

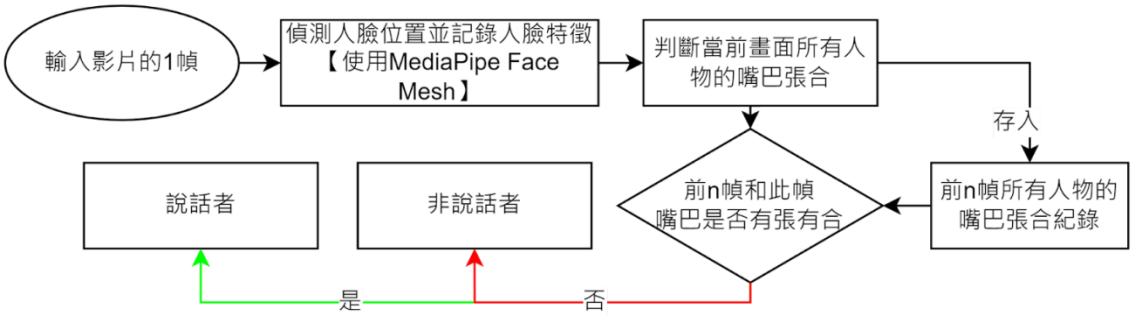


圖6、偵測說話者流程圖

四、人臉辨識(Face Recognition):

由於影片畫面時常會出現多人快速輪流說話的情形，導致偵測到不只一位說話者，難以判斷字幕的嵌入座標。於是我們利用人臉辨識來辨識畫面中被判定為「說話者」的身分，再將已標記語者的字幕標記於正確語者邊。

圖 7 為人臉辨識的實作流程圖，共分為人臉偵測、人臉對齊、特徵提取、特徵聚類和人臉辨識五個步驟，並介紹了各步驟的說明及實作方式：

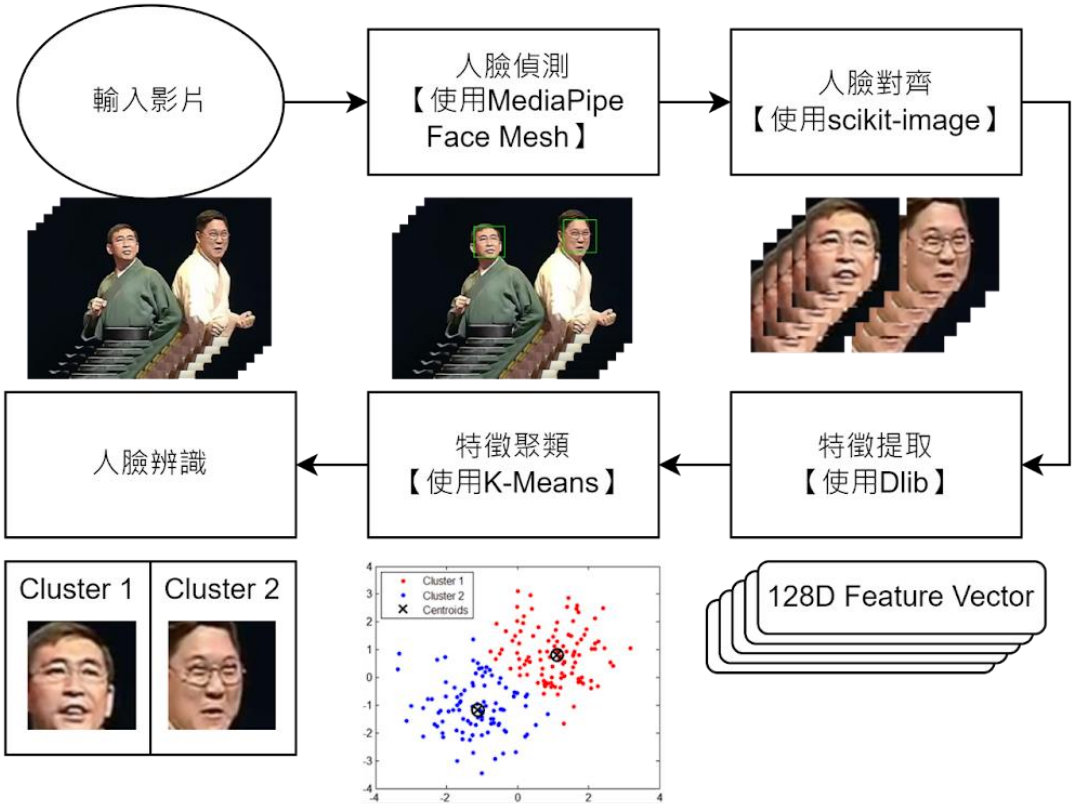


圖7、人臉辨識流程圖

(一)、人臉偵測 (Face Detection)：

人臉偵測的目的為檢測影像中是否存在人臉，本研究使用 MediaPipe Face Mesh 作為人臉偵測的工具。相比於其他人臉偵測的分類器，MediaPipe Face Mesh 可以在偵測時取得 468 個臉部特徵點，省去後續重新偵測臉部特徵點的步驟。

(二)、人臉對齊 (Face Alignment) :

人臉對齊的目的為將偵測到的人臉校正到同一標準的大小與角度，以便後續的特徵提取，本研究使用 scikit-image 進行人臉對齊。scikit-image 是一個基於 Python 語言的開源圖像處理庫，利用人臉的五個特徵座標就能將圖片旋轉、切割，使圖像標準化。

(三)、特徵提取 (Feature Extraction)：

從對齊後的人臉圖像中提取特徵向量，以便後續進行特徵聚類，本研究使用 Dlib 進行特徵提取。Dlib 是一個跨平台的 C++函示庫，其中人臉識別模塊的主要功能是將人臉圖像轉換成 128 維的特徵向量，作為機器學習的資料。

(四)、特徵聚類 (Feature Clustering)：

將提取到的特徵向量進行聚類，聚類指的是將特徵以及屬性不同的物件通過靜態分類分成不同的組別。本研究使用 K-Means Clustering 進行特徵聚類，我們也利用輪廓分析法使演算法能自動將資料分成最佳的組數。此時分群的模型已經建立好，意味著所有的人臉都有各自對應的組別。

(五)、人臉辨識 (Face Recognition) :

由於模型已建立好，只需取得未知人臉的特徵向量，即可使用模型來判斷此未知人臉的所屬組別為何，達成人臉辨識的目的。

經過以上五步驟，即可辨識出每個畫面中出現人物的身分，在畫面中有不只一人被判斷為「說話者」時，就能將語者辨識標記好的字幕標示於對應語者的身旁，提升字幕標示的精確度。

五、泡泡字幕

為了使聽障人士在觀看影片時有最好的體驗，我們找出以下三個聽障人士在觀影時會遇到的困難來當作改善字幕的方向：

1、無法辨識畫面中各個字幕的語者為何

2、字幕切換過快，造成閱讀上的困難

3、影片出現音效時聽障人士無法得知，造成資訊的落差

為了解決以上三個困難我們拋棄了標示在畫面下方的傳統字幕，使用會逐漸上飄並跟隨語者的泡泡字幕。為了找出對應字幕的語者，我們利用先前提到的「偵測說話者」以及「人臉辨識」兩種方式判斷不同情況下的各字幕的語者，如圖 8 流程圖所示。

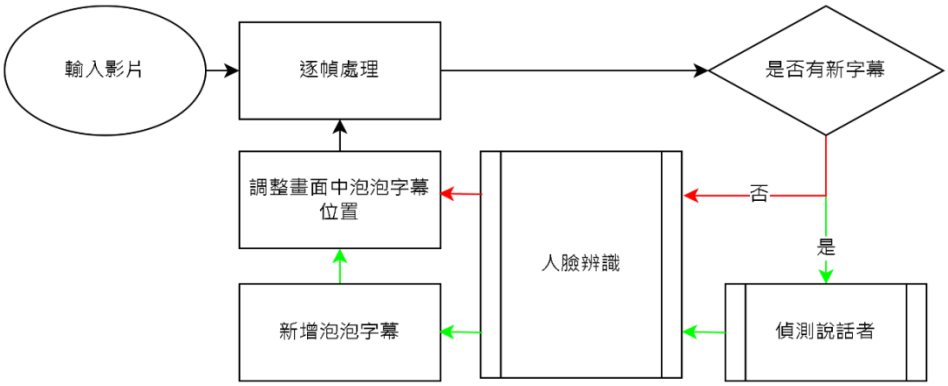


圖8、泡泡字幕生成流程圖

新增字幕的同時我們也紀錄字幕的語者，因此人臉辨識後能使所有畫面中的字幕跟隨著各自對應的語者，解決聽障人士無法辨識字幕語者的問題。而泡泡字幕被設定成直到飄出螢幕範圍或出現時間超過 5 秒（避免畫面字幕過多過度雜亂）之前都不會消失，解決傳統字幕切換過快的問題。而關於聽障人士無法得知影片音效的問題會在後續的「環境音效辨識」介紹解決方式。

六、語者辨識

要在加入字幕時可以準確的標示在正確語者旁邊，需要知道每段音訊是由誰說出的，由「偵測說話者」這步驟我們可以知道哪段語句畫面中只有一人說話並將其正確標記，不過一旦遇到畫面中偵測到有多個人都張開嘴混淆「偵測說話者」時，就需要更明確的語者辨識。為此，我們會先將影片所有語句聲音片段進行聲音分群，使語句被簡單分類為數個語者後，再透過畫面中人臉說話資訊確切地將語句群對照給每個人

臉群，如此一來便可完成音訊與人臉的對照，處理多人說話的畫面也能依照辨識出來的結果準確標記。

(一)、語者分群(Speaker Clustering)

語者分群處理過程如下：

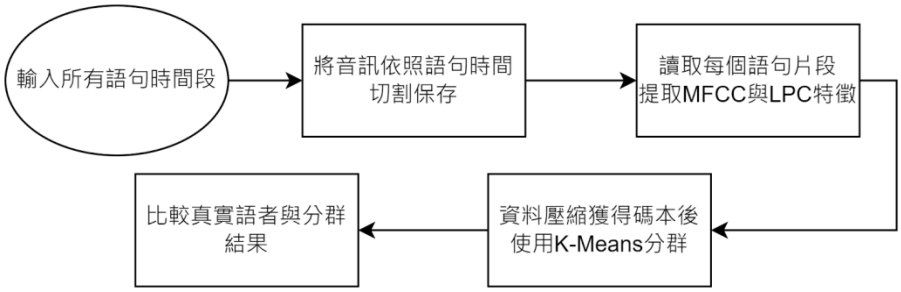


圖9、語者分群流程圖

最一開始獲得語句時間後，我們會將每個語句的音檔進行特徵提取，分析完 MFCC 以及 LPC 特徵，我們會分析分群輸出結果來跟手動分析的結果計算準確率，選擇效果較好的特徵，作為後續研究的主要依據。

(二)、語句分類

目前我們有了語音的分群結果，但是仍然不清楚哪個聲音群是對應畫面中哪個人臉，因此我們需要借助畫面中的一些資訊，結合人臉分群的結果，才能知道語句對應的語者臉部。

當畫面中僅有一人開口說話的時候，該時段的語句就會被認定為畫面中該人的聲音，透過找尋僅有一人的畫面片段，我們便可以獲得部分語句分群對應人臉分群的資訊，再運用統計的方式，從而將語句群跟人臉群建立配對。

圖 10 是語句分類的詳細說明步驟示意圖以及文字說明：

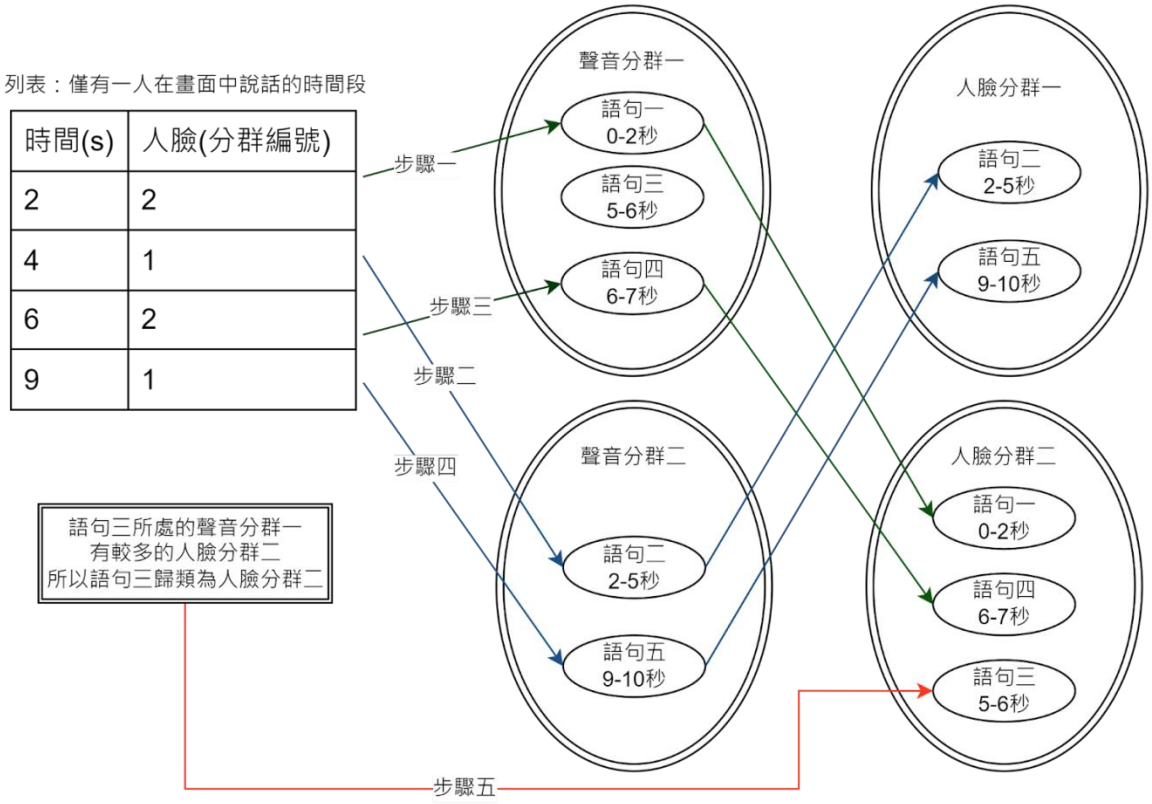


圖10、語句分類示意圖

步驟說明：

1、對照列表第一列，第二秒時(語句一)只有人臉二在講話，將語句一分給人臉分群二。

2、對照列表第二列，第四秒時(語句二)只有人臉一在講話，將語句二分給人臉分群一。

3、對照列表第三列，第六秒時(語句四)只有人臉二在講話，將語句四分給人臉分群二。

4、對照列表第四列，第九秒時(語句五)只有人臉一在講話，將語句五分給人臉分群一。

5、列表已經對照完而語句三尚未對應到人臉，統計出語句三所在的聲音分群一有較多人臉分群二，因此將語句三分類給人臉分群二。

如此一來，已經將所有的語句跟語者配對完成了，可以使複雜畫面也能精準標記語者，後續研究結果我們將會討論分群標記的準確度以及兩種聲音特徵的效果差異。

七、語音辨識

語音辨識的流程圖如下：

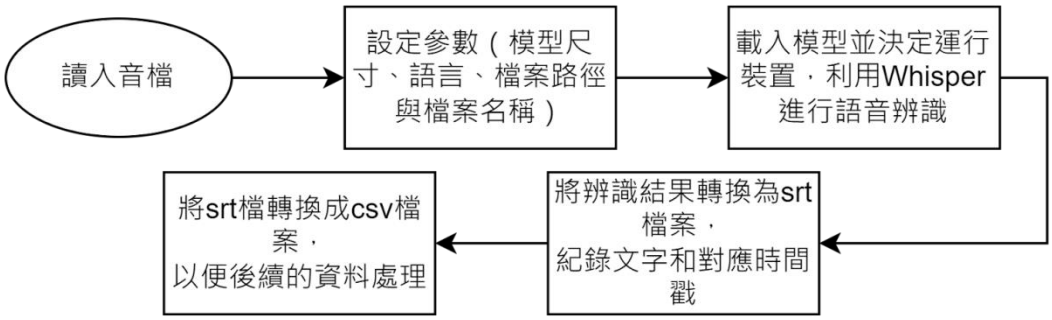


圖11、語音辨識流程圖

將目標音檔讀入後，程式會依據設定的模型和語言，輸出標示編號、起始時間、結束時間和相對應字幕的 srt 檔案。為了讓後續程式方便取用資料，再將生成的 srt 檔轉換為 csv 檔。

八、環境音效辨識

環境音效辨識流程圖：

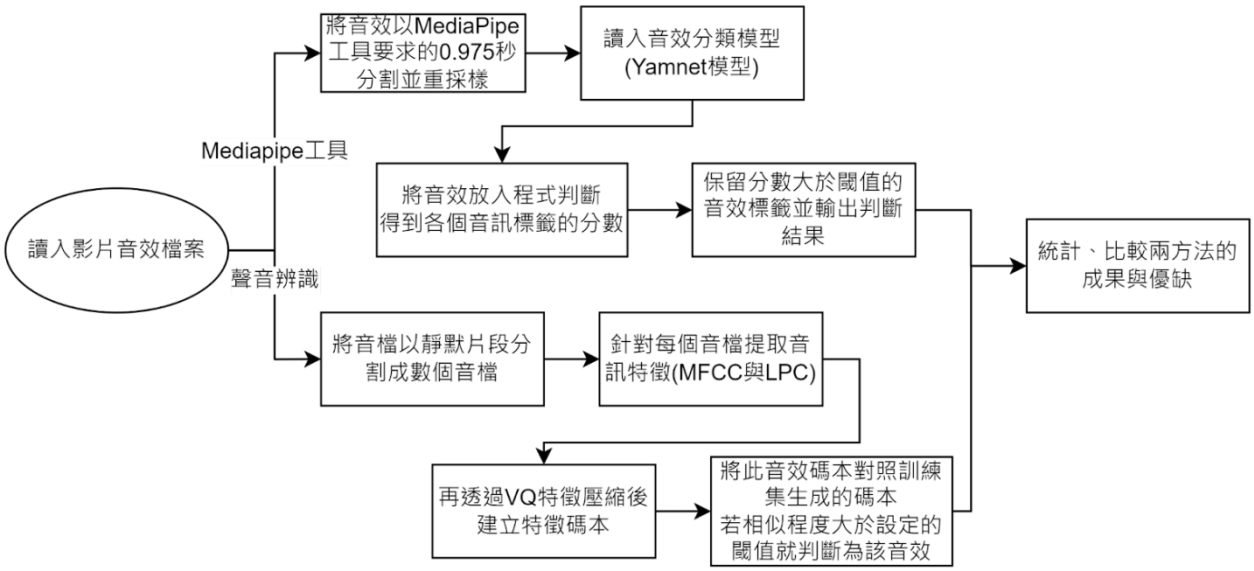


圖12、環境音效辨識流程圖

為了能夠將影片中的聽覺訊息轉換成視覺化提示字卡，我們需要針對影片的聲音進行音訊分析與分類，將這些特殊的音效辨識出來，嵌入影片畫面，成為情境化字幕的一部分。

而我們針對這個目的設想了一個辦法，就是使用語者辨識[9]這套演算法。語者辨識主要是偵測語者的音色特徵來當作辨別標準，而我們認為各種特殊的環境音效也會有相對應的音色特徵，足以分辨不同音效，但是同時我們也想到這個聲音辨識方法會需要許多提前標籤好的大量音效庫，在效率以及應用廣泛程度上可能較受限制，因此我們另外找到了一個音訊分類的工具是 Mediapipe 的 audio classification 模組，這個工具的特色就是它引用了大量的音效訓練集且在判斷上多了音效相似分數(可以理解為有多少比例是相似於某個音效)，可以更方便我們設立顯示音效字卡時的閾值，僅顯示較重要且分數較高的音效，提升音效字卡的品質。

此外，聲音辨識的訓練集是由我們自行找尋該音效數個音檔，提取特徵後做成碼本，將所有訓練集音檔碼本取平均就是最終音效碼本。接下來辨識音效的時候我們會將待辨識音訊讀入，提取出特徵後轉成碼本，與所有已經經過訓練的音效資料進行比對，找出差異最小(特徵距離最近)的作為辨識結果輸出。

九、統整自動化程式

有了以上的各部分成果，我們會按照圖 13 的流程圖，透過 Python 將其統整成一個完整的程式，只需將原影片輸入就能自動嵌入情境化字幕。而後續我們將會在研究結果部分進一步探討這個完整的程式處理影片的效率與效果。

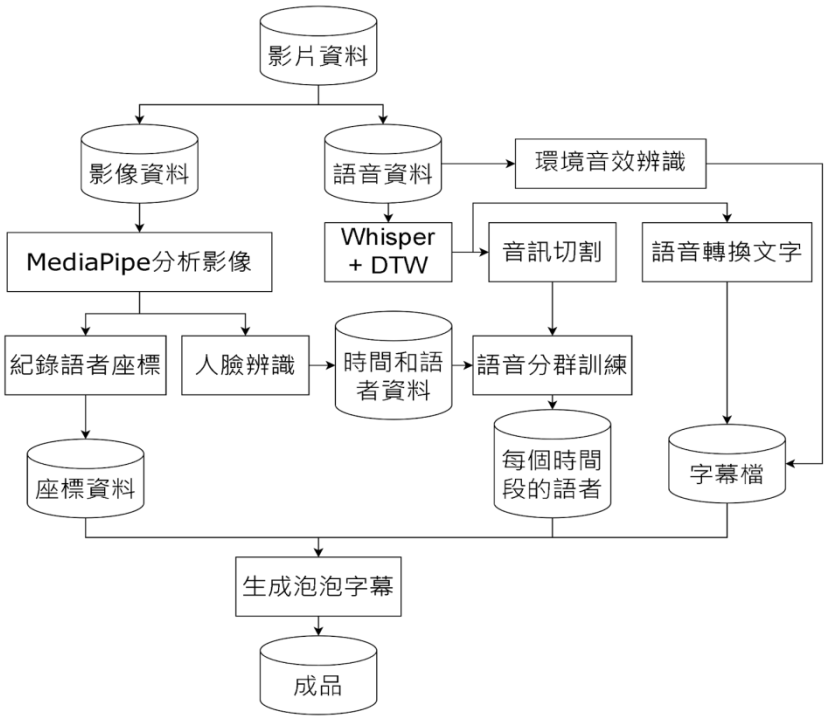


圖13、影片自動化生成流程圖

肆、 研究結果

一、素材來源及編號：

素材編號	影片名稱	來源	影片人數與類型
A	【三重標準】剪頭髮的例子	YouTube	單人（演講）
B	【那些讓我們難以習慣的台式口味】請不要送我們這些東西	YouTube	雙人（談話）
C	壹加壹男蠢女醜沒看點？情侶 YouTuber 刻意秀恩愛很假很尷尬！| #酸民說 ft.壹加壹	YouTube	三人（訪談）
D	雙蕨之間	Netflix	雙人（英文訪談）
E	一把青	Netflix	三人（有環境音效）

表2、影片素材說明

二、人臉辨識的效果：

下表是關於各個素材的人臉辨識成果，分別討論各影片的分群數量以及人臉辨識成功率，其中人臉辨識成功率的計算方式為 ：

$$\frac{\text{辨識正確人臉數}}{\text{所有人臉數}} \times 100\%$$

素材編號	影片人數與類型	分群數量	人臉辨識成功率
B	雙人 (談話)	2	100%
C	三人 (訪談)	4	98.7%
D	雙人 (英文訪談)	2	100%

表3、各素材人臉辨識成功率

由表 3 的結果可知，三個影片素材的人臉辨識都十分精確，能夠正確的分辨不同人物的身分。值得注意的是素材 C 的結果，影片人數僅三人卻分成了四個群組，其原因是該影片中有放入除了主要三位說話者之外的人臉照片，故人臉辨識也分出第四個群組對應多出來的人臉。

三、語句切割的效果：

表 4 呈現的是利用 Whisper 和 Whisper-timestamped 生成的字幕起始與結束時間進行語句切割，以及使用空白切割的正確率比較。

語句切割正確率的計算方式為：

$$\frac{\text{含完整語句片段數量}}{\text{切割片段總數量}} \times 100\%$$

切割方式	空白切割	Whisper	Whisper-timestamped
語句切割正確率	85.7%	90.5%	92.9%

表4、不同切割語句方式的正確率

由表 4 可見，使用 Whisper-timestamped 生成的時間戳記確實較為準確，比起其他組別有較多完整語句。空白切割的正確率較其餘兩者低，推測是影片中說話時節奏較快，語句中沒有明顯的靜默時段。

四、語者辨識結果：

測試語音辨識效果時我們採用了兩種不同的語音特徵以及三種方式對應三個素材，藉以比較不同處理方式的辨識效果差異。

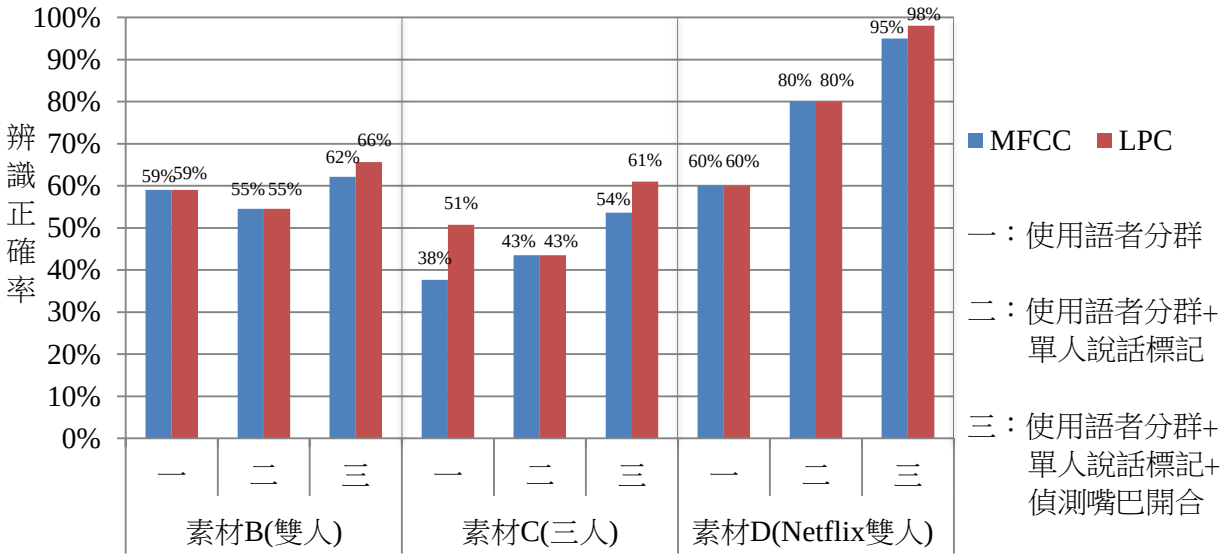


圖14、語者辨識正確率比較圖

透過圖 14 發現，大部分素材無論聲音特徵使用 MFCC 還是 LPC 效果都差異不大，而隨著語者辨識標記方式增加，準確度都會有一些成長，但是還是會因為素材類型差異而有不同的效果，像是素材 B 跟素材 C 大部分都是有多人同時在鏡頭內，那在偵測

嘴巴開合與標記字幕時就很容易混淆導致準確度不好，素材 D 則有許多的單人畫面使字幕可以被精準判斷為正確的語者。

五、語音辨識結果：

下表是針對每部素材在不同模型或不同套件的辨識下字幕的正確率。正確率的計算方式為：

$$\left[1-\frac{\text{差異字數}}{\max(\text{輸出結果字數},\ \text{正確字幕字數})}\right]\times 100\%$$

其中差異字數是根據編輯距離得出的。

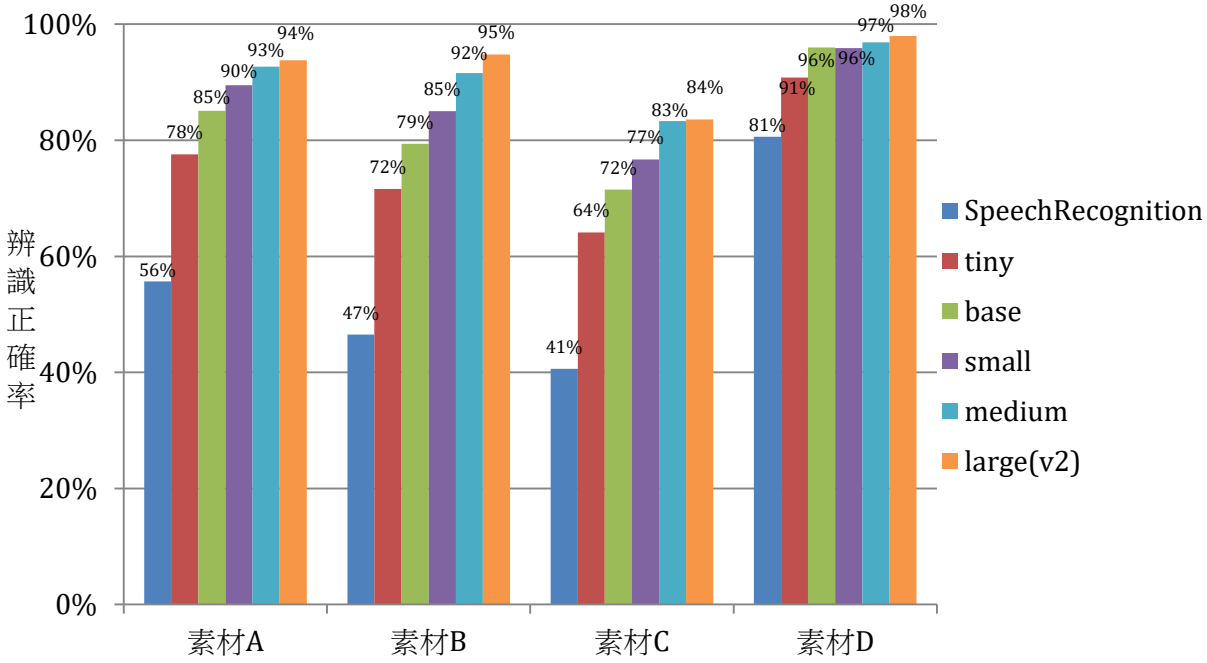


圖15、語音辨識正確率比較圖

在素材 A 和 B 中，medium 和 large 間的差值皆是該組最小（1.1%和 3.2%），顯示出隨著模型尺寸放大，正確率的提升會漸漸趨緩。素材 C 中，medium 對比 large 模型的語音辨識正確率只相差 0.3%，但執行時間卻是 large 的一半，因此使用 medium 模型的效益較高。而素材 D 的任何模型正確率皆在 90%以上，可能是因為英文有較多訓練資料，因此成效較好。

六、環境音效辨識效果：

測試音效辨識成功率時，我們會隨機挑選該種音效的 20 個音檔，計算成功辨識的數量以換算辨識成功率。

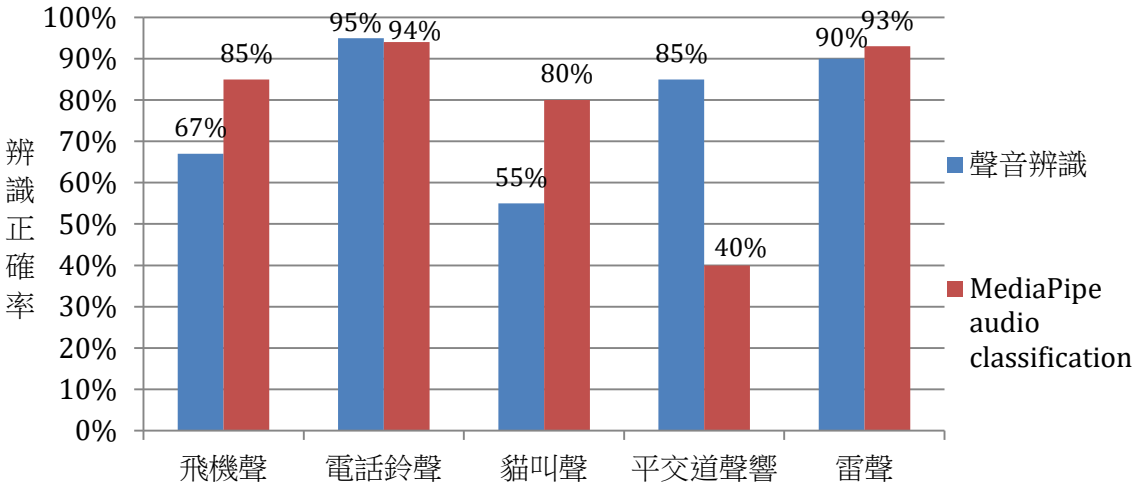


圖16、環境音效辨識成功率

從圖 16 中我們可以發現大部分音效 MediaPipe 音效辨識工具的效果都比我們自行設計訓練的聲音辨識好，唯獨平交道警聲的正確率比較低，我們推測是因為實驗使用的聲音分類器模型中較少平交道聲音的訓練資料，以至於遇到該音效時無法準確辨識。

七、影片處理效率：

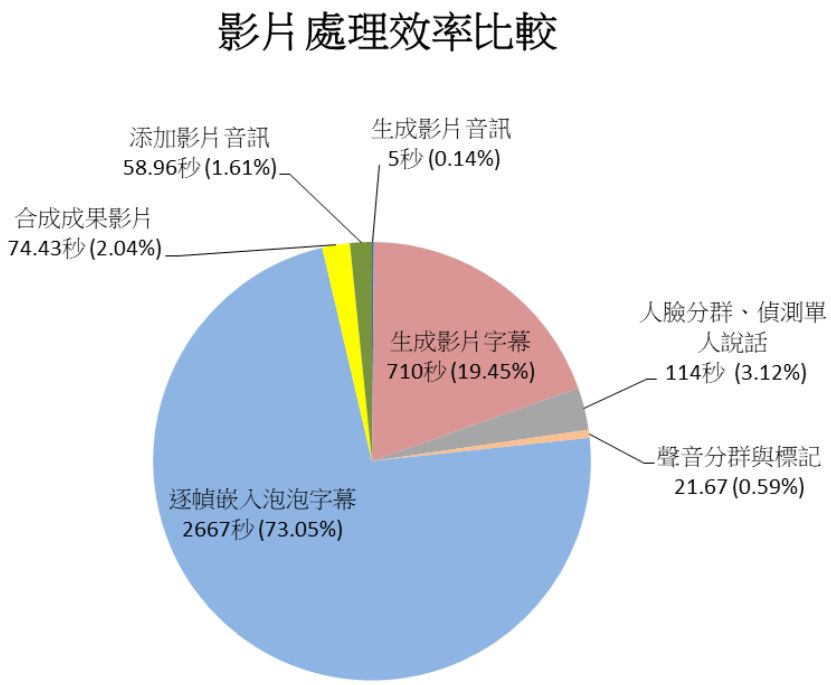


圖17、素材 B 影片生成效率比較圖

圖 17 顯示素材 B 兩分鐘的影片生成耗時約等於影片時長的 26 倍，最花時間的是「逐幀添加泡泡字幕」與「生成影片字幕」這兩個步驟，除了生成影片字幕會因為使用的 Whisper 模型不同影響生成速率(可以參考表 1、Whisper 相關模型相關資訊中的耗時倍率，此處素材 B 是採用 large-v2 的成果)，「逐幀添加泡泡字幕」因為要將影片的每一個畫面都進行人臉偵測與嵌入文字，所以花費最久的時間。

八、自動化處理影片成果：

圖18、素材 A 結果畫面

圖 18 顯示了單人畫面下的泡泡字幕效果，字幕會上飄停留較久時間，使觀影者有更充足的時間閱讀字幕理解劇情。

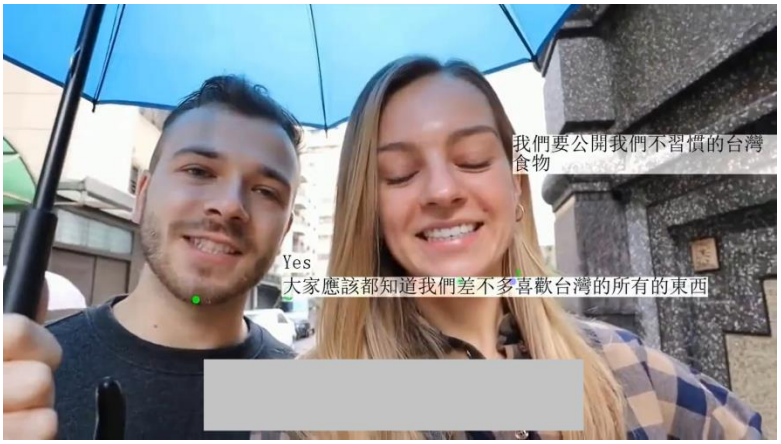


圖19、素材 B 結果畫面

圖 19 展示了雙人同框下泡泡字幕的畫面效果，字幕會透過語者辨識的結果嵌入在正確的語者臉旁。

圖20、素材 C 結果畫面

圖 20 展現了三人同框下影片處理的效果，同樣是結合語者辨識的結果使字幕可以嵌入在正確的語者臉旁。

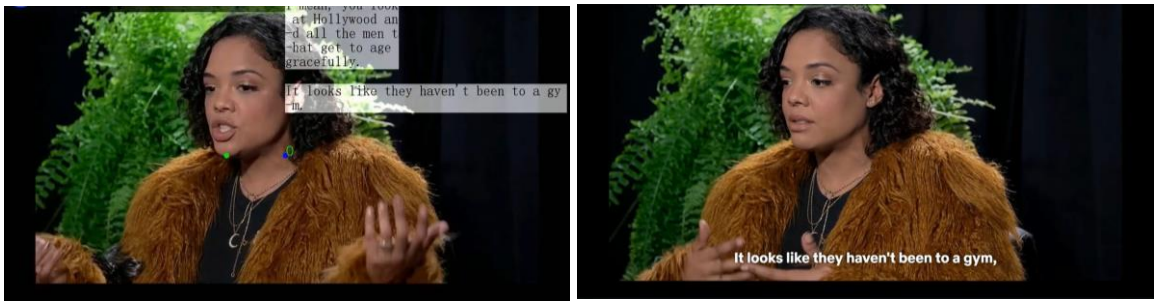


圖21、素材 D 結果畫面(左) 對照 Netflix 影片字幕(右)

圖 21 展示了結果影片的泡泡字幕效果與 Netflix 影片結果對比，我們能夠將字幕標記在語者臉旁並且在處理英文影片時字幕也可以很準確。另外可以發現我們的成果畫面中上一句的字幕還顯示在畫面上，相比 Netflix 影片的字幕已經換到下一句了，這就是泡泡字幕提供較多時間閱讀的優點。

圖22、素材 E 結果畫面(左)對照 Netflix 影片字幕(右)

圖 22 主要是展示我們的環境音效提示字卡效果對比 Netflix 影片描述性字幕，在使用音效辨識下我們可以達到與 Netflix 影片字幕相同的效果。

伍、 討論

一、人臉辨識效果：

人臉辨識有著 98.7%-100%的辨識成功率，可精準地辨識影片中各個人物的身分。在「偵測說話者」與「語者辨識」的協助下，能夠將字幕標示到對應的語者旁，使聽障人士能輕鬆地得知不同字幕的語者。

二、語句切割效果：

Whisper-timestamped 的單詞時間戳比 Whisper 更準確，且使用兩者提供的時間戳記進行語句切割效果比空白切割好。

三、語者辨識效果：

隨著輔助語者標記的方式增加，辨識的準確度也都有隨之上升，同時兩種特徵辨識方式準確度相差不大，LPC 微微勝出而已。而素材的不同辨識率有 60%-90%的效果，而語者辨識效果不彰的素材我們認為最主要還是受限於一開始語者分群的狀況不理想，未來可從音訊特徵選用、提取與分群方式去做更多的嘗試才能較好分辨不同語者。

四、語音辨識效果：

由研究結果中發現，在使用 Whisper 套件下，每部素材利用 large 模型進行語音辨識的正確率都為同組最高，而 tiny 模型的辨識正確率皆最低。SpeechRecognition 在各素材下的辨識正確率相較於 tiny 都低了不少。另外，Whisper 每種尺寸的純英文模型皆比多語言模型的辨識正確率高。

五、環境音效辨識效果：

聲音辨識方法需要大量的音檔作為訓練集，而且隨著訓練集內的音效標籤數量增加也更容易誤判使得辨識結果出錯，難以真正實現自動化製作音效字卡。另一方面 MediaPipe 的音效辨識成功率也是相當精準，穩定度與音效標籤(描述音效的字詞)精確度都較高，不過因為音效訓練庫受限於該工具支援的模型，可辨識的音效標籤也較難自行調整，靈活性相對較低。

六、影片處理效率分析：

針對影片處理效率低落的狀況，首先要改善的是最耗時間的生成泡泡字幕這個步驟，目前想到的解決的方案是利用「轉場偵測」來提高效率，每一個場景只需要進行一次人臉偵測，並假設人物間的相對位置不會更動，就能利用人物間的相對位置得知所有人的身分，避免每幀都進行人臉辨識所耗費的大量時間。

七、自動化影片處理成果畫面分析：

在目前的結果畫面中可以發現泡泡字幕增加的閱讀時間以及語者標記帶來的分辨語者的效果都有達到我們預期的目標，同時跟商業化的 Netflix 字幕相比無論語言、字幕正確率和音效描述都有達到相同的水準。

八、聽障受試者回饋

為了瞭解嵌入情境化字幕的影片是否能夠真正地改善聽障人士的觀影體驗，我們實地採訪了聽障協進會的理事長。理事長本身是輕度聽障人士，並接觸過各類型的聽障人士，包括中、重度的聽障人士。以下是我們統整理事長在觀看結果影片後所給出的回饋：

(一)、情境化字幕的優點:

1、逐幀上飄的泡泡字幕能避免傳統字幕切換過快的問題

2、能得知各個字幕的語者

3、環境音效字卡能幫助理解劇情

(二)、情境化字幕的缺點：

1、泡泡字幕的出現位置不固定，影響辨識

2、泡泡字幕跟隨人物移動會造成閱讀的困難

3、缺乏箭頭等標示物標示語者，影響判斷字幕語者

4、習慣傳統字幕的呈現方式，需要時間適應新的情境化字幕

(三)、建議的改進方向：

1、不同語者的字幕可用不同顏色呈現

2、固定字幕的出現位置，維持畫面穩定

3、語者辨識須更精準

4、可將整部影片放慢，增加閱讀時間

最後理事長也提到，此系統對於聽障人士，尤其是中、重度聽障人士有非常大的助益。若再對系統進行優化和改良，在未來必定能成為聽障人士不可或缺的觀影輔助工具。

陸、 結論

一、聽障人士的無障礙訴求：

本研究著重在為聽障人士創造無障礙的觀影體驗。我們所撰寫的程式能夠自動將影片進行處理，生成跟隨語者移動的泡泡字幕以及環境音效字幕，消除聽障人士在觀影時遭遇的不平等。

二、人臉辨識：

使用 MediaPipe Face Mesh、Dlib、K-Means 工具和演算法實現人臉辨識，達成 98.7%以上的辨識成功率，在偵錯說話者與語者辨識的協助下能有效的辨識每個字幕的語者。

三、語句切割：

在將語句聲音分群時需要有語句的完整時間才能截取出正確的音檔，同時標記字幕時也需要字幕起始時間，在實驗中我們嘗試了針對音訊空白處切割、Whisper 模型輸出的語句時間與 Whisper-timestamped 等方式，並且最終語句切割準確度可達 90%以上。

四、語者辨識：

為了使字幕能夠正確的標示語者，研究中我們透過了語者聲音分群加上畫面中單人說話的時段以及語者嘴巴開合等方式判斷每個語句對應畫面中的語者，且效果隨著

不同的影片類型有著 60%-90%左右的辨識成功度，越多語者的影片標記上越為困難，精準度也就受到影響。

五、語音辨識：

研究中我們嘗試了使用 Whisper 的模型與 SpeechRecognition 套件協助我們進行語音轉文字的工作，同時探討了 Whisper 不同模型辨識文字正確率與時間效率差異，使影片的字幕準確率達到 90%以上，保證了成果影片的字幕貼近影片原語句。

六、環境音效辨識：

我們比較聲音辨識演算法與 MediaPipe 音效分類工具在實現環境音效辨識時的效果，發現在音效標籤種類與描述音效精準度上都是 MediaPipe 音效分類工具效果較好，不過在增加辨識特定音效與調整判斷音效的分數閾值上聲音辨識演算法的靈活度較高。

七、自動化生成效果：

目前研究做出的自動化生成影片程式都有達到研究目的中的效果，並且跟商業化 Netflix 影片的字幕相比可以做到相同的水準，儘管目前影片處理效率無法做到處理時間與影片時間等長，但是自動化的處理還是能提供不少的方便性。

八、聽障人士的回饋：

我們採訪了有許多與聽障人士互動經驗且自身也為輕中度聽障的聽障協進會理事長，在看完本系統產生的結果影片後也認定本系統可以為聽障人士在觀看影片時帶來一定程度上的幫助和便利，特別是重度聽障的聾啞人士。

九、未來展望：

(一)、人臉辨識更進步：

近年來戴口罩成為了影片中常出現的情況，我們在查找文獻時有發現戴口罩下的人臉辨識研究，未來會嘗試將這個技術融入研究中使作品更全能。

(二)、語者辨識再加強：

改善目前聲音分群的結果，從聲音特徵選用、提取方式與分群方式調整嘗試，進一步提升語者辨識的精準度。

(三)、環境音效程式整合：

目前因為偵測環境音效效果較好的 MediaPipe audio classification 工具尚未支援 Windows 系統，暫時只能使用 Google Colab 環境使用，將來開發者整合完畢後可以將偵測環境音效的程式碼結合進自動化 Python 程式中。

(四)、影片處理效率改善

可以改善先前討論到的語音辨識可以使用辨識效果差不多但速度更快的 medium 模型外以及增加轉場偵測避免每幀人臉辨識耗費大量時間，使系統處理速度更快、性價比更高。

(五)、實時的自適應字幕標記回饋：

未來希望這套系統可以嘗試應用在實體場合即時顯示結果字幕（例如：實體演講），加上擴增實境技術達到實時標示語者和顯示語句的效果，且隨著語句增多自適應系統就可以強化語者辨識達到準確標示語者的功能。如此一來，這項技術與裝置可以補足沒有提供手語的場合，讓聽障人士在接收實時資訊的時候更為方便。

柒、 參考資料與其他

參考文獻資料

[1]、陳妤甄. (2021, January). 聽覺障礙者使用同步聽打服務經驗之探究，取自 https://viis.ntl.edu.tw/ntldo/resources/e/7/e7f6ece2c7d4965d76c87823acc787d1/110%E7%8D%8E_%E9%99%B3%E5%A6%A4%E7%94%84_%E8%81%BD%E8%A6%BA%E9%9A%9C%E7%A4%99%E8%80%85%E4%BD%BF%E7%94%A8%E5%90%8C%E6%AD%A5%E8%81%BD%E6%89%93%E6%9C%8D%E5%8B%99%E7%B6%93%E9%A9%97%E4%B9%8B%E6%8E%A2%E7%A9%B6.pdf

[2]、Fernandez, J. MediaPipe Face Mesh. from https://github.com/google/mediapipe/blob/master/docs/solutions/face_mesh.md

[3]、Wang, J. (2018, April 9). K 平均法 (K Means). Retrieved April 8, 2023, from https://rstudio-pubs-static.s3.amazonaws.com/378455_ddbefe5075b941d1a1f6a1bf9cf1e85f.html

[4]、Peng, Y. H. (2018, April). SpeechBubbles: Enhancing Captioning Experiences for Deaf and Hard-of-Hearing People in Group Conversations. from https://www.yihaopeng.tw/pdf/CHI18_SpeechBubbles.pdf

[5]、OpenAI Whisper：

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. ArXiv Preprint ArXiv:2212.04356.

[6]、whisper-timestamped：

Louradour, J. (2023). whisper-timestamped. GitHub Repository. from https://github.com/linto-ai/whisper-timestamped

[7]、Dynamic-Time-Warping：

Giorgino, T. (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. Journal of Statistical Software, 31(7). doi:10.18637/jss.v031.i07

[8]、Mediapipe 網站提供 Yamnet 模型音訊分類標籤列表，取自 https://storage.googleapis.com/mediapipe-tasks/audio_classifier/yamnet_label_list.txt

[9]、Das, O. SPEAKER RECOGNITION. from https://ccrma.stanford.edu/~orchi/Documents/speaker_recognition_report.pdf

【評語】052507

1. 此作品在改善聽障人士無法完整接收影音類型資訊的狀況，探討各種影片處理技術，尋找、嘗試並比較各種方法，整合出最適合的系統自動替影片嵌入情境化字幕。作品具實用性，且作品完整度高。

2. 本作品的名稱為「影片情境化」，似乎與內容不符合，情境化描述指的應是"由當事人的動作、關係來描述當下發生的情境"。題目的構思可以再精準一些。

3. 本作品的實驗資料需要再增加一些，尤其是因緣辨識，聲音轉文字等。

作品海報

影片情境化字幕實現探討

研究動機

平時我們接觸的網路媒體多數都是影音的形式，如直播、新聞、娛樂電影、政治節目與政策說明會等等，然而並非所有內容都會加上字幕，聽障人士也就無法完整地理解這些資訊。因此為了保障他們的權益，大部分的政見說明會都會有手語協助他們了解內容，美國電影院更是推出了隱藏式字幕來讓所有人都能享受影音樂趣，但是以上這些服務成本較高且難以普及。

而聯合國於2015年宣布了「2030永續發展目標」(Sustainable Development Goals, SDGs)，第十項目標「消除不平等」強調要保障身心障礙人士的權益。因此我們想實現一套系統，可以自動化將影片聽覺訊息視覺化呈現，保障這類族群媒體接收的權利；同時研究當中我們也參考商業化的影片字幕效果，探討我們系統可以改進的方向。

研究目的

一、透過語音辨識將原影片音檔自動轉換為文字

二、透過語者分群辨識來判斷影片中的說話者

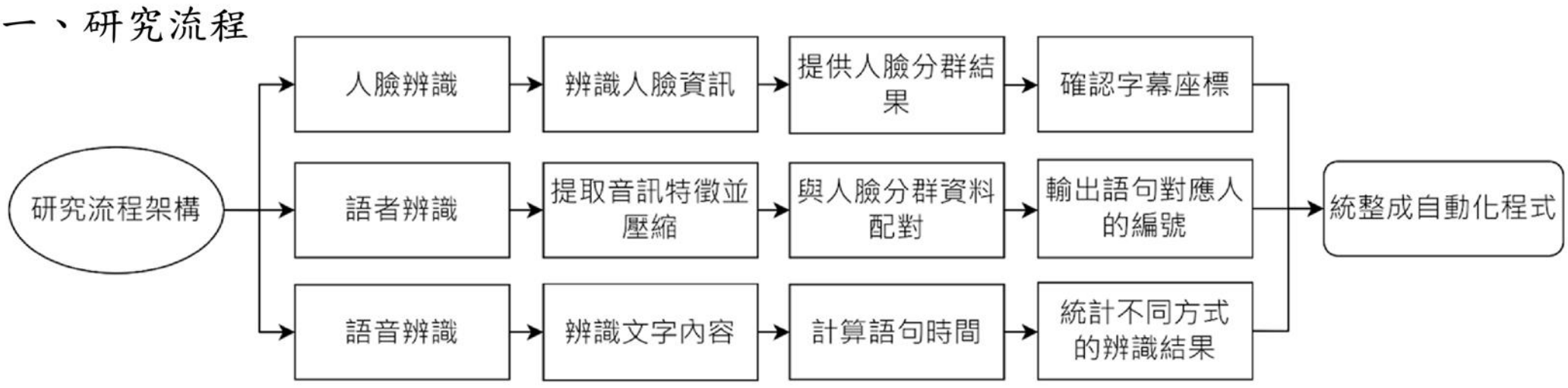
三、結合人臉分群辨識來達到將字幕標示於說話者旁

四、將環境音效標示於畫面中

五、將上述功能整合成自動化流程，為一部影片嵌入情境化字幕

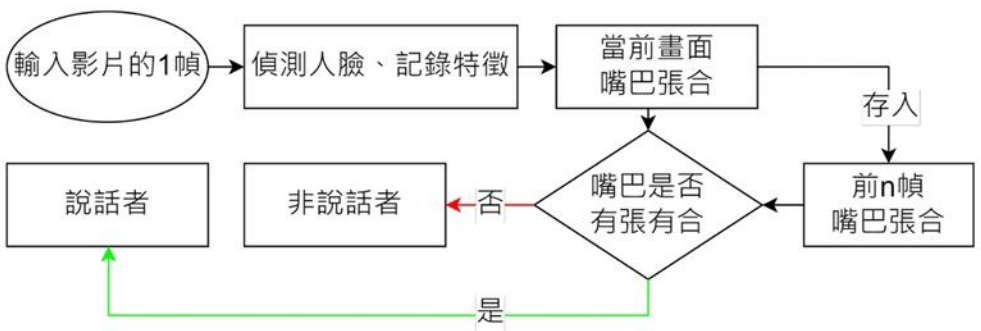
研究方法

一、研究流程



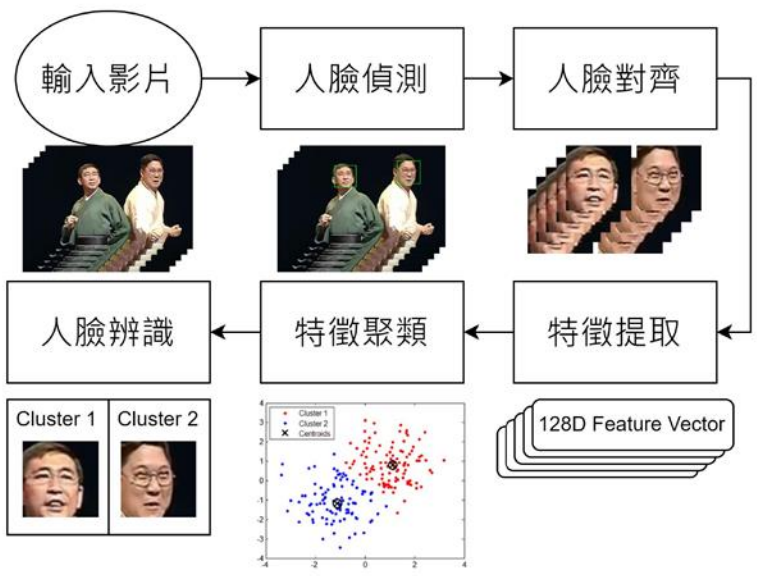
二、偵測說話者

本研究利用人物嘴巴開合情形來判斷畫面中說話者，為避免將笑的人誤判為說話者，我們將畫面前n幀的嘴巴張合情形記錄下來。當前n幀和此幀的嘴巴有張有合時則判斷此人物為說話者。



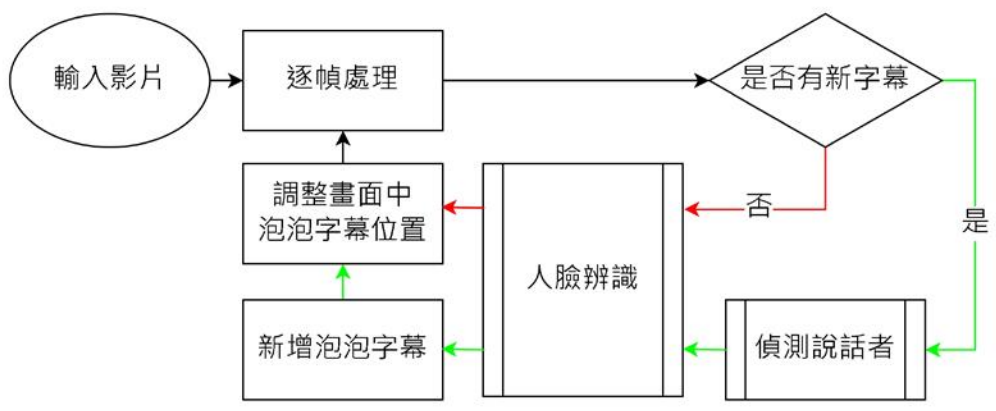
三、人臉辨識

畫面中若偵測出二位以上的說話者，則需使用人臉辨識來幫助判斷字幕的語者。人臉辨識分成五個步驟，依序是人臉偵測、人臉對齊、特徵提取、特徵聚類和人臉辨識。



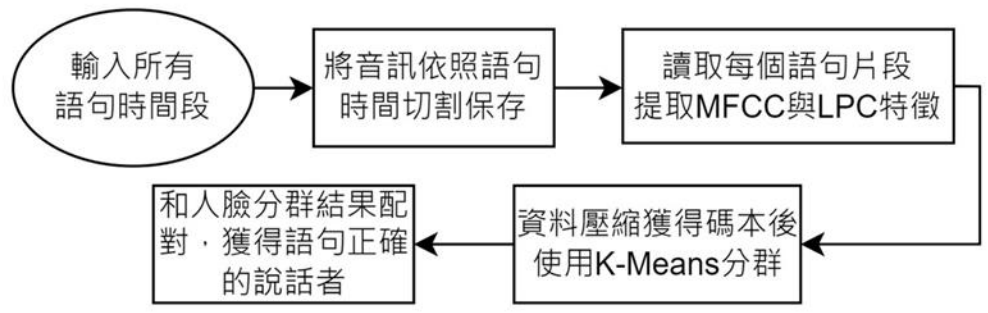
四、泡泡字幕

為了解決傳統字幕「未標記各字幕語者」以及「切換過快」的問題，我們使用泡泡字幕取代傳統字幕。利用逐幀調整畫面中泡泡字幕位置的方式使泡泡字幕能在逐漸上飄的同時跟隨各字幕的語者。



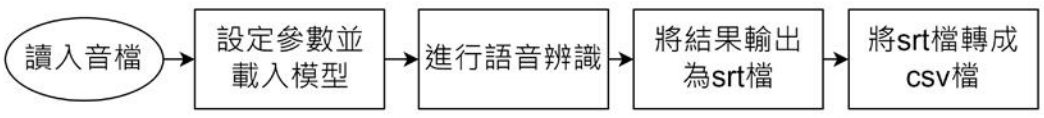
五、語者辨識

在將字幕嵌入至語者旁時 ，一旦遇到畫面中偵測到有多個人都張開嘴混淆「偵測說話者」時，就需要更精確的語者辨識來找出正確的說話者。一開始，我們需要將影片所有語句聲音片段進行聲音分群，再透過語句分類與畫面中人臉說話資訊配對，如此一來便可完成音訊與人臉的對照。



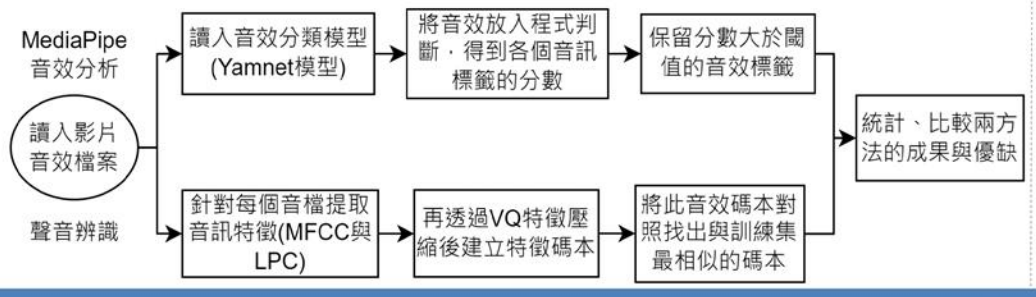
六、語音辨識

將目標音檔讀入後，程式會依據設定的模型和語言進行語音辨識，輸出標示編號、起始時間、結束時間和相對應字幕的 srt 檔案。為了讓後續程式方便取用資料，再將生成的 srt 檔轉換為 csv 檔。



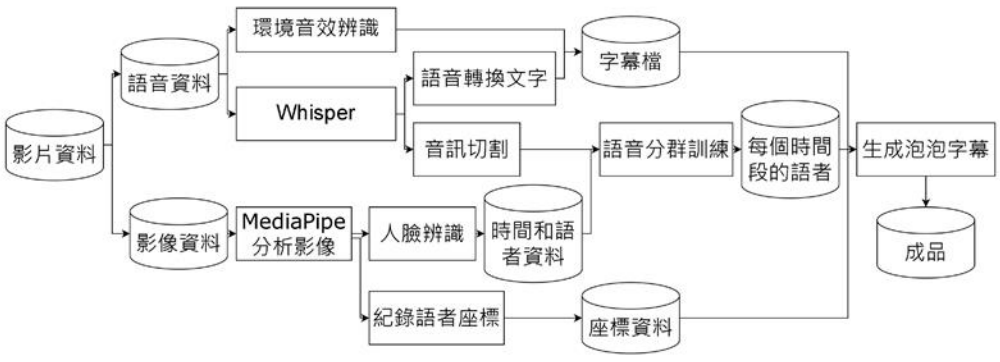
七、環境音效辨識

為了將影片中的聽覺訊息轉換成視覺化提示字卡，我們需要針對影片的聲音進行音訊分析與分類，我們使用了聲音辨識來嘗試不同音訊類型的辨識與MediaPipe音訊分析工具來補足可辨識音效的數量。



八、統整自動化程式

有了以上各部分的成果，透過Python將其統整成一個完整程式。此程式可自動將輸入影片處理成適合聽障人士觀賞的結果影片。



研究結果

一、素材來源與編號

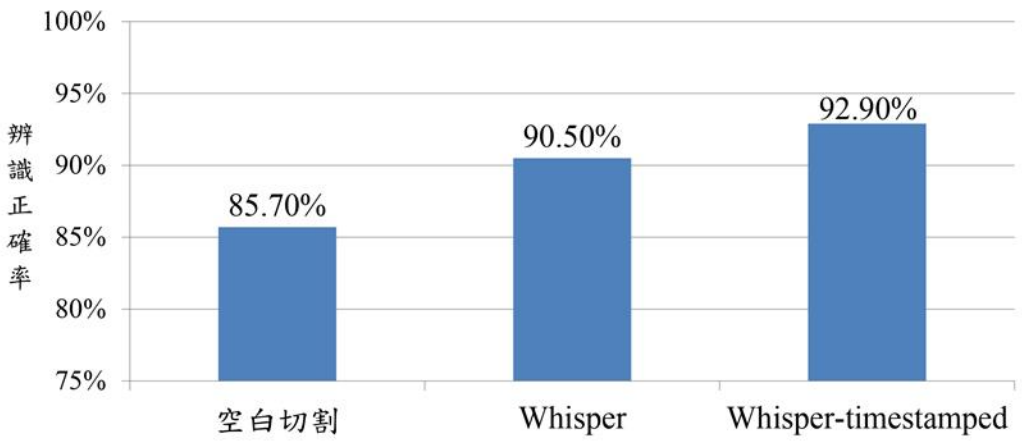
編號	來源	影片人數與類型
A	YouTube	單人(演講)
B	YouTube	雙人(談話)
C	YouTube	三人(訪談)
D	Netflix	雙人(英文訪談)
E	Netflix	三人(環境音效)

二、人臉辨識的效果

素材編號	影片人數與類型	分群數量	人臉辨識正確率
B	雙人(談話)	2	100%
C	三人(訪談)	4	98.7%
D	雙人(英文訪談)	2	100%

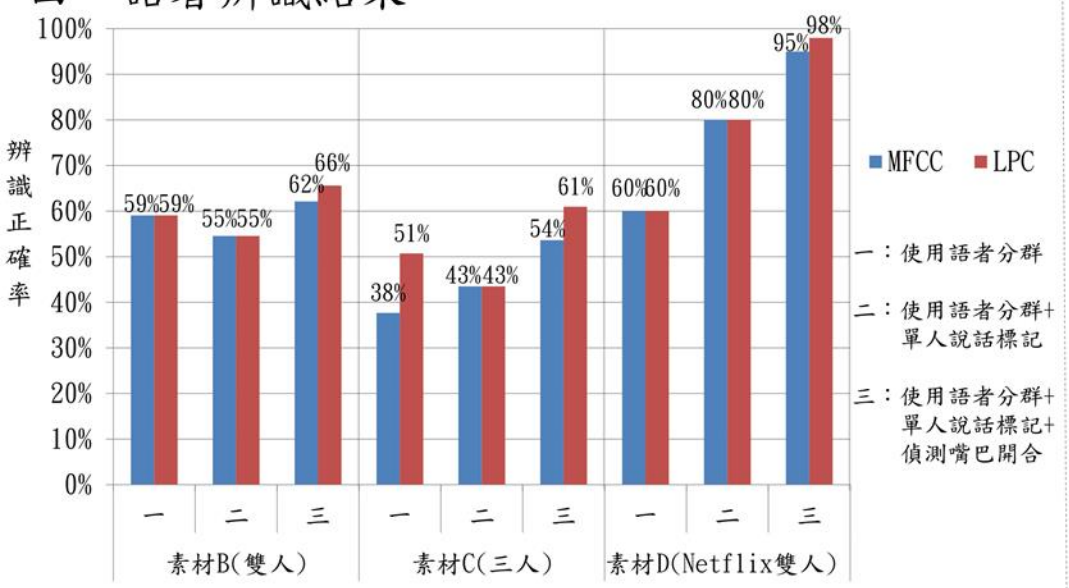
由上表可知各素材的人臉辨識正確率都相當高，達到98.7%以上。

三、語句切割的效果



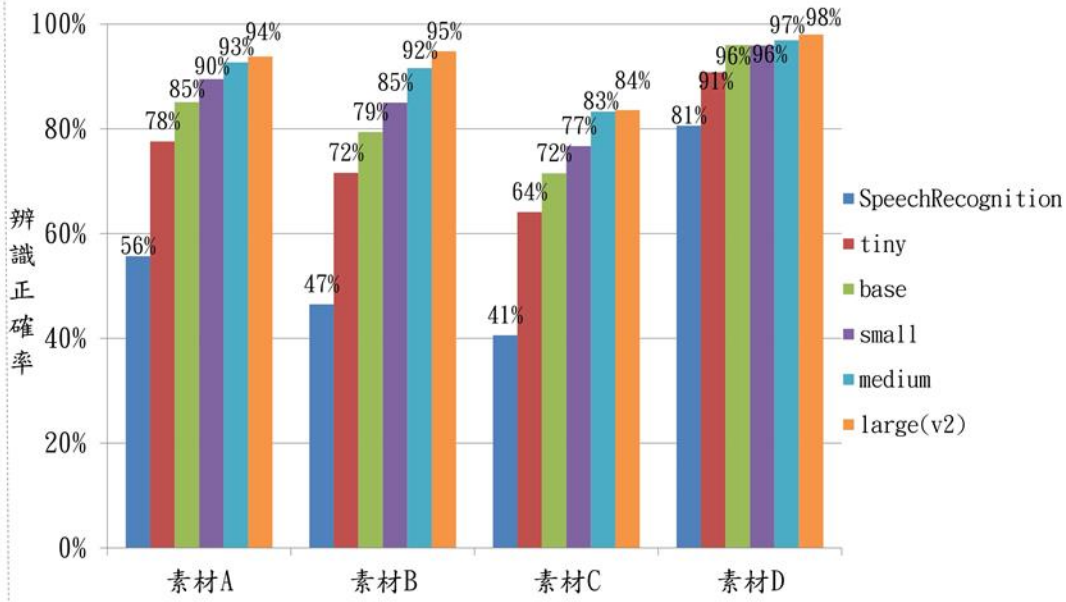
由上圖可知，使用Whisper-timestamped生成的時間戳記確實較為準確，比起其他組別有較多完整語句。空白切割的正確率較其餘兩者低，推測是影片中說話時節奏較快，語句中沒有明顯的靜默時段。

四、語者辨識結果



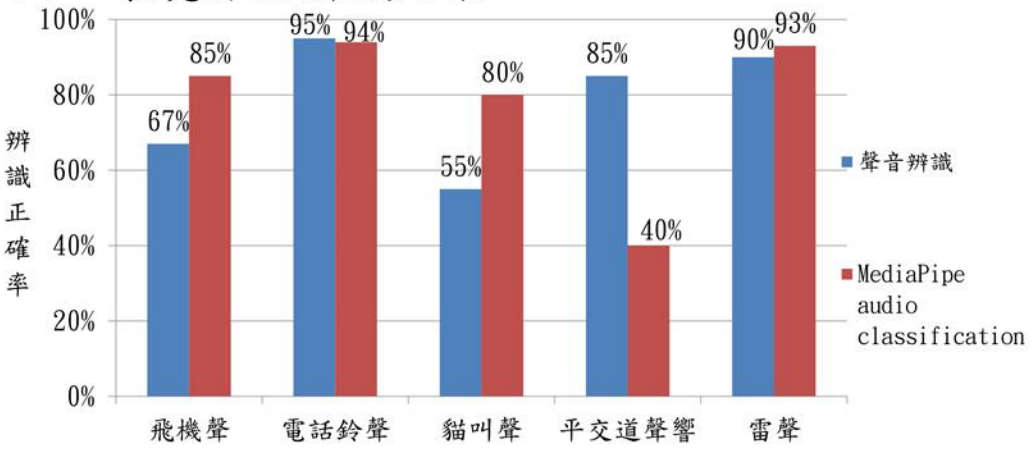
透過上圖發現，大部分素材兩種聲音特徵效果都差異不大，而隨著語者辨識標記方式增加，正確率都會有一些成長，但是還是會因為素材類型差異而有不同的效果，像是素材B跟素材C大部分都是有多人同時在鏡頭內，很容易混淆嘴巴張合偵測導致準確度不好，而素材D則有許多的單人畫面使字幕可以被精準判斷為正確的語者。

五、語音辨識結果



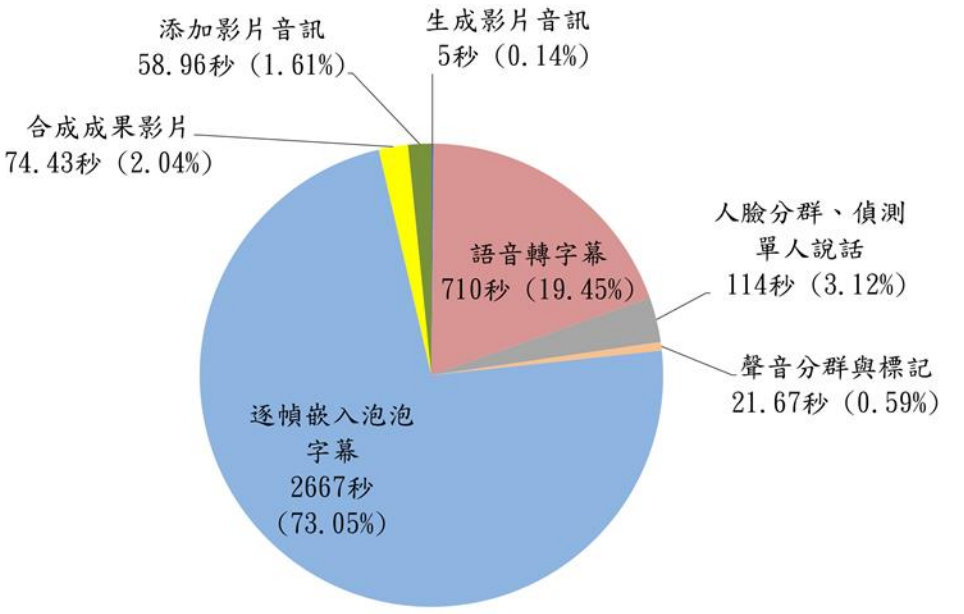
上圖中我們可以發現每個素材隨著模型尺寸放大，正確率的提升都會逐漸趨緩。此外，素材D的任何模型正確率皆在90%以上，推測是因為英文有較多訓練資料，因此成效較好。

六、環境音效辨識結果



上圖中我們可以發現大部分音效MediaPipe音效辨識工具的效果都比我們自行設計訓練的聲音辨識好，唯獨平交道聲響這項的正確率較低，我們推測是因為實驗使用的聲音分類器模型中較少平交道聲音的訓練資料，以至於遇到該音效時無法準確辨識。

七、影片處理效率



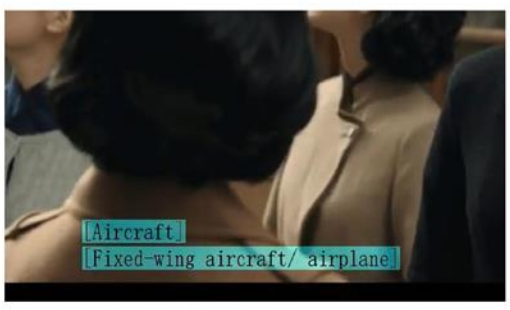
上圖顯示素材B兩分鐘的影片生成最花時間的是「逐幀嵌入泡泡字幕」與「語音轉字幕」這兩個步驟，若要改善語音轉字幕效率可以改採用辨識效果與large模型相差不大但耗時更少的medium模型，可減少為原步驟二分之一的耗時；逐幀嵌入泡泡字幕則可以添加場景轉換偵測，就不用每幀偵測人臉，進而提高效率。

八、自動化處理影片成果：

上圖為單人畫面下的泡泡字幕效果，文字會上飄停留較久時間，使觀影者有更充足的時間閱讀字幕理解劇情。

上圖為雙人同框下泡泡字幕的畫面效果，語句也會透過語者辨識的結果標記在正確的語者臉旁。

上圖為三人同框下泡泡字幕的畫面效果，同樣是結合語者辨識的結果使泡泡字幕可以標記在正確的語者臉旁。



上圖為我們的環境音效提示字卡效果，音效辨識可以達到與Netflix描述性字幕相同的效果。

回饋與結論

一、聽障受試者回饋：

為了瞭解嵌入情境化字幕的影片是否能夠真正地改善聽障人士的觀影體驗，我們實地採訪了聽障協進會的理事長。理事長本身是輕度聽障人士，並接觸過各類型的聽障人士，包括中、重度的聽障人士。以下是我們統整理事長在觀看結果影片後所給出的回饋：

（一）情境化字幕的優點：

1、逐幀上飄的泡泡字幕能避免傳統字幕切換過快的問題
2、能得知各個字幕的語者
3、環境音效字卡能幫助理解劇情

（二）情境化字幕的缺點：

1、泡泡字幕的出現位置不固定，影響辨識
2、泡泡字幕跟隨人物移動會造成閱讀的困難
3、缺乏箭頭等標示物標示語者，影響判斷字幕語者
4、習慣傳統字幕呈現方式，需重新適應新的情境化字幕

（三）建議的改進方向：

1、不同語者的字幕可用不同顏色呈現
2、固定字幕的出現位置，維持畫面穩定
3、語者辨識需更精準
4、可將整部影片放慢，增加閱讀時間

最後理事長也提到，此系統對於聽障人士，尤其是中、重度聽障人士有非常大的助益。若再對系統進行優化和改良，在未來必定能成為聽障人士不可或缺的觀影輔助工具。

二、結論：

(一) 本研究實作出的系統能夠自動將影片進行處理，生成跟隨語者移動的泡泡字幕以及環境音效字幕，消除聽障人士在觀影時遭遇的不平等。

(二) 使用MediaPipe Face Mesh和Dlib工具以及K-Means演算法實現人臉辨識，達成98.7%以上的辨識正確率。

(三) 使用Whisper-timestamped進行語句切割，最終正確率可達90%以上。

(四) 研究中我們透過了語者聲音分群加上單人說話的標記以及判斷嘴巴開合等方式辨識語者，效果隨著不同的影片類型有著 60%-90%左右的辨識正確率。

(五) 使用Whisper模型進行語音辨識，影片的字幕正確率達到90%以上。

(六) 我們發現在音效標籤種類多樣性與描述音效精準度上MediaPipe音效分類工具效果較好，不過在增加辨識特定音效與調整判斷音效的分數閾值上，聲音辨識演算法的靈活度較高。

(七) 本研究實作出的系統達成研究預期的目標，並且跟商業化Netflix影片的字幕相比可以做到相同的水準。

(八) 本系統可以為聽障人士在觀看影片時帶來一定程度上的幫助和便利，特別是重度聽障的聾啞人士。

未來展望

一、加入戴口罩人臉辨識，使系統能處理疫情下人物戴口罩的情況
二、提升語者辨識精準度，使字幕標記正確率提升
三、環境音效程式整合，使其融入自動化Python程式中
四、提升影片處理效率，使系統的實用性提升
五、提供實時的自適應字幕標記回饋，並結合AR技術，使系統能即時展現處理結果

參考資料

[1]、Peng, Y. H. (2018, April). SpeechBubbles: Enhancing Captioning Experiences for Deaf and Hard-of-Hearing People in Group Conversations.
[2]、OpenAI Whisper: Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. ArXiv Preprint ArXiv:2212.04356.
[3]、whisper-timestamped: Louradour, J. (2023). whisper-timestamped. GitHub Repository. from https://github.com/linto-ai/whisper-timestamped
[4]、Das, O. SPEAKER RECOGNITION. from https://ccrma.stanford.edu/~orchi/Documents/speaker_recognition_report.pdf