

第二十三屆旺宏科學獎

成果報告書

參賽編號：SA23-044

隊伍名稱：代碼十字軍

作品名稱：跨模態聲音訊號分析：肺音音頻視覺化與深度學習的應用研究

參賽類別：資訊

關鍵字：遠程醫療、肺音分析、卷積神經網路、視覺化技術

摘要

COVID-19 疫情對全球醫療系統造成了巨大衝擊，凸顯了遠程醫療技術的重要性。然而，遠距離肺部聽診的無法實行對聲音品質和診斷準確性提出了重大挑戰。為了解決這一問題，本研究提出了一種創新的遠程肺音分析系統。該系統利用廣泛可獲取的肺音數據集，並應用先進的人工智慧演算法進行分析。首先，通過快速傅立葉變換（FFT）和頻譜分析技術，系統能夠有效提取肺音的關鍵時頻特徵。為了便於直觀理解這些特徵，將其視覺化為波形圖、梅爾頻率倒譜係數（MFCCs）和梅爾頻譜圖。進一步地，引入卷積神經網路（CNN）模型，將視覺化數據轉化為圖像。通過 CNN 模型的自動特徵提取和分類能力，該系統能夠準確識別肺部聲音中的關鍵特徵，實現高效的疾病分類。此外，研究還探討了 Gammatone 頻率倒譜係數（GFCCs）和線性頻率倒譜係數（LFCCs）的應用，證明這些技術在肺音分析中的有效性和潛力。

關鍵詞：遠程醫療、肺音分析、卷積神經網路、視覺化技術

目錄

摘要.....	1
目錄.....	2
第一章 研究動機.....	3
第二章 研究目的.....	4
第三章 研究過程.....	5
第一節：建立研究架構與繪製流程圖.....	5
第二節：資料集獲取收集與整理.....	6
第三節：文獻探討.....	10
第四節：研究設備及器材.....	14
第五節：實驗流程.....	15
第四章 研究結果.....	21
第二節：指標分析：.....	23
第五章 討論及應用.....	25
第一節：MFCCs 和 Mel 頻譜的敏感度改善.....	25
第二節：MFCCs 變體-GFCCs 和 LFCCs 探討.....	25
第三節：當前應用.....	26
第四節：未來展望.....	27
第六章 參考文獻.....	29
附錄.....	31

第一章 研究動機

自 2020 年初爆發以來，新型冠狀病毒肺炎（COVID-19）疫情對全球造成了前所未有的衝擊，徹底改變了人們的生活方式。為降低病毒傳播風險，許多國家實施了嚴格的社交距離措施，促使遠距工作、線上教學和遠距醫療等新興服務模式蓬勃發展。在醫療領域，傳統的面對面診療模式受到嚴重限制，迫使醫療機構加速數位轉型，探索視訊問診、遠距監測等創新服務。臺灣衛生福利部於 2021 年疫情警戒提升至三級時，大力推動醫療機構開展視訊門診。這使得病情穩定的患者能透過電話完成問診、檢查和領藥，減少感染風險。醫師運用視訊和線上平臺等技術，在不直接接觸患者的情況下進行診斷和治療。然而，在遠距醫療模式下，醫師無法像傳統門診那樣直接使用聽診器為患者進行肺部聽診，這對呼吸系統疾病的診斷造成了重大挑戰。事實上，聽診在醫學史上一直扮演著關鍵角色。在聽診器發明之前，醫生使用的是傳統直接聽診法，即用手巾或布料貼在患者身上「聽診」，但這種方法不僅局限性大，也有衛生上的疑慮。1816 年，法國醫生勒內·泰奧菲勒·亞森特·拉埃內克（法語：René-Théophile-Hyacinthe Laennec）的發明徹底改變了這一情況。他發明的聽診器（如圖 1-1 所示）不僅革新了胸腔疾病的診斷方式，更大幅提升了診斷的準確性和有效性。由於這項重大貢獻，拉埃內克醫生被後世尊稱為「胸腔醫學之父」，他的發明對現代醫學的發展產生了深遠的影響。



圖 1-1：聽診器發明前後聽診方式

圖片來源：左圖 Weibo.com/u/1291471320；

右圖 Sakula, A. R. T. H. (1981). RTH Laennec 1781--1826 his life and work: a bicentenary appreciation. *Thorax*, 36(2), 81-90.

肺音是評估呼吸系統健康狀況的重要生理指標，對於診斷肺炎、氣喘等常見

疾病具有關鍵作用。在傳統臨床實踐中，醫師透過聽診器仔細聆聽患者的肺部音，以判斷是否存在異常，進而了解病情嚴重程度。然而，在遠程醫療的背景下，如何獲取和分析患者的肺音資訊成為了一大挑戰。行動通訊和物聯網技術的突飛猛進，讓智慧型手機和可穿戴設備的普及率節節攀升。根據國家通訊傳播委員會的統計，截至 2024 年 1 月，台灣的行動寬頻用戶已超過 2,980 萬人。這意味著絕大多數民眾已具備使用智慧型裝置的基礎，為遠程醫療服務的推廣提供了有利條件。在此背景下，開發一種基於智慧聽診器和人工智慧技術的遠程肺音分析系統，對於解決遠程醫療中肺部聽診的難題，提升呼吸系統疾病診療水準，具有重要意義。這也是本研究的主要動機。

第二章 研究目的

本研究的主要目的是開發一種基於智慧聽診器和人工智慧技術的遠程肺音分析系統，以期解決遠程醫療中肺部聽診的難題，提升呼吸系統疾病的診療水準。具體目的如下：

- (一)開發肺音圖譜轉換演算法：設計並實現將聽診器採集的肺音訊號轉換為視覺化圖譜（如波形圖、頻譜圖等）的演算法，提取能夠反映肺音特徵的關鍵資訊。這一步驟將為後續的智慧分類模型提供重要的輸入資料。
- (二)構建肺音智慧分類模型：利用深度學習技術，使用卷積神經網路（CNN），訓練能夠自動識別和分類正常/異常肺音的智慧模型，並優化其性能指標。這一模型將成為遠程肺音分析系統的核心組件，實現對肺音的快速、準確判讀。
- (三)開發遠程肺音分析系統雛型：整合智慧聽診器硬體、肺音圖譜轉換演算法和智慧分類模型，搭建一個用於遠程肺音採集、傳輸、分析和診斷的雲端系統雛型。這一系統將為醫療人員提供遠程肺音診療解決方案。

我們希望能夠突破傳統面對面聽診的局限，為醫療人員提供更智慧、更高效的遠程診療工具，切實提升呼吸系統疾病的診療品質和可及性。同時，本研究也將為人工智慧技術在智慧醫療領域的創新應用提供有益的探索和實踐。

第三章 研究過程

第一節：建立研究架構與繪製流程圖

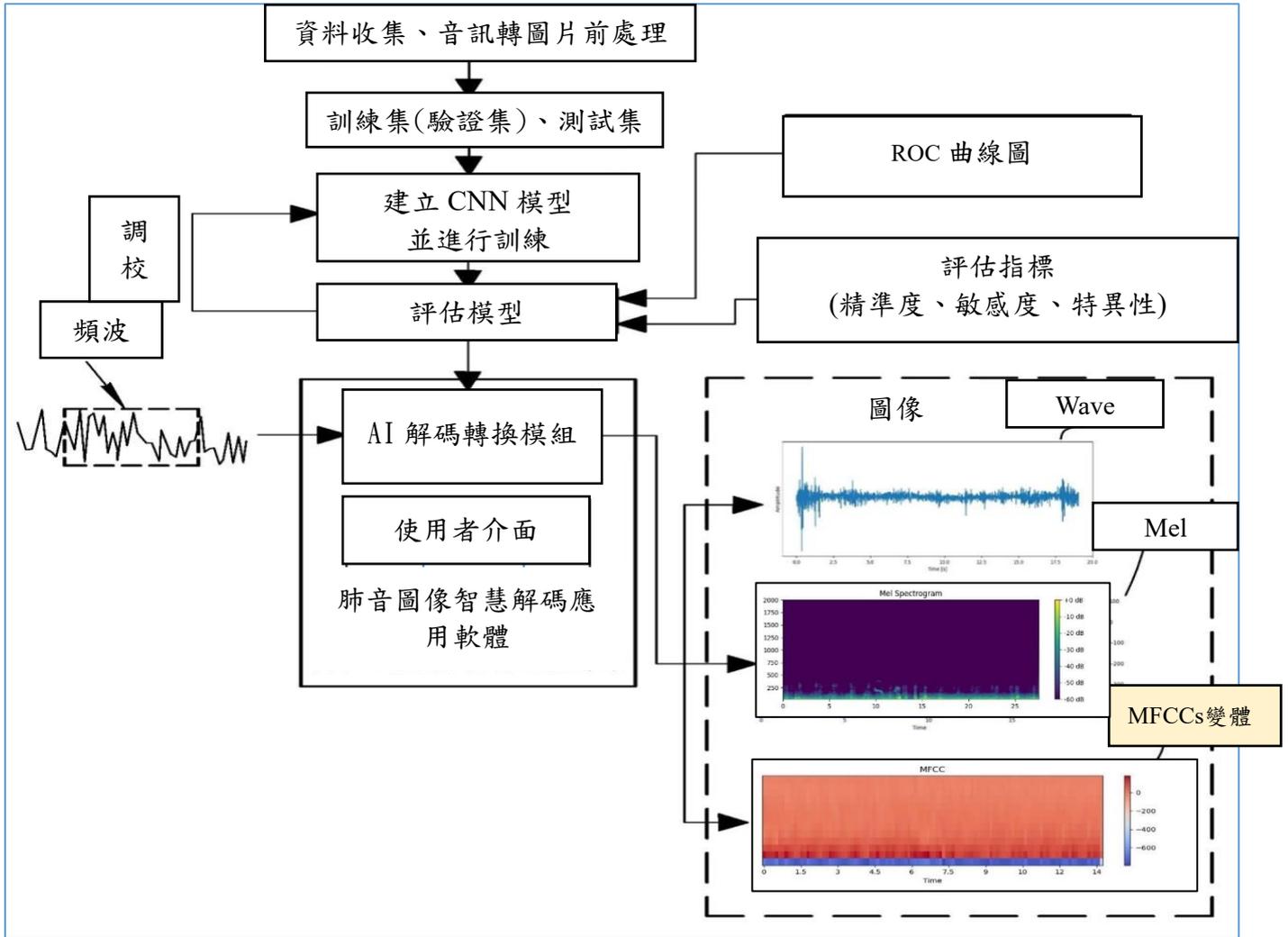


圖 3-1：研究架構圖

圖片來源：本研究整理繪製產出

根據研究架構圖，本研究的整體流程可以總結如下：

- (一)肺音採集：收集健康以及不同疾病患者錄製肺部音頻。
- (二)訊號前處理：對採集到的原始肺音進行必要的預處理，包括降噪、去除雜訊等以提高音訊品質。
- (三)特徵提取、圖像轉換：運用快速傅立葉變換(FFT)等方法，從預處理後的肺音訊號中提取能夠反映其特徵的關鍵資訊，將提取的特徵轉換為視覺化的波形圖(Wave)、梅爾頻譜圖(Mel-spectrogram)、梅爾頻率倒譜係數 (MFCCs)、Gammatone 頻率倒譜係數(GFCCs) 和線性頻率倒譜係數 (LFCCs) 五種圖形

為分類提供直觀的輸入。

(四)CNN 模型訓練：使用轉換後的圖像數據訓練卷積神經網路模型，使其能夠自動識別和分類正常/七種異常的肺音。這一步是整個分析的核心。

(五)ROC 曲線圖：基於 CNN 模型的分類結果，繪製 ROC 曲線圖評估模型性能。

(六)評估指標：計算準確率、精準度、敏感性三項指標，衡量模型的分類效果。

(七)輔助醫師診斷：將訓練好的 CNN 模型整合到雲端的遠程肺音分析系統中，為醫師提供輔助診斷的功能，幫助其做出更加準確、高效的臨床決策。

第二節：資料集獲取收集與整理

(一)資料集來源與驗證：

本研究所使用的肺部音頻資料集是通過多方管道獲取和驗證的。首先，我們在知名的資料科學社區和競賽平台 Kaggle 上找到了一個名為 "Lung Sound Database" 的資料集，其中包含了大量肺部音頻資料。為了進一步確認該資料集的可靠性和來源，我們對其進行了詳細的追蹤和核實。經過調查，我們確認該資料集源自 Fraiwan, Fraiwan, Khassawneh, & Ibnian 在 2021 年發表的論文 "A dataset of lung sounds recorded from the chest wall using an electronic stethoscope"。為了推動該領域的研究進展，本篇論文的作者們已將他們的資料集公開，並透過 Mendeley Data 平台進行分享。Mendeley Data 作為一個集中化的資源庫，便於研究人員發掘、存取及共同利用這些數據。此舉使得更多研究者得以發現並應用這些資料，進一步促進了學術交流和合作的深化。

(二)資料集分布：

本研究使用的資料集包括 112 名受試者，年齡範圍從 21 至 90 歲，平均年齡為 50.5 歲(標準差 19.4 歲)，男女比例為 69:43。其中 35 名(31.3%)為健康人士，77 名(68.8%)患有各類呼吸道疾病(如表 3-1)。

(三)資料集使用與規範

在使用該資料集進行研究的過程中，我們嚴格遵循了資料集的使用條款和來源出處的標註要求。完整的資料集可以通過以下連結直接從 Mendeley Data 平台獲取：[Mendeley Data](#)。資料集涵蓋健康及 7 種疾病狀態（氣喘、肺炎、慢性阻塞

性肺病、支氣管炎、心臟衰竭、肺纖維化、胸腔積水)，包括 112 名受試者的音頻樣本。每個樣本都用三種過濾器錄製，總音頻數達 336 個。所有音頻均由專科醫師在確認肺音聽取位置後錄製，確保資料的可靠性和專業性。受試者的年齡和性別資訊包含在音頻檔名中，為後續分析提供背景資訊。

表 3-1：受試者分布資料表

類別	組別	人數	百分比
年齡	21-30 歲	18	16.07%
	31-40 歲	22	19.64%
	41-50 歲	19	16.96%
	51-60 歲	20	17.86%
	61-70 歲	14	12.50%
	71-80 歲	12	10.71%
	81-90 歲	7	6.25%
性別	女性	43	38.39%
	男性	69	61.61%
健康狀況	健康人士	35	31.25%
	呼吸道疾病	77	68.75%
總計		112	100.00%

資料來源：整理自 Fraiwan et al. (2021)

(四)音頻資料詳述：

聽診器的鐘面 (B-mode, Bell) 和膜面 (D-mode, Diaphragm) 在聲音傳導方面具有獨特的特性。B-mode 專為低頻聲音設計，其圓形且略微凹陷的形狀和材質能有效接收低頻率聲波，適合聽診如心臟的低音雜音。相對而言，D-mode 則專為高頻聲音設計，其堅固的材料結構適合接收高頻聲波，如心臟瓣膜閉鎖的聲音或肺部的呼吸聲。根據 Nowak 與 Nowak (2017) 的研究，D-mode 會衰減低頻聲音，主要傳導高頻聲音，而 B-mode 則能均衡地傳導各種頻率的聲音，無需進行選擇性過濾。此外，電子聽診器 (E-Scope) 利用數位技術增強聽診效果，透過提高聲音的清晰度和放大，改善臨床診斷的精確性。原始資料中三種過濾器模式的強調頻率範圍如下 (見圖 3-2)

1. B 模式(B- mode)過濾器：

- (1) 整體放大範圍：20 Hz 至 1000 Hz。
- (2) 特別強調範圍：20 Hz 至 200 Hz 的低頻聲音。

2. D 模式(D-mode)過濾器：

(1) 整體放大範圍：20 Hz 至 2000 Hz。

(2) 特別強調範圍：100 Hz 至 500 Hz。

3. E 模式(E-Scope)過濾器：

(1) 整體放大範圍：20 Hz 至 1000 Hz。

(2) 特別強調範圍：50 Hz 至 500 Hz。

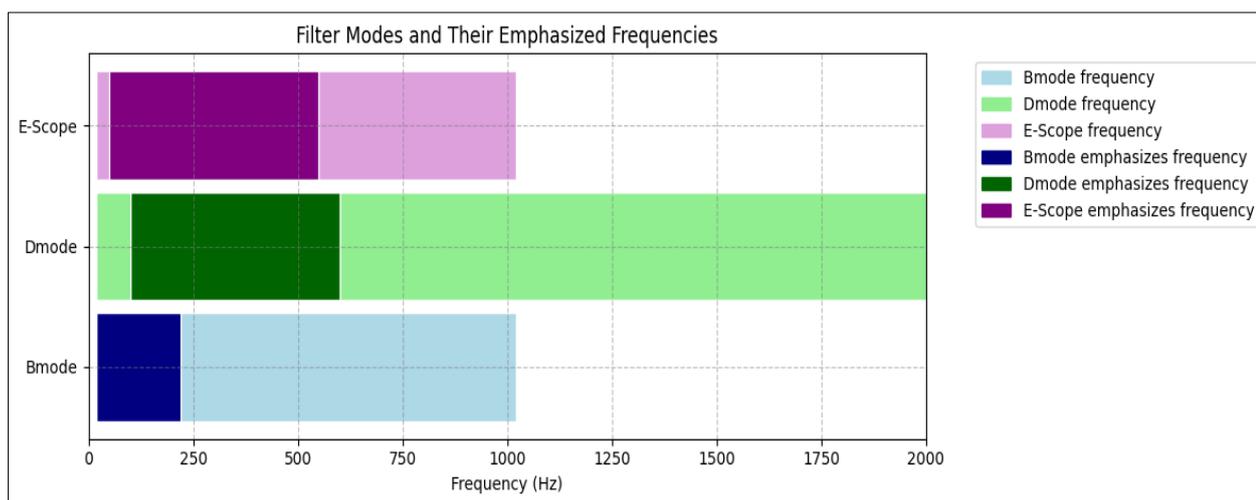


圖 3-2：音頻過濾器頻率分析

圖片來源：本研究程式整理繪製產出

經過對這三種過濾器的頻率範圍（見附錄一）和持續時間（見附錄二）進行詳細分析，我們發現：

1. 頻率範圍分析：

(1) 三種模式的最高頻率範圍均為 0 Hz 至 1999.97 Hz。

(2) 平均頻率和中位數頻率均接近 999.97 Hz。

2. 錄音時長分析：

(1) 大多數情況下，B、D 和 E 三種模式的錄音時長相同，顯示穩定性良好。

這表明在大部分病例中，三種聽診器模式能夠在相似的時間內捕捉到足夠的診斷資訊，反映了它們在臨床應用中的一致性和可靠性。

(2) 錄音時長範圍:在整個資料集中，錄音時長從最短的 5 秒(如 BP22_BRON)到最長的 30 秒(如 BP1_Asthma、BP2_Asthma 等)不等。

不同病理狀況的錄音時長反映了病理狀況的複雜性，例如氣喘、心衰竭和肺纖維化的錄音時長相對較長，顯示出這些狀況可能需要更長的時間來捕捉到關鍵

的診斷信息。

(五)音頻檔案命名規則：

音頻檔案的命名方式非常重要，因為它包含了許多關鍵的臨床資訊。以下是一個音頻檔案名稱的示例：B_P120_Asthma_Wheezing_PRUL_56_M.wav 從這個檔名可以提取出以下資訊：

1. 模式：B，鐘面 (B-mode, Bell)
2. 病人編號：P120
3. 診斷結果：氣喘 (Asthma)
4. 聲音類型：哮鳴音 (Wheezing)
5. 胸部測量位置：PRUL (胸部後側，右上部)(整理如表 3-2)
6. 年齡：56 歲
7. 性別：男性 (M)

通過這種命名方式，研究者可以方便地根據不同的臨床特徵對音頻資料進行分類和分析。例如，可以比較不同疾病狀態下的肺音特點，或者探究不同胸部位置的聲音差異。同時，年齡和性別資訊也為研究提供了重要的人口統計學背景。

表 3-2：胸部位置的標記

位置	字母	中文	英文全名
胸部方位	A	前部	Anterior
	P	後部	Posterior
胸部側邊	L	左側	Left
	R	右側	Right
胸部高度	L	下部	Lower
	U	上部	Upper
	M	中部	Middle

資料來源：整理自 Fraiwan et al. (2021)

(六)資料標註與分析：

為深入探索肺音與呼吸道疾病之間的關聯，本研究對資料集中的標註資訊

進行了詳細分析。除音頻檔案外，資料集還包括相應的 Excel 標註文件，提供每個音頻樣本的診斷結果和肺音類型等關鍵資訊。基於這些標註，我們統計了各種診斷條件下不同類型肺音的分佈情況，如表 3-3 所示。總共收集了 112 名受試者的資料，其中 35 名為健康人士，77 名患有各類呼吸道疾病，包括氣喘、肺炎、慢性阻塞性肺病 (COPD)、支氣管炎 (BRON)、心臟衰竭、肺纖維化和胸腔積液。通過表 3-3 中的分析數據，我們發現不同疾病狀態下肺音的類型和分佈存在顯著差異：

表 3-3：肺音/診斷分佈

疾病狀況	收集到的肺音	音頻數	總音頻數	總人數
正常 (Normal)	正常 (Normal)	105	105	35
氣喘 (Asthma)	呼氣時哮鳴 (Expiratory : Wheezes)	90	96	30
	吸氣與呼氣時哮鳴 (Inspiratory & Expiratory:Wheezes)	6		2
肺炎 (Pneumonia)	支氣管音 (Bronchial)	3	15	1
	捻髮音 (Crepitations)	9		3
	細爆裂音 (Crackle)	3		1
慢性阻塞性肺病 (COPD)	呼氣時哮鳴 (Expiratory:Wheezes)	27	27	9
支氣管炎 (BRON)	捻髮音 (Crepitations)	9	9	3
心臟衰竭 (Heart Failure)	捻髮音 (Crepitations)	39	63	13
	爆裂音 (Crackle)	15		5
	吸氣時爆裂音 (Inspiratory:Crackles)	6		2
	呼氣時哮鳴 (Expiratory:Wheezes)	3		1
肺纖維化 (Lung Fibrosis)	捻髮音 (Crepitations)	12	15	4
	細爆裂音 (Crackle)	3		1
胸腔積液 (Pleural Effusion)	吸氣時爆裂音、支氣管音 (Inspiratory:Crackles, Bronchial)	6	6	2

資料來源：本研究整理

第三節：文獻探討

(一)肺音(lung sounds)：

肺音(lung sounds)是指在胸壁或胸腔內與呼吸過程相關的所有聲音，其中包

含了基本的呼吸音和其他在該區域檢測到的聲音。Sarkar, Madabhavi, Niranjana, & Dogra (2015)的研究指出，呼吸音是肺音的一部分，主要由氣流在呼吸過程中通過氣道時產生。這些聲音不僅反映了呼吸系統的生理特徵，也是評估呼吸系統健康狀態的重要臨床工具。Andrès, Gass, Charloux, Brandt, & Hentzler (2018)在其研究中詳細介紹了各種異常呼吸音的聲學特徵，並分析了它們的臨床意義。這些異常呼吸音通常表現為週期性波形，其特徵與呼吸系統的健康狀態密切相關。透過對呼吸音的仔細分析和評估，可以為臨床診斷和治療提供寶貴的線索和依據。

本研究涵蓋的肺音種類如表 3-4 所示。其中，肺部聲音和氣管聲音屬於正常肺音，頻率範圍分別為 100-1,000 Hz 和 100-4,000 Hz。而細爆裂音(又稱 Fine Crackles)、喘鳴(Wheezes)和爆裂音(又稱 Coarse Crackles)則屬於異常肺音，它們與特定的呼吸道疾病密切相關。例如，細爆裂音常見於肺炎、充血性心臟衰竭和肺纖維化患者，而喘鳴則常見於氣喘、慢性阻塞性肺病和氣道異物患者。爆裂音則常見於慢性支氣管炎、支氣管擴張和 COPD 患者。

表 3-4：肺音種類表

異常	肺音類型	頻率範圍	相關疾病
	肺部聲音(Lungs)	100-1,000 Hz	-
	氣管聲音(Bronchial)	100-4,000 Hz	-
✓	細爆裂音 (Crepitations, 又稱 Fine Crackles)	650 Hz;duration	肺炎、充血性心臟衰竭、肺纖維化
✓	喘鳴 (Wheezes)	100-1,000 Hz; duration >80ms	氣喘、慢性阻塞性肺病、異物
✓	爆裂音(Crackles 又 稱 coarse crackles)	<350 Hz	慢性支氣管炎、支氣管擴張、COPD

資料來源：整理自 Andrès et al. (2018)

(二)音頻訊號的分析技術：

1. 波形圖(Waveform)是音頻信號的時域表示，其中橫軸表示時間，縱軸表示音頻信號的幅度。波形圖直觀地展示了音頻信號的變化趨勢，可用於觀察音頻的持續時間、振幅變化及聲音的節奏特徵。然而，波形圖並不能提供關於音頻信號頻率成分的信息，因此在某些應用中，如語音識別，其表現力有限 (Rabiner & Schafer, 2010)。

2. Stevens, Volkman, & Newman (1937) 首次提出了 Mel 頻譜圖的尺度概念，他們描述了人耳對頻率的非線性感知。這個理論後來被廣泛應用於聲音訊號處理領域。Mel 頻譜圖通常是透過對音頻訊號進行短時傅立葉變換 (STFT) 獲得，然後將頻率轉換至 Mel 尺度，並應用對數轉換以模擬人類的聲音感知機制 (Rabiner & Schafer, 1993)。這種轉換使得頻譜圖更符合人耳的聽覺特性，從而提升了音頻處理技術在多種應用中的表現和準確度。
3. 梅爾頻率倒譜係數 (MFCCs) 是一種基於人耳聽覺特性的音頻特徵表示方法，廣泛應用於語音識別和音頻分類任務中。MFCCs 的計算包括多個步驟：首先對音頻信號進行預加重，接著進行分幀和加窗處理，然後執行傅立葉變換，通過梅爾濾波器組過濾，計算對數能量，最後進行離散餘弦變換 (DCT)。MFCCs 通過擷取音頻訊號的頻譜包絡資訊，模擬人耳對各種頻率聲音的感知反應，使其在語音辨識領域表現出色 (Davis & Mermelstein, 1980)。

3.1 MFCCs 的計算步驟示意圖如圖 3-3

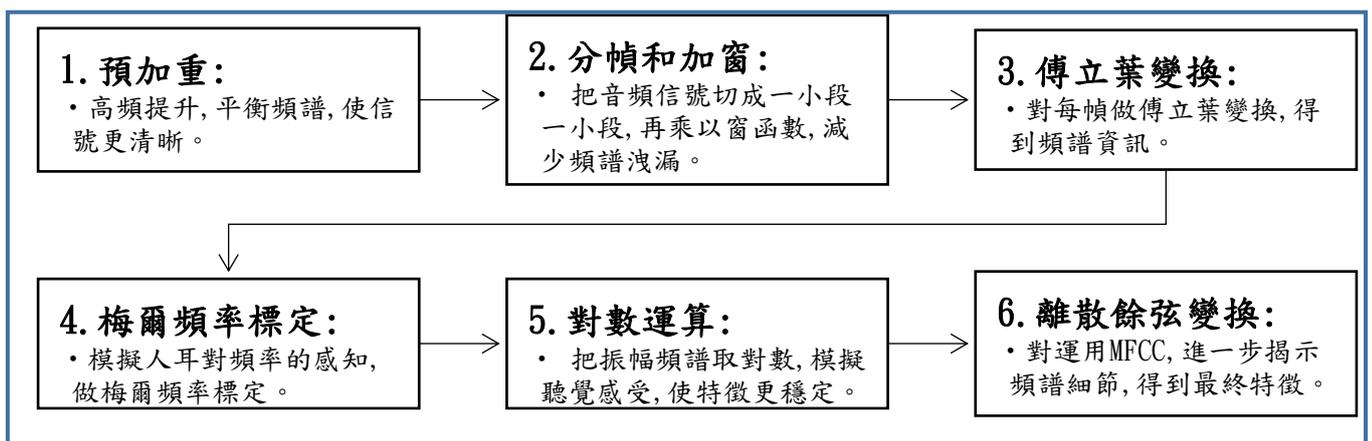


圖 3-3：MFCCs 的計算步驟

資料來源：本研究整理

4. GFCCs (Gammatone Frequency Cepstral Coefficients)：首先將語音信號進行預強調處理，以增強語音的高頻帶。接著將語音信號分為多個非重疊的連續幀，每個幀被視為靜態信號。然後，利用圖形頻譜分析 (GFT) 來提取 GFCC 特徵，這有助於更準確地表示語音樣本的結構關係 (Xu, Tian, Guo, Shan, Jia, Peng, Yang, & Das, 2022)。
5. LFCCs (Linear Frequency Cepstral Coefficients)：使用線性頻率刻度，適用於一些需要線性頻率分辨率的應用 (如語音辨識或音樂分析)。LFCC 在這些應用

中可能比傳統的 MFCC 更加合適，因為它提供了不同的頻率分辨率 (Wu, Evans, Kinnunen, Yamagishi, Alegre, & Li, 2015)。

表 3-5：MFCCs 與不同變體的比較

特徵	MFCCs	GFCCs	LFCCs
基本原理	梅爾尺度	Gammatone 濾波器	基於線性頻率尺度
頻率尺度	非線性 (梅爾尺度)	非線性 (ERB 尺度)	線性
濾波器	梅爾濾波器	Gammatone 濾波器	線性濾波器

資料來源：Davis & Mermelstein (1980) ; Xu et al. (2022); Wu et al. (2015)

表 3-6：各變體的算法步驟

變種	算法步驟
MFCCs	預加重→分幀→加窗→FFT→梅爾頻率標度→對數運算→DCT。
GFCCs	預加重→分幀→加窗→FFT→Gammatone 濾波器→對數運算→DCT→GFT (Gabor 濾波器變換)。
LFCCs	預加重→分幀→加窗→FFT→線性頻率標度→對數運算→DCT。

資料來源：Davis & Mermelstein (1980) ; Xu et al. (2022); Wu et al. (2015)

(三) 卷積神經網路(Convolutional Neural Network, CNN)：

CNN 是為處理具有顯著空間結構特性的數據而設計的深度學習架構，特別適合於圖像、視頻和時間序列等數據的分析 (LeCun, Bottou, Bengio, & Haffner, 1998; Goodfellow, Bengio, & Courville, 2016)。CNN 主要由以下幾層組成：

1. 卷積層是 CNN 進行特徵提取的核心，它通過應用不同的卷積核(或稱卷積過濾器)來識別輸入數據中的特徵。這些卷積核可以捕捉到圖像的局部特徵，進而在整個網絡中提供有用的信息(Krizhevsky, Sutskever, & Hinton, 2012)。
2. 池化層(Pooling Layer)在減少參數數量和計算量中起著關鍵作用，有助於緩解過擬合問題並提升模型的泛化能力(Scherer, Müller, & Behnke, 2010)。通過對特徵圖進行下採樣，池化層能夠保留最重要的特徵資訊。常見的池化方法包括最大池化(取窗口內的最大值)和平均池化(取窗口內的平均值)。
3. 展平(Flatten)操作將卷積和池化層產生的高維特徵圖轉換為一維數組，以便送入後續的全連接層進行處理(LeCun et al., 1998)。
4. 全連接層(Fully Connected Layer)對前面層提取的特徵進行非線性變換和分類預測。通過將展平後的特徵向量與權重矩陣相乘，再應用激活函數，全連接層可以生成最終的預測結果(Goodfellow et al., 2016)。

CNN 的工作流程如下：輸入數據首先經過一系列卷積層提取特徵，然後通過池化層進行下採樣，再經過展平操作轉換為一維向量，最後由全連接層生成預測結果。這種結構使得 CNN 在圖像識別、語音處理、自然語言處理等領域取得了巨大成功(LeCun et al., 1998; Krizhevsky et al., 2012; Goodfellow et al., 2016)。

(四)Google Colaboratory：

Google Colaboratory(簡稱 Google Colab)提供了一種創新的計算平台，旨在促進複雜數據分析和機器學習模型的開發。Google Colab 是一個基於雲端的 Jupyter 筆記本環境,它允許用戶在無需進行任何配置的情況下，即可開始使用。Bisong(2019)指出,這樣的設計大大降低了進行高級數據分析的門檻，使得學術研究和教育領域的用戶能夠輕鬆地存取包括 GPU 和 TPU 在內的 Google 的雲端計算資源。這對於需要大量計算資源進行數據分析和模型訓練的機器學習和深度學習項目尤其有益，因為它顯著加速了模型的訓練過程。Bisong(2019)提出了 Google Colab 的幾個關鍵優勢，包括免費訪問先進的計算資源、無需繁瑣設置的即時計算環境，以及與 Google Cloud Platform 的緊密整合。這為從事機器學習和深度學習研究的學者和開發者提供了一個強大的工具，特別是對於希望利用雲計算加速其研究的人來說。此外，Google Colab 的環境支持協作，使得團隊成員能夠共享和審閱筆記本這進一步提高了研究和開發的效率。這一點在現代研究和開發中尤為重要，因為協作已成為科學探索和創新的關鍵驅動力(Wang, Zhong, & Li, 2020)。

除了 Bisong (2019) 的研究外，其他學者也強調了 Google Colab 在機器學習和深度學習領域的應用。例如，Carneiro, Nóbrega, Nepomuceno, Bian, Albuquerque 和 Rebouças Filho (2018) 展示了如何使用 Google Colab 進行神經網絡的訓練和部署。而 Lau, Sim, Tan 和 Low (2020) 則探討了 Google Colab 在自然語言處理任務中的應用。這些研究進一步凸顯了 Google Colab 作為一個強大的數據分析和模型開發平台的重要性。

第四節：研究設備及器材

(一)硬體與軟體設備整理如表 3-7。

表 3-7：硬體與軟體設備

硬體設備	
1.主機	HP EliteDesk 800 G4 TWR
2.CPU	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
3.GPU	NVIDIA GeForce GTX 1660 Ti
4.顯示器	DELL P2719H
軟體	
1.Google Colaboratory(NVIDIA- A100)	
2.Anaconda	
3.Visual Studio Code	
4.Python	
5.GPT-4	

資料來源：本研究整理

(二)套件

研究過程使用套件整理如表 3-8。

表 3-8：套件

1.	TensorFlow：深度學習框架
2.	Keras：深度學習框架，高級 API
3.	NumPy：數學運算庫
4.	Pandas：數據分析庫
5.	Matplotlib：數據視覺化庫
6.	Seaborn：數據視覺化庫，基於 Matplotlib
7.	OpenCV (cv2)：計算機視覺和圖像處理庫
8.	Librosa：音頻處理庫
9.	Scipy：科學計算庫
10.	Scikit-learn：機器學習庫
11.	Imbalanced-learn (imblearn)：處理不平衡數據的庫
12.	ntpath：處理路徑操作的庫
13.	pathlib：處理文件路徑的庫
14.	collections：高效數據結構
15.	random：隨機數生成
16.	os：操作系統相關功能
17.	pyinstaller：打包 Python 應用程序為可執行文件 (.whl 文件)

資料來源：本研究整理

第五節：實驗流程

先前的研究中，我們提出了一個卷積神經網絡 (CNN) 模型，並針對聲音數據的特徵進行了分類。初步結果顯示，模型在處理不同特徵 (如梅爾頻率倒譜係

數 (MFCCs) 和梅爾頻譜 (Mel-spectrogram)) 時具有不同的性能表現 (表 3-9: 原始結果回顧)。然而, 敏感度方面仍有提升空間。本研究旨在通過一系列優化措施進一步提升模型的敏感度。為了進一步提高模型的敏感度和整體性能, 加入了 Gammatone 頻率倒譜係數 (GFCCs) 和線性頻率倒譜係數 (LFCCs) 作為額外的特徵, 並對它們的效果進行了對比和分析。

表 3-9: 原始結果回顧

分類	準確度 (Accuracy)	敏感性 (Sensitivity)	特異性 (Specificity)
WAVE	98.88%	97.79%	99.27%
MFCCs	92.35%	62.53%	95.89%
Mel	99.82%	71.67%	99.73%

資料來源: 本研究整理

聲音是一種隨時間變化的信號, 通常以時間域的形式錄製。然而, 直接分析時域數據並不容易辨識聲音的主要成分或特徵。為了更好地理解聲音的組成, 我們利用快速傅立葉變換 (Fast Fourier Transform, FFT) 將時域數據轉換成頻率域的表示形式。頻率域的數據揭示了聲音的頻率成分, 但失去了時間順序的信息。為了同時捕捉聲音的頻率和時間特徵, 我們引入了頻譜圖的概念。生成過程如下:

(一) 生成頻譜圖:

首先在聲音信號上使用一個小的時間窗口進行滑動, 每次移動一小段距離。對每個時間窗口內的數據片段應用 FFT 獲得其頻域表示, 從而形成一維的頻率分佈。然後, 將這些隨時間變化的頻域數據按時間順序排列, 形成二維的頻譜圖。頻譜圖中橫軸表示時間, 縱軸表示頻率, 每個像素的亮度或顏色代表對應時間和頻率下的能量強度。

(二) 數據讀取與正規化:

在本研究中, 使用 Python 的 `scipy.io.wavfile` 模組讀取 `.wav` 格式的音頻檔案, 提取採樣率、音訊數據和總長度等基本資訊。為了方便後續的處理和分析, 我們對音訊數據進行了正規化, 將其縮放到 $[0, 1]$ 的範圍內。具體操作是將音訊數據除以其絕對值的最大值, 使所有數據點都落在 $[0, 1]$ 區間內。這種正規化操作不僅有助於提高數值計算的穩定性, 也使不同音頻檔案之間的幅度大小具有可比性。正規化後的音訊數據被用於繪製圖像, 以直觀地展示聲音信號隨時間的變化情況

(三)特徵計算與可視化

1. 繪製波形圖 (Waveform)：展示音頻在不同時間點的振幅變化，這有助於我們直觀地理解音頻波形的特性。

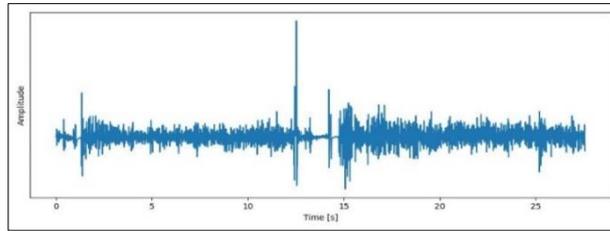


圖 3-4：Asthma 在波形圖圖示
圖片來源：本研究程式繪製產出

2. 計算並繪製 Mel 頻譜圖：利用 Mel 刻度來模擬人耳對聲音的感知方式，這使我們能夠更精確地分析和理解音訊的頻率特性。在繪製 Mel 頻譜圖時，我們對頻譜能量值進行對數變換和正規化，使得圖像的動態範圍更加均衡，有利於視覺分析和特徵提取。

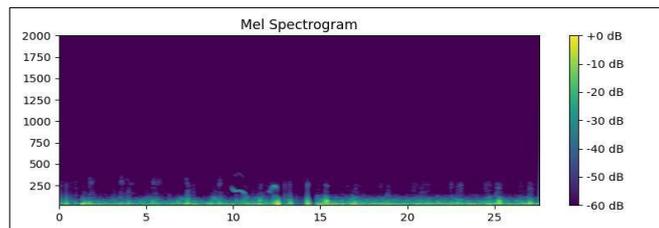


圖 3-5：Asthma 在 Mel 頻譜圖圖示
圖片來源：本研究程式繪製產出

3. 計算並繪製梅爾頻率倒譜係數 (MFCCs)：MFCCs 在語音識別和音訊處理領域中被廣泛應用，能有效捕捉音訊的頻譜特性。在計算 MFCCs 時，將梅爾頻率倒譜進行了正規化處理，使其值落在 $[0, 1]$ 區間內。

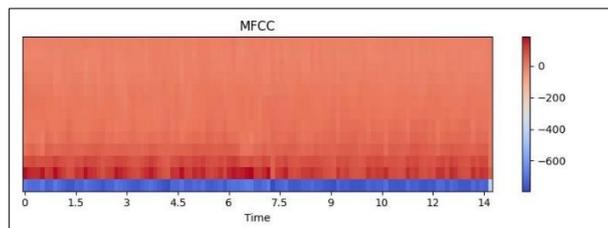


圖 3-6：Asthma 在 MFCCs 圖示
圖片來源：本研究程式繪製產出

4. 計算並繪製 Gammatone 頻率倒譜係數 (GFCCs)：GFCCs 使用 Gammatone 濾波器來模擬人耳的聽覺特性，這使得其在某些應用中比 MFCCs 更有效。具體而言，GFCCs 在噪聲環境下的語音識別、複雜音樂信號分析以及生物聲學分析中表現優異，因為它更精確地模擬了人耳的聽覺過濾特性。在計算 GFCCs 時，並同樣進行了正規化處理。

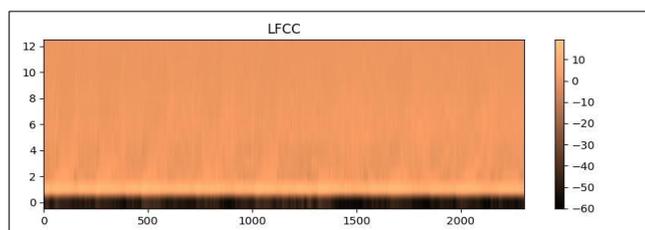


圖 3-7：Asthma 在 GFCCs 圖示
圖片來源：本研究程式繪製產出

5. 計算並繪製線性頻率倒譜係數 (LFCCs)：LFCCs 使用線性頻率刻度，適用於一些需要線性頻率分辨率的應用。在計算 LFCCs 後對其進行了正規化處理。

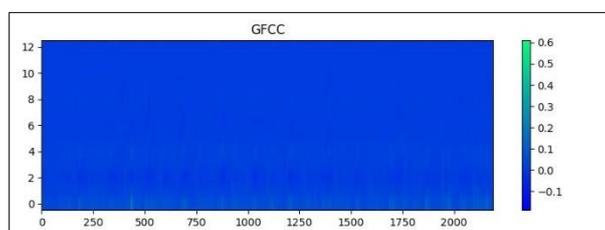


圖 3-8：Asthma 在 LFCCs 圖示
圖片來源：本研究程式繪製產出

(四) 建立與優化 Python 程式碼的 GPT-4 智能輔助系統

我們使用思維鏈 (Chain-of-Thought, CoT) 技術，透過引導模型模仿人類的推理步驟，提高其推理能力。這個方法有助於大型語言模型克服需要邏輯思維或多個步驟解決的推理任務。例如，ChatGPT 在剛推出時，其數學運算的回答正確率不佳，因為其擅長語言生成而非邏輯運算。利用思維鏈 (CoT) 技術能有效改善其邏輯運算的正確性 (Wei, Wang, Schuurmans, Bosma, Ichter, Xia, Chi, Le, & Zhou, 2022)。以下是使用思維鏈 (CoT) 技術建立與優化 Python 程式碼的 GPT-4 智能輔助系統的步驟：

1. 環境設置與檔案準備：在 Google Colab 平台下載 Wave、MFCCs、

Mel_spectrogram 三個 Python 程式檔案，並存為 .py 檔。隨後，使用記事本打開這些 .py 檔，再存為 .txt 格式備用。

2. 登入 GPT-4 平台：選擇左側欄的「探索 GPT」，在畫面右上方選擇「+建立」以開始新的對話及模型建立過程。
3. GPT 建立介面中，提出需求：上傳三個 .txt 格式的 Python 程式檔至知識庫作為 RAG（檢索增強生成）功能使用，並請求優化程式碼及協助撰寫其他 Python 程式碼。設置 GPT 以專門協助優化現有程式碼和撰寫新程式碼，並確保遵循實踐以提高效率、可讀性和維護性。
4. 分享模型：在配置界面中，確立建立完成後。選擇分享 GPT 並設置為「擁有連結的任何人都可訪問」，以便團隊成員可以參與線上討論。
5. 啟動與測試 GPT 模型：啟動 GPT-4 模型，確認其是否能夠正確理解並讀取上傳的程式檔案。進行必要的互動以測試和調整 GPT 的回應和功能，確保其滿足具體需求。
6. 應用思維鏈（Chain-of-Thought, CoT）提示：在與 GPT-4 模型互動時，使用 CoT 提示方法引導模型進行逐步推理。此方法可提高模型在複雜任務上的表現，例如程式碼優化和撰寫。CoT 提示包括將任務分解為多個步驟，並引導模型依序思考每個步驟。這有助於 GPT-4 模型更有條理地處理問題，提供更準確、全面的回應。

7. 對話界面圖：

對話界面的圖像放置於附錄三，提供界面配置與互動示例。

(五)標籤建立與分層抽樣

1. 數據集路徑定義：

設定數據集的存儲位置以方便後續程序高效地讀取文件。

2. 標籤提取與轉換：從音頻文件名中提取標籤信息，並將其轉換成統一的小寫格式以便處理。這一步驟是為了確保標籤的一致性和後續處理的便利性。

3. 標籤處理與合併：根據原始研究論文的分類，對於包含多重疾病標籤的文件名，整合為單一疾病狀態的標籤進行處理。合併後的類別分佈詳見表 3-10。

表 3-10：標籤處理與合併

診斷狀況	原分類人數	原分類音頻數	合併後人數	合併後音頻數
Normal	35	105	35	105
Asthma	32	96	32	96
Pneumonia	5	15	5	15

COPD	9	27	9	27
BRON	3	9	3	9
Heart failure	18	54	21	63
Heart failure + Lung fibrosis	1	3	-	-
Heart failure _and_ COPD	2	6	-	-
Asthma and Lung fibrosis	1	3	-	-
Lung fibrosis	4	12	5	15
Pleural effusion	2	6	2	6

資料來源：本研究整理

4. 標籤編碼與 One-hot 編碼：使用 LabelEncoder 將處理後的文本標籤轉換為數字標籤，然後利用 to_categorical 方法將數字標籤轉換為 One-hot 編碼格式，以滿足神經網絡模型的輸入要求。

(六)分層抽樣應用：為了確保交叉驗證過程中每個 fold 中的類別比例均衡，我們採用了分層抽樣方法。這確保在五個 folds 的分配過程中，每種類別的樣本比例大致與原始數據集中的比例相符，從而提高模型的泛化能力和評估的可靠性。使用 Stratified Fold，將資料分成 5 個 folds，然後每次使用其中的 4 個 folds 進行訓練，並使用剩餘的 1 個 fold 進行驗證如圖 3-9。

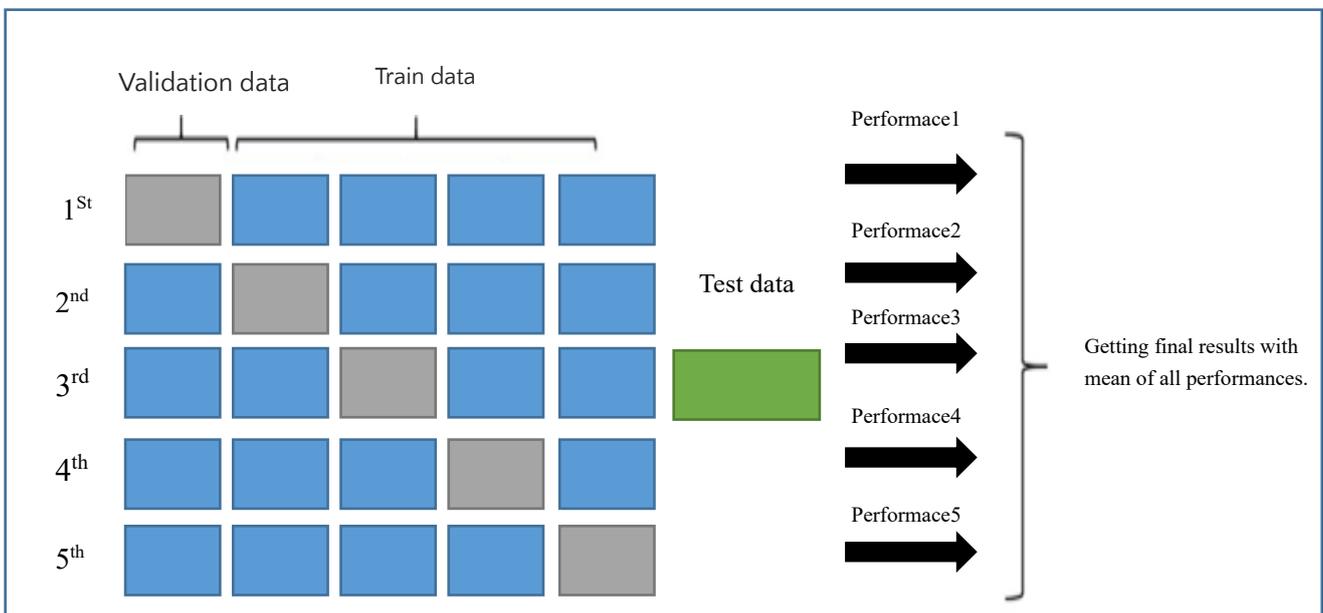


圖 3-9：5K-Fold Cross Validation

資料來源：本研究整理

(七)使用 CNN：選擇使用卷積神經網絡 (CNN) 來處理聲音數據，因為其在特徵提取和分類方面表現優異。該模型的架構包含了 3 個卷積層、3 個最大池化層、1 個扁平層及 2 個全連接層。具體參數配置如表 3-11 所示。

表 3-11：CNN 架構圖

CNN 參數	值
卷積層數量	3
濾波器大小	3x3
池化層數量	3
池化濾波器大小	2x2
全連接層數量	2
激活函數	隱藏層使用 ReLU，輸出層使用 Softmax
正則化方法	Dropout (0.5)、L2 regularization
輸出層分類函數	softmax
損失函數	categorical_crossentropy
優化器	Adam (使用學習速率衰減)
初始學習率	0.0001 (1e-4)
學習率衰減	Exponential Decay (decay rate : 0.9)

資料來源：本研究整理

(八)優化策略

1. StratifiedKFold 交叉驗證：StratifiedKFold 將資料分成若干個折 (folds)，每個折內的類別比例與整個資料集相同。這有助於確保每次訓練和驗證時，各類別的分佈是一致的，從而提高模型的穩定性和泛化能力。
2. 隨機過採樣 (RandomOverSampler)：過採樣用來平衡訓練數據中各類別的樣本數量。這是通過增加少數類別樣本的數量來實現的，從而避免模型偏向於多數類別。
3. 模型預訓練：首先在整個訓練集上預訓練模型，以獲得較好的初始權重。這樣在進行交叉驗證時，每次訓練都是基於已經相對穩定的模型，而不是從零開始。
4. 早期停止 (EarlyStopping)：監控驗證損失 (val_loss)，如果連續幾個周期 (epochs) 驗證損失不再降低，就停止訓練，並恢復到最佳權重。可以防止模型過擬合。
5. 最佳模型權重選擇：在每次交叉驗證中，保存當前 fold 的最佳模型權重，並在所有 fold 完成後，選擇驗證損失最小的模型作為最終模型。

第四章 研究結果

第一節：Fold Validation 結果

通過 5-Fold 交叉驗證，我們計算了每個 fold 的訓練準確率、驗證準確率、訓練損失、驗證損失、精確率、召回率以及早停時的迭代次數。以下是詳細結果：

表 4-1：波形圖 (Waveform) Fold Validation

Item	訓練 準確率	驗證 準確率	訓練 損失	驗證 損失	精確率 (Precision)	召回率 (Recall)	早停時的 迭代次數
Fold 1	98.51%	100.00%	0.12	0.08	98.95%	98.07%	29
Fold 2	95.54%	98.51%	0.22	0.11	96.93%	93.90%	5
Fold 3	95.54%	100.00%	0.18	0.08	96.53%	95.09%	7
Fold 4	95.54%	100.00%	0.21	0.08	96.81%	94.94%	5
Fold 5	97.17%	92.54%	0.16	0.35	97.59%	96.28%	9

資料來源：本研究整理

表 4-2：Mel 頻譜圖 Fold Validation

Item	訓練 準確率	驗證 準確率	訓練 損失	驗證 損失	精確率 (Precision)	召回率 (Recall)	早停時的 迭代次數
Fold 1	94.49%	95.59%	0.27	0.21	96.57%	92.26%	11
Fold 2	94.79%	97.01%	0.24	0.17	96.32%	93.45%	13
Fold 3	91.07%	98.51%	0.33	0.19	93.91%	89.43%	9
Fold 4	93.15%	97.01%	0.27	0.18	95.94%	91.52%	14
Fold 5	80.95%	77.61%	0.60	0.86	86.41%	73.81%	2

資料來源：本研究整理

表 4-3：梅爾頻率倒譜係數 (MFCCs) Fold Validation

Item	訓練 準確率	驗證 準確率	訓練 損失	驗證 損失	精確率 (Precision)	召回率 (Recall)	早停時的 迭代次數
Fold 1	95.24%	98.53%	0.20	0.11	96.35%	94.35%	13
Fold 2	94.79%	97.01%	0.22	0.12	95.30%	93.60%	11
Fold 3	85.27%	97.01%	0.47	0.15	89.47%	82.14%	2
Fold 4	91.82%	98.51%	0.30	0.13	93.75%	89.29%	4
Fold 5	93.90%	82.09%	0.24	0.57	95.12%	92.86%	11

資料來源：本研究整理

表 4-4：GFCCs Fold Validation

Item	訓練 準確率	驗證 準確率	訓練 損失	驗證 損失	精確率 (Precision)	召回率 (Recall)	早停時的 迭代次數
Fold 1	91.07%	91.18%	38.53%	31.92%	94.17%	86.61%	4
Fold 2	91.22%	92.54%	35.30%	28.06%	94.23%	87.50%	6
Fold 3	89.88%	98.51%	39.86%	18.82%	93.29%	84.82%	5
Fold 4	94.35%	95.52%	24.70%	27.23%	95.26%	92.71%	13
Fold 5	95.83%	76.12%	21.40%	84.05%	96.39%	95.39%	15

資料來源：本研究整理

表 4-5：LFCCs Fold Validation

Item	訓練 準確率	驗證 準確率	訓練 損失	驗證 損失	精確率 (Precision)	召回率 (Recall)	早停時的 迭代次數
Fold 1	94.79%	97.06%	24.06%	17.60%	96.10%	91.67%	12
Fold 2	95.68%	97.01%	22.07%	20.05%	96.66%	94.64%	15
Fold 3	91.96%	100.00%	27.71%	12.82%	93.52%	90.18%	18
Fold 4	93.30%	97.01%	26.14%	18.87%	95.92%	90.92%	13
Fold 5	95.98%	82.09%	16.06%	57.47%	96.83%	95.54%	25

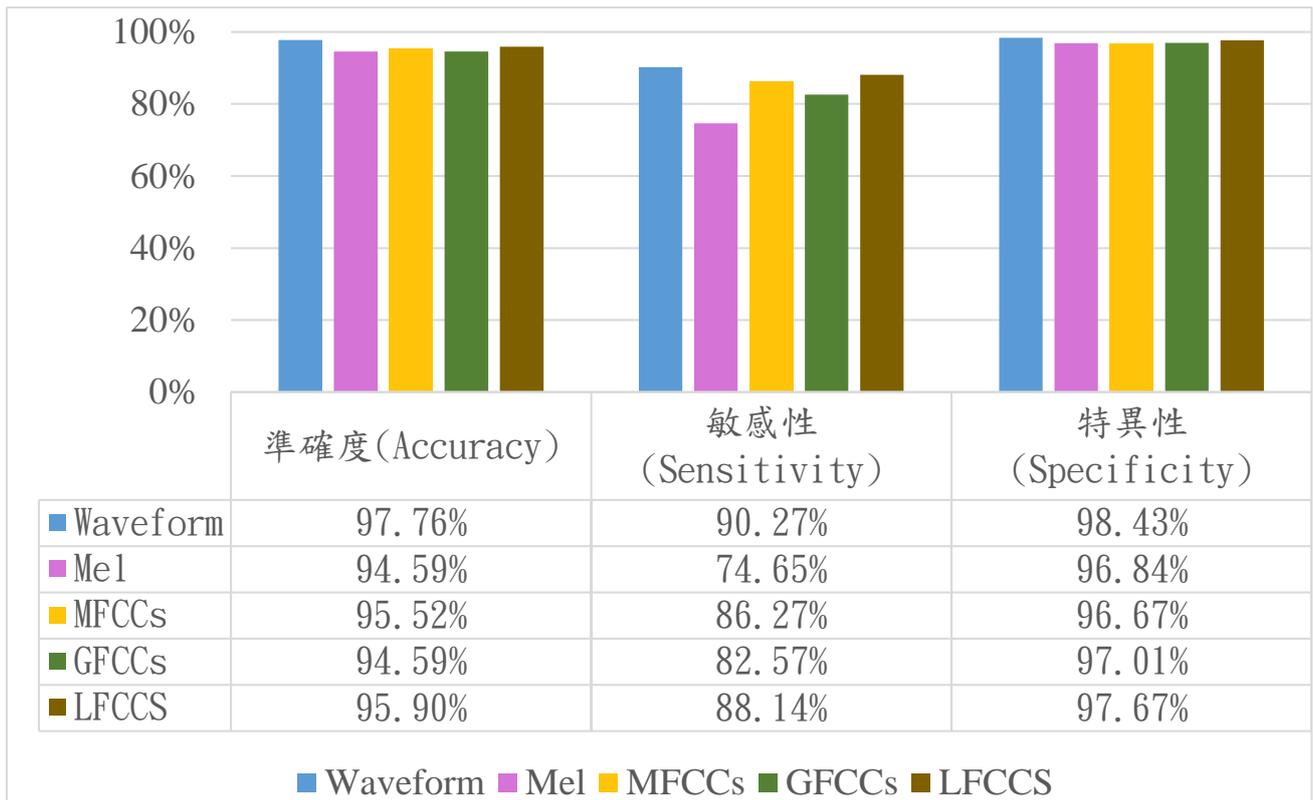
資料來源：本研究整理

第二節：指標分析：

在本研究中，我們深入探討了各種特徵提取方法在識別呼吸系統疾病方面的表現。我們通過三個關鍵性能指標來評估模型效能：準確度 (Accuracy)、敏感性 (Sensitivity) 和特異性 (Specificity)。這些指標的詳細數據呈現在表 4-6 中。值得注意的是，根據我們的分析，Waveform 特徵在所有三個指標上都展現出卓越的表現：準確度達到 97.76%，敏感性為 90.27%，特異性高達 98.43%。這一發現突出了原始波形特徵在肺音分析中的潛力，提醒我們在進行相關研究時不應忽視這一基本但 powerful 的特徵提取方法。為了更直觀地展示和比較各種特徵提取方法的性能，我們繪製了接收者操作特徵 (ROC) 曲線，如圖 4-1 所示。ROC 曲線不僅能夠視覺化地呈現每種方法的表現，還能幫助我們更全面地理解它們在不同閾值下的敏感性和特異性權衡。通過這種綜合分析，我們能夠更清晰地看到各種特徵提取方法的優劣，為後續的肺音分析研究和臨床應用提供了有力的依據。

- 準確度：Accuracy = $\frac{(TP+TN)}{(TP+FP+TN+FN)}$
- 靈敏度 (也稱為召回率)：Sensitivity = $\frac{TP}{(TP+FN)}$
- 特異性：Specificity = $\frac{TN}{(TN+FP)}$
- TP (真陽性)：正確預測為陽性的陽性樣本數。
- FN (假陰性)：錯誤預測為陰性的陽性樣本數。
- TN (真陰性)：正確預測為陰性的陰性樣本數。
- FP (假陽性)：錯誤預測為陽性的陰性樣本數。

表 4-6：關鍵性能指標



資料來源：本研究整理

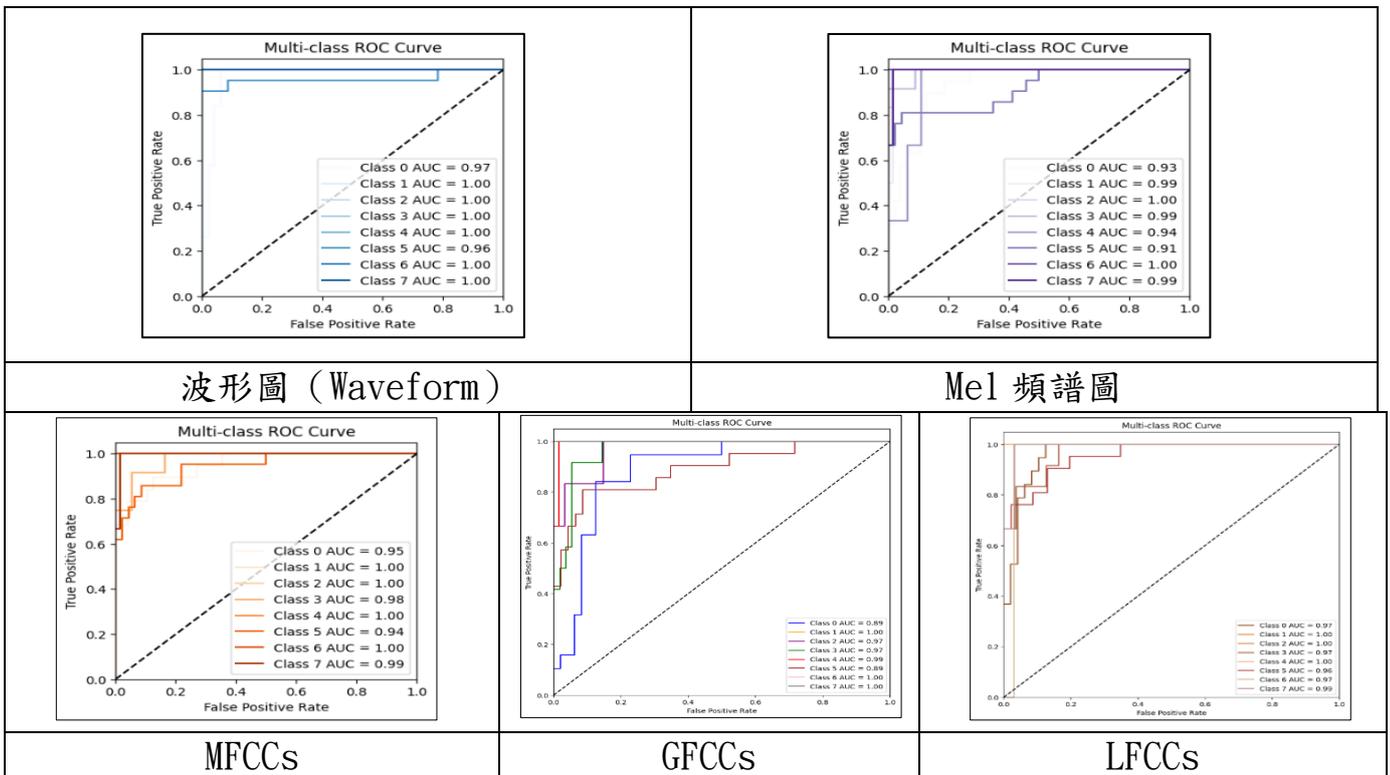


圖 4-1：ROC 曲線圖

資料來源：本研究整理

第五章 討論及應用

第一節：MFCCs 和 Mel 頻譜的敏感度改善

本研究顯著提升了 MFCCs 和 Mel 頻譜在肺音分析中的敏感度。通過增加 Dropout 比例、隨機過採樣 (RandomOverSampler)、模型預訓練、添加 L2 正則化和使用學習率衰減策略，以及早期停止 (EarlyStopping)，CNN 模型在 MFCC 特徵上的敏感度從 62.53% 提升至 86.27%，而 Mel 頻譜的敏感度從 71.67% 提升至 74.65%。這些改進不僅提升了模型對特徵的敏感度，還改善了模型對於不同類別分布數據的適應能力，使其在不同環境中均能保持較高的識別率。

第二節：MFCCs 變體-GFCCs 和 LFCCs 探討

MFCCs 在語音識別和音訊處理領域廣泛應用，並且在肺音分析中展現了其潛力。MFCCs 能夠有效地捕捉音訊的頻譜特性，模擬人耳對聲音的聽覺特性，這對於區分不同類型的肺音（如哮喘音、捻髮音等）具有重要意義。根據研究結果，MFCCs 的準確度為 95.52%，敏感性為 86.27%，特異性為 96.67%，顯示出其在肺音分析中的有效性和穩定性。以下針對 MFCCs 變體做討論。

(一)GFCCs (Gammatone 頻率倒譜係數)

GFCCs 基於 Gammatone 濾波器組提取特徵，模擬人類聽覺系統的頻率選擇性。根據 Xu 等人 (2022) 的研究，GFCCs 的提取過程包括對語音信號進行預強調處理，將信號分為非重疊的連續幀，然後利用圖形頻譜分析 (GFT) 來提取 GFCC 特徵，這有助於更準確地表示語音樣本的結構關係。在本研究中，GFCCs 的準確度為 94.59%，敏感性為 82.57%，特異性為 97.01%，表明其在特定應用中仍具有潛力。

(二)LFCCs (線性頻率倒譜係數)

LFCCs 使用線性頻率刻度，適用於需要線性頻率分辨率的應用。根據 Wu 等人 (2015) 的研究，LFCC 在某些應用中（如語音辨識或音樂分析）可能比傳統的 MFCC 更加合適，因為它提供了不同的頻率分辨率。在本研究中，LFCCs 的準確度為 95.90%，敏感性為 88.14%，特異性為 97.67%，顯示其在肺音分析中具

有良好的應用潛力和優異的性能。

這二種特徵提取方法各有優勢，並且都能在肺音分析中發揮作用：

1. GFCCs：特異性高（97.01%），適合需要高特異性的肺音分析場景。
2. LFCCs：在準確度（95.90%）、敏感性（88.14%）和特異性（97.67%）方面均表現優異，適合需要全面性能的應用場景。

未來的研究應探索如何將這些方法結合，以提升肺音分析的整體性能。例如，將 LFCCs 的高敏感性與 GFCCs 和 LFCCs 的高特異性相結合，可能會提供更全面和精確的診斷工具。通過進一步優化和結合這些特徵提取方法，可以提升遠程醫療診斷的準確性和效率。

第三節：當前應用

(一)專利申請及保護：

為了保護發明，在取得研究成果後，於 2024 年 2 月 17 日提交了新型專利申請，並於 5 月 27 日經經濟部智慧財產局進行說明書補正。最終，專利於 6 月 28 日通過審核，並於 7 月 21 日正式公布（如圖 5-1），以確保技術的獨特性和創新性。



圖 5-1：新型專利具肺音圖像智慧解碼功能的個人健康監控系統及其設備
資料來源：本研究

(二)疾病辨識應用介面

圖 5-2 展示了疾病辨識應用的操作流程。在透過圖像預測疾病類型，並包含以下步驟：

1. 應用的圖標執行檔，該圖標用於代表疾病辨識功能。
2. 主畫面：應用的主畫面，可在此選擇要預測疾病類型的圖像。

3. 預測結果：右下方顯示了應用的預測結果。所選圖像經過分析後，應用會顯示預測的疾病類型，此處顯示的結果為“Asthma”（氣喘）。從選擇圖像到得到預測結果的過程，應用的操作簡單且直觀，能夠有效地幫助用戶進行疾病預測。



圖 5-2 疾病辨識操作流程
資料來源：本研究整理

第四節：未來展望

- (一)跨學科合作未來計劃與醫學工程專家合作，開發肺音收集自動化提示系統器，前後各四點取呼吸音並設定音質警示。(例如：剛好放到骨頭上的位置導致音質不好)。以應對臨床案例的複雜性並確保非專業人士亦能正確使用聽診器如圖 5-3。

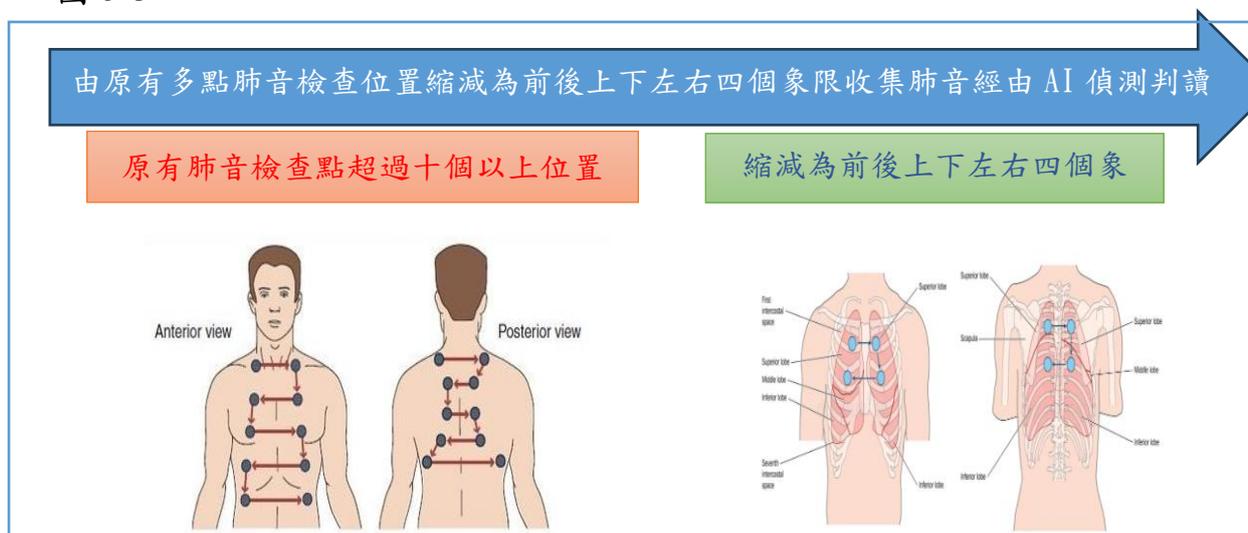


圖 5-3：呼吸音收集圖示
資料來源：本研究整理

- (二)技術發想設計：個人健康監控設備「家用智慧 Lung Sound」(見圖 5-4)：我們想要將這套系統運用於家用智慧 Sound 這項產品。這項產品的功能的特點是它能夠實現遠距醫療提供有效的臨床診斷，進行日常監控以提早發現肺部的

健康問題。我們結合了圖形分析即時分析肺音並用人工智慧來判斷這個患者是否有產生什麼肺部的問題。這些過程只要使用行動裝置應用程式即可實現，因此非常方便。



圖 5-4：個人健康監控設備「家用智慧 Lung Sound」
資料來源：本研究設計規畫

附錄順序

附錄一：B、D、E_Mode frequency ranges

附錄二：B、D、E_Mode durations

附錄三：建立與優化 Python 程式碼的 GPT-4 智能輔助系統

附錄四：調整前後 CNN 訓練碼

附錄五：音訊圖檔 mapping

附錄六：診斷分類圖檔範例

附錄七：新型申請專利範圍

附錄八：音訊圖檔 mapping

第六章 參考文獻

1. 國家通訊傳播委員會 (2024)。行動通信服務客戶統計。
https://www.ncc.gov.tw/chinese/news.aspx?site_content_sn=2017
2. Andrès, E., Gass, R., Charloux, A., Brandt, C., & Hentzler, A. (2018). Respiratory sound analysis in the era of evidence-based medicine and the world of medicine 2.0. *Journal of medicine and life*, 11(2), 89.
3. Bisong, E. (2019). Building machine learning and deep learning models on Google cloud platform (pp. 59-64). Berkeley, CA: Apress.
4. Carneiro, T., Nóbrega, R. V. M., Nepomuceno, T. G., Bian, G.-B., Albuquerque, V. H. C., & Rebouças Filho, P. P. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685. doi:10.1109/access.2018.2874767
5. Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4), 357-366.
6. Fraiwan, M., Fraiwan, L., Khassawneh, B., & Ibnian, A. (2021). A dataset of lung sounds recorded from the chest wall using an electronic stethoscope. *Data in Brief*, 35, 106913.
7. Giannakopoulos, T., & Pikrakis, A. (2014). Introduction to audio analysis: a MATLAB® approach. Academic Press.
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
9. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
10. Lau, J., Sim, D., Tan, C. H., & Low, J. (2020). Using Google Colab for NLP Experiments. arXiv preprint arXiv:2006.02635. Retrieved from <https://arxiv.org/abs/2006.02635>
11. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
12. Nowak, L. J., & Nowak, K. M. (2017). Acoustic characterization of stethoscopes using auscultation sounds as test signals. *The Journal of the Acoustical Society of America*, 141(3), 1940-1946.
13. Rabiner, L., & Schafer, R. (2010). *Theory and applications of digital speech processing*.

Prentice Hall Press.

14. Rabiner, L. R., & Juang, B. H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall.
15. Sakula, A. R. T. H. (1981). RTH Laennec 1781--1826 his life and work: a bicentenary appreciation. *Thorax*, 36(2), 81-90.
16. Sarkar, M., Madabhavi, I., Niranjana, N., & Dogra, M. (2015). Auscultation of the respiratory system. *Annals of thoracic medicine*, 10(3), 158-168.
17. SCHERER, Dominik; MÜLLER, Andreas; BEHNKE, Sven. Evaluation of pooling operations in convolutional architectures for object recognition. In: *International conference on artificial neural networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 92-101.
18. Stevens, S. S., Volkman, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *Journal of the Acoustical Society of America*, 8, 185–190. <https://doi.org/10.1121/1.1915893>
19. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824-24837.
20. Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., & Li, H. (2015). Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66, 130-153. doi:10.1016/j.specom.2014.12.005.
21. Xu, L., Tian, M., Guo, X., Shan, Z., Jia, J., Peng, Y., Yang, J., & Das, R. K. (2022). A novel feature based on graph signal processing for detection of physical access attacks. *Proceedings of the Speaker and Language Recognition Workshop (Odyssey 2022)*, 107-111. doi:10.21437/Odyssey.2022-15.

附錄

附錄一：B、D、E_Mode frequency ranges

No	Filename	B_Maximum Frequency (Hz)	D_Maximum Frequency (Hz)	E_Maximum Frequency (Hz)
1	Asthma	0~1999.97	0~1999.97	0~1999.97
2	Asthma	0~1999.97	0~1999.97	0~1999.97
3	Asthma	0~1999.97	0~1999.97	0~1999.97
4	Heart Failure + Lung Fibrosis	0~1999.96	0~1999.96	0~1999.96
5	Heart Failure + COPD	0~1999.96	0~1999.96	0~1999.96
6	Plueral Effusion	0~1999.96	0~1999.96	0~1999.96
7	Heart Failure + COPD	0~1999.95	0~1999.95	0~1999.95
8	Plueral Effusion	0~1999.95	0~1999.95	0~1999.95
9	Asthma	0~1999.93	0~1999.93	0~1999.93
10	Asthma	0~1999.83	0~1999.83	0~1999.83
11	Heart Failure	0~1999.96	0~1999.96	0~1999.96
12	Asthma	0~1999.92	0~1999.92	0~1999.92
13	Heart Failure	0~1999.93	0~1999.93	0~1999.93
14	Heart Failure	0~1999.92	0~1999.92	0~1999.92
15	Asthma	0~1999.9	0~1999.9	0~1999.9
16	heart failure	0~1999.92	0~1999.92	0~1999.92
17	COPD	0~1999.94	0~1999.94	0~1999.94
18	pneumonia	0~1999.96	0~1999.96	0~1999.96
19	heart failure	0~1999.96	0~1999.96	0~1999.96
20	Asthma and lung fibrosis	0~1999.95	0~1999.95	0~1999.95
21	BRON	0~1999.95	0~1999.95	0~1999.95
22	BRON	0~1999.8	0~1999.8	0~1999.8
23	Lung Fibrosis	0~1999.91	0~1999.91	0~1999.91
24	heart failure	0~1999.9	0~1999.9	0~1999.9
25	copd	0~1999.91	0~1999.91	0~1999.91

26	Lung Fibrosis	0~1999.94	0~1999.94	0~1999.94
27	asthma	0~1999.93	0~1999.93	0~1999.93
28	BRON	0~1999.94	0~1999.94	0~1999.94
29	heart failure	0~1999.88	0~1999.88	0~1999.88
30	Normal	0~1999.92	0~1999.92	0~1999.92
31	Normal	0~1999.95	0~1999.95	0~1999.95
32	Normal	0~1999.95	0~1999.95	0~1999.95
33	Asthma	0~1999.94	0~1999.94	0~1999.94
34	heart failure	0~1999.95	0~1999.95	0~1999.95
35	Asthma	0~1999.94	0~1999.94	0~1999.94
36	pneumonia	0~1999.97	0~1999.97	0~1999.97
37	pneumonia	0~1999.91	0~1999.91	0~1999.91
38	Asthma	0~1999.94	0~1999.94	0~1999.94
39	heart failure	0~1999.96	0~1999.96	0~1999.96
40	heart failure	0~1999.93	0~1999.93	0~1999.93
41	pneumonia	0~1999.89	0~1999.89	0~1999.89
42	heart failure	0~1999.92	0~1999.92	0~1999.92
43	asthma	0~1999.93	0~1999.93	0~1999.93
44	asthma	0~1999.93	0~1999.93	0~1999.93
45	heart failure	0~1999.9	0~1999.9	0~1999.9
46	asthma	0~1999.93	0~1999.93	0~1999.93
47	asthma	0~1999.94	0~1999.94	0~1999.94
48	Normal	0~1999.94	0~1999.94	0~1999.94
49	Normal	0~1999.93	0~1999.93	0~1999.93
50	Normal	0~1999.95	0~1999.95	0~1999.95
51	Normal	0~1999.94	0~1999.94	0~1999.94
52	Lung Fibrosis	0~1999.89	0~1999.89	0~1999.89
53	asthma	0~1999.88	0~1999.88	0~1999.88
54	heart failure	0~1999.93	0~1999.93	0~1999.93

55	asthma	0~1999.89	0~1999.89	0~1999.89
56	asthma	0~1999.92	0~1999.92	0~1999.92
57	COPD	0~1999.94	0~1999.94	0~1999.94
58	asthma	0~1999.96	0~1999.96	0~1999.96
59	heart failure	0~1999.95	0~1999.95	0~1999.95
60	heart failure	0~1999.93	0~1999.93	0~1999.93
61	COPD	0~1999.94	0~1999.94	0~1999.94
62	COPD	0~1999.97	0~1999.97	0~1999.97
63	COPD	0~1999.94	0~1999.94	0~1999.94
64	asthma	0~1999.96	0~1999.96	0~1999.96
65	asthma	0~1999.96	0~1999.96	0~1999.96
66	heart failure	0~1999.94	0~1999.94	0~1999.94
67	heart failure	0~1999.95	0~1999.95	0~1999.95
68	asthma	0~1999.95	0~1999.95	0~1999.95
69	pneumonia	0~1999.96	0~1999.96	0~1999.96
70	Normal	0~1999.93	0~1999.93	0~1999.93
71	Normal	0~1999.93	0~1999.93	0~1999.93
72	Normal	0~1999.96	0~1999.96	0~1999.96
73	Normal	0~1999.95	0~1999.95	0~1999.95
74	Normal	0~1999.96	0~1999.96	0~1999.96
75	Normal	0~1999.96	0~1999.96	0~1999.96
76	Normal	0~1999.95	0~1999.95	0~1999.93
77	Asthma	0~1999.93	0~1999.93	0~1999.93
78	Asthma	0~1999.95	0~1999.95	0~1999.95
79	asthma	0~1999.95	0~1999.95	0~1999.95
80	asthma	0~1999.94	0~1999.94	0~1999.94
81	Normal	0~1999.96	0~1999.96	0~1999.96
82	Normal	0~1999.95	0~1999.95	0~1999.95
83	Normal	0~1999.96	0~1999.96	0~1999.96

84	Normal	0~1999.96	0~1999.96	0~1999.96
85	Normal	0~1999.92	0~1999.92	0~1999.92
86	Normal	0~1999.93	0~1999.93	0~1999.93
87	Normal	0~1999.94	0~1999.94	0~1999.94
88	Normal	0~1999.96	0~1999.96	0~1999.96
89	Normal	0~1999.95	0~1999.95	0~1999.95
90	heart failure	0~1999.93	0~1999.93	0~1999.93
91	asthma	0~1999.92	0~1999.92	0~1999.92
92	Normal	0~1999.92	0~1999.92	0~1999.92
93	Normal	0~1999.9	0~1999.9	0~1999.9
94	Normal	0~1999.93	0~1999.93	0~1999.93
95	Normal	0~1999.93	0~1999.93	0~1999.93
96	Normal	0~1999.94	0~1999.94	0~1999.94
97	Asthma	0~1999.95	0~1999.95	0~1999.95
98	Normal	0~1999.95	0~1999.95	0~1999.95
99	Normal	0~1999.96	0~1999.96	0~1999.96
100	Normal	0~1999.93	0~1999.93	0~1999.93
101	Asthma	0~1999.93	0~1999.93	0~1999.93
102	Normal	0~1999.95	0~1999.95	0~1999.95
103	Normal	0~1999.92	0~1999.92	0~1999.92
104	Asthma	0~1999.91	0~1999.91	0~1999.91
105	Lung Fibrosis	0~1999.96	0~1999.96	0~1999.96
106	Asthma	0~1999.95	0~1999.95	0~1999.95
107	Asthma	0~1999.92	0~1999.92	0~1999.92
108	COPD	0~1999.96	0~1999.96	0~1999.96
109	Normal	0~1999.92	0~1999.92	0~1999.92
110	COPD	0~1999.92	0~1999.92	0~1999.92
111	COPD	0~1999.94	0~1999.94	0~1999.94
112	Normal	0~1999.92	0~1999.92	0~1999.92

```

import os
import csv
import numpy as np
from scipy.io import wavfile
from scipy.fft import fft

# 設定 WAV 檔案的目錄路徑
directory = r"C:\LungSounds\input\AudioFiles"

# CSV 檔案名稱和路徑
duration_output_csv = r"C:\LungSounds\wav_durations.csv"
frequency_output_csv = r"C:\LungSounds\frequency_ranges.csv"

# 定義函數來計算頻率範圍
def find_frequency_range(wav_filename):
    # 讀取音頻檔案
    samplerate, data = wavfile.read(wav_filename)

    # 如果是立體聲，只取一個聲道
    if len(data.shape) > 1:
        data = data[:, 0]

    # 執行 FFT 變換
    fft_spectrum = fft(data)

    # 計算頻率軸
    freqs = np.fft.fftfreq(len(fft_spectrum), 1 / samplerate)

    # 取絕對值並去除負頻率部分
    magnitude = np.abs(fft_spectrum)[:len(fft_spectrum) // 2]
    positive_freqs = freqs[:len(freqs) // 2]

    # 找出最高和最低頻率的非零值

```

```

min_freq = np.min(positive_freqs[np.nonzero(magnitude)])
max_freq = np.max(positive_freqs[np.nonzero(magnitude)])
return min_freq, max_freq
# 開啟 CSV 檔案進行寫入持續時間數據
with open(duration_output_csv, 'w', newline='') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(['Filename', 'Length (seconds)']) # 寫入表頭
    # 遍歷目錄下的所有檔案
    for filename in os.listdir(directory):
        if filename.endswith('.wav'): # 確保只處理 WAV 檔案
            filepath = os.path.join(directory, filename)
            samplerate, data = wavfile.read(filepath)
            length = data.shape[0] / samplerate
            csvwriter.writerow([filename, length]) # 寫入檔案名稱和時長
print(f"Duration data written to {duration_output_csv}")
# 開啟 CSV 檔案進行寫入頻率範圍數據
with open(frequency_output_csv, 'w', newline='') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(['Filename', 'Minimum Frequency (Hz)', 'Maximum Frequency (Hz)'])
    # 遍歷目錄下的所有檔案
    for filename in os.listdir(directory):
        if filename.endswith('.wav'):
            filepath = os.path.join(directory, filename)
            min_freq, max_freq = find_frequency_range(filepath)
            csvwriter.writerow([filename, min_freq, max_freq])
print(f"Frequency range data written to {frequency_output_csv}")

```

附錄二：B、D、E_Mode durations

No	Filename	B_Mode_Length (seconds)	D_Mode_Length (seconds)	E_Mode_Length (seconds)
1	Asthma	30.00	30.00	30.00
2	Asthma	5.94	5.94	5.94
3	Asthma	13.55	13.55	14.96
4	Heart Failure + Lung Fibrosis	13.55	13.55	13.55
5	Heart Failure + COPD	19.06	19.06	19.06
6	Plueral Effusion	12.85	12.85	12.85
7	Heart Failure + COPD	11.76	11.76	11.76
8	Plueral Effusion	25.26	25.26	25.26
9	Asthma	19.82	19.82	19.82
10	Asthma	12.66	12.66	12.66
11	Heart Failure	22.83	22.83	22.83
12	Asthma	13.23	13.23	13.23
13	Heart Failure	24.62	24.62	24.62
14	Heart Failure	12.66	12.66	12.66
15	Asthma	17.07	17.07	17.07
16	heart failure	12.66	12.66	12.66
17	COPD	12.53	12.53	12.53
18	pneumonia	14.58	14.58	14.58
19	heart failure	12.78	12.78	12.78
20	Asthma and lung fibrosis	10.35	10.35	10.35
21	BRON	13.17	13.17	13.17
22	BRON	15.54	15.54	15.54
23	Lung Fibrosis	23.47	23.47	23.47
24	heart failure	23.86	23.86	23.86
25	copd	30.00	30.00	30.00
26	Lung Fibrosis	18.48	18.48	18.48
27	asthma	19.38	19.38	19.38

28	BRON	5.00	5.00	5.00
29	heart failure	11.38	11.38	11.38
30	Normal	9.58	9.58	9.58
31	Normal	11.44	11.44	11.44
32	Normal	15.73	15.73	15.73
33	Asthma	13.94	13.94	13.94
34	heart failure	15.86	15.86	15.86
35	Asthma	8.37	8.37	8.37
36	pneumonia	30.00	30.00	30.00
37	pneumonia	12.14	12.14	12.14
38	Asthma	20.08	20.08	20.08
39	heart failure	18.35	18.35	18.35
40	heart failure	17.57	17.57	17.57
41	pneumonia	19.18	19.18	19.18
42	heart failure	15.98	15.98	15.98
43	asthma	30.00	30.00	30.00
44	asthma	11.18	11.18	11.18
45	heart failure	15.98	15.98	15.98
46	asthma	22.26	22.26	22.26
47	asthma	24.94	24.94	24.94
48	Normal	13.36	13.36	13.36
49	Normal	9.07	9.07	9.07
50	Normal	12.66	12.66	12.66
51	Normal	14.83	14.83	14.83
52	Lung Fibrosis	14.38	14.38	14.38
53	asthma	9.78	9.78	9.78
54	heart failure	13.55	13.55	13.55
55	asthma	16.94	16.94	16.94
56	asthma	15.47	15.47	15.47

57	COPD	13.36	13.36	13.36
58	asthma	27.57	27.57	27.57
59	heart failure	19.44	19.44	19.44
60	heart failure	16.69	16.69	16.69
61	COPD	9.26	9.26	9.26
62	COPD	8.69	8.69	8.69
63	COPD	14.06	14.06	14.06
64	asthma	9.14	9.14	9.14
65	asthma	11.95	11.95	11.95
66	heart failure	15.98	15.98	15.98
67	heart failure	23.54	23.54	23.54
68	asthma	18.74	18.74	18.74
69	pneumonia	28.53	28.53	28.53
70	Normal	13.36	13.36	13.36
71	Normal	15.47	15.47	15.47
72	Normal	30.00	30.00	30.00
73	Normal	17.33	17.33	17.33
74	Normal	23.09	23.09	23.09
75	Normal	24.94	24.94	24.94
76	Normal	15.54	15.54	15.54
77	Asthma	18.86	18.86	18.86
78	Asthma	19.89	19.89	19.89
79	asthma	24.18	24.18	24.18
80	asthma	22.13	22.13	22.13
81	Normal	13.74	13.74	13.74
82	Normal	14.38	14.38	14.38
83	Normal	24.24	24.24	24.24
84	Normal	19.57	19.57	19.57
85	Normal	26.35	26.35	26.35

86	Normal	23.47	23.47	23.47
87	Normal	20.14	20.14	14.58
88	Normal	14.58	14.58	14.58
89	Normal	20.27	20.27	20.27
90	heart failure	20.34	20.34	20.34
91	asthma	22.06	22.06	22.06
92	Normal	18.03	18.03	18.03
93	Normal	23.54	23.54	23.54
94	Normal	18.67	18.67	18.67
95	Normal	25.65	25.65	25.65
96	Normal	25.78	25.78	25.78
97	Asthma	12.27	12.27	12.27
98	Normal	14.45	14.45	14.45
99	Normal	17.46	17.46	17.46
100	Normal	23.73	23.73	23.73
101	Asthma	22.06	22.06	22.06
102	Normal	15.34	15.34	15.34
103	Normal	14.96	14.96	14.96
104	Asthma	12.14	12.14	12.14
105	Lung Fibrosis	12.46	12.46	12.46
106	Asthma	9.84	9.84	9.84
107	Asthma	14.38	14.38	14.38
108	COPD	13.55	13.55	13.55
109	Normal	15.47	15.47	15.47
110	COPD	21.94	21.94	21.94
111	COPD	18.93	18.93	18.93
112	Normal	27.89	27.89	27.89

附錄三:建立與優化 Python 程式碼的 GPT-4 智能輔助系統

1.登入並初始化 GPT-4 模型：

- 登入 GPT-4 平台並選擇左側欄的「探索 GPT」，在畫面右上方選擇“+建立”以開始新的對話及模型建立過程。



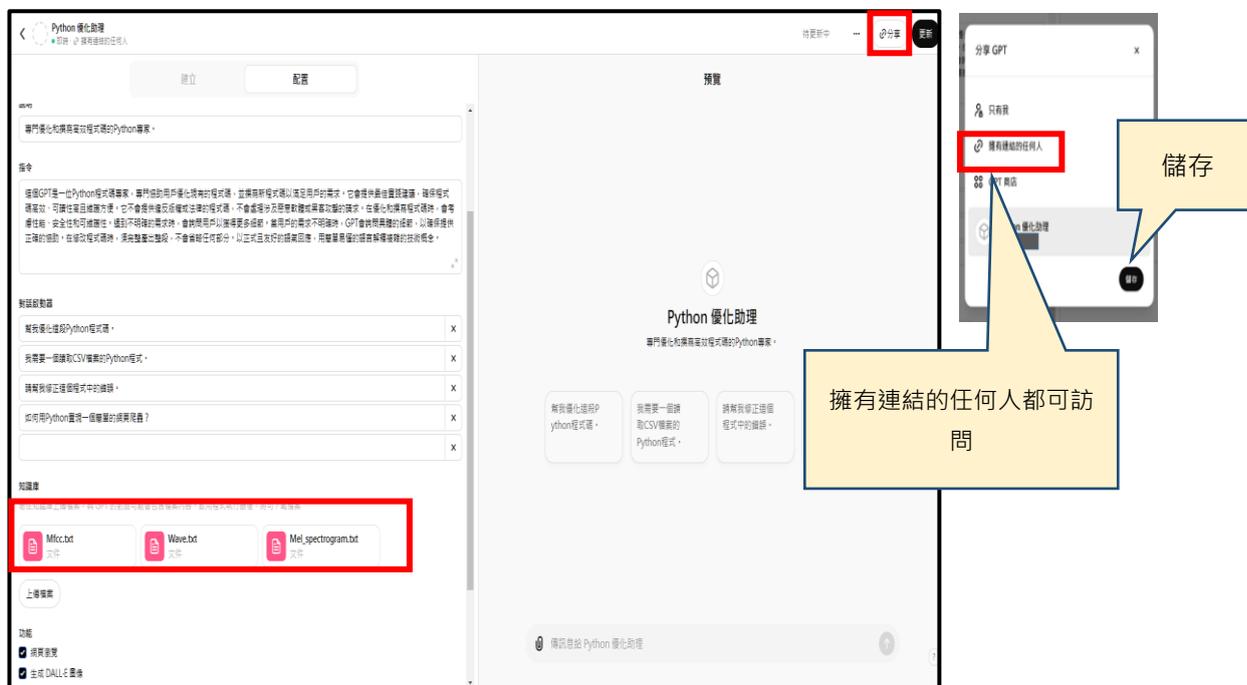
2.創建與配置 GPT 模型：

- 在 GPT 建立介面中，提出需求：上傳三個.txt 格式的 Python 程式檔至知識庫作為 RAG 使用，並請求優化程式碼及協助撰寫其他 Python 程式碼。設置 GPT 以專門協助用戶優化現有程式碼和撰寫新程式碼，並確保遵循最佳實踐以提高效率、可讀性和維護性。



3.上傳檔案並分享模型：

- 在配置界面中，上傳之前準備的三個.txt 檔案供 GPT-4 RAG 檢索。建立完成後，選擇分享 GPT 並設置為“擁有連結的任何人都可訪問”，以便團隊成員可以參與線上討論。



4.啟動與測試 GPT 模型：

- 啟動 GPT-4 模型，確認其是否能夠正確理解並讀取上傳的程式檔案。進行必要的互動以測試和調整 GPT 的回應和功能，確保其滿足具體需求。



5.應用思維鏈（Chain-of-Thought, CoT）提示：

- 在與 GPT-4 模型互動時，使用 CoT 提示方法引導模型進行逐步推理。此方法可提高模型在複雜任務上的表現，例如程式碼優化和撰寫。CoT 提示包括將任務分解為多個步驟，並引導模型依序思考每個步驟。這有助於 GPT-4 模型更有條理地處理問題，提供更準確、全面的回應。

Prompt 前加入：

Let's think about it step by step. Use COT to explain the thinking process of answering the question.

Python 優化助理 ▾

Let's think about it step by step.
Use COT to explain the thinking process of answering the question.
比較知識庫三個python程式碼 列出可以優化敏感度的方法
使用繁體中文回答

在比較三個程式碼（Wave、Mfcc、Mel_spectrogram）之後，我們可以找出幾個共同的優化方向，特別是在敏感度（sensitivity）方面。以下是逐步分析和優化建議：

1. 資料預處理

資料的準確預處理對於提高模型的敏感度至關重要。

- 波形（Wave）和 MFCC: 確保音訊資料正確讀取並標準化。

```
python
def read_wave(wav_file_name):
    if os.path.exists(wav_file_name):
        sample_rate, data = wavfile.read(wav_file_name)
        data = data.astype(np.float32) / np.max(np.abs(data)) # 標準化
        return data, sample_rate
    else:
        print(f"File not found: {wav_file_name}")
        return None, None
```

- 梅爾頻譜（Mel Spectrogram）: 使用 `librosa` 處理音訊資料，並確保頻譜圖正確生成。

```
python
def plot_mel_spectrogram(data, sample_rate, count):
    n_mels = 300
```

附錄四：調整前後 CNN 訓練碼

調整前

```
import numpy as np
from keras.optimizers import Adam
from keras.layers import Conv2D, Flatten, Dense, Dropout, MaxPooling2D
from keras.models import Sequential
from keras.callbacks import EarlyStopping
from sklearn.metrics import roc_auc_score, roc_curve, auc, confusion_matrix
import pandas as pd
import matplotlib.pyplot as plt
import itertools
from keras.metrics import Precision, Recall
from keras.backend import cast, sum
import tensorflow as tf
from keras import backend as K
from sklearn.model_selection import StratifiedKFold
from imblearn.over_sampling import RandomOverSampler # 從對應模組導入 RandomOverSampler
from keras.layers import BatchNormalization
from keras.regularizers import l2
from keras.optimizers.schedules import ExponentialDecay
from keras.layers import Activation
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from keras.optimizers import Adam
from keras.metrics import Precision, Recall
from keras.callbacks import EarlyStopping
# 定義 CNN 模型結構
def cnn_model() :
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(300,800,3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(128, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Dropout(0.3))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(8, activation='softmax'))
```

```

optimizer = Adam(learning_rate=1e-4)
model.compile(loss='categorical_crossentropy', optimizer=optimizer,
              metrics=['accuracy', Precision(name='precision'), Recall(name='recall')])
return model

model = cnn_model()
print(model.summary())
early_stop = EarlyStopping(monitor='val_loss', patience=5, verbose=1, restore_best_weights=True)
# 使用整個訓練集進行預訓練
print("進行模型的預訓練...")
history_pretrain = model.fit(X_train, y_train, epochs=100, batch_size=16, validation_data=(X_valid, y_valid),
                             callbacks=[early_stop])
# 儲存預訓練模型的權重
pretrained_weights = 'pretrained_weights.h5'
model.save_weights(pretrained_weights)
# StratifiedKfold 微調
best_val_loss = float('inf') # 初始化為正無窮，表示我們還沒有找到任何模型
best_weights = None # 這將存儲最佳模型的權重
for i, (train_idx, valid_idx) in enumerate(skf.split(img_list, np.argmax(oh_labels, axis=1))) :
    X_train_fold, X_valid_fold = np.array(img_list)[train_idx], np.array(img_list)[valid_idx]
    y_train_fold, y_valid_fold = np.array(oh_labels)[train_idx], np.array(oh_labels)[valid_idx]
    # 使用 RandomOverSampler 處理不平衡數據
    X_train_fold_2d = X_train_fold.reshape((X_train_fold.shape[0], -1))
    X_train_resampled_2d, y_train_resampled = ros.fit_resample(X_train_fold_2d, y_train_fold)
    X_train_resampled = X_train_resampled_2d.reshape((-1, X_train_fold.shape[1], X_train_fold.shape[2],
X_train_fold.shape[3]))
    # 使用預訓練的權重初始化模型
    model = cnn_model()
    model.load_weights(pretrained_weights)
    # 微調模型
    print(f"進行模型的微調 - Fold {i + 1} ...")
    history = model.fit(X_train_resampled, y_train_resampled, epochs=100, batch_size=16,
validation_data=(X_valid_fold, y_valid_fold), callbacks=[early_stop])
    model.save_weights(f"/content/drive/MyDrive/LungSounds/results/Melmodel_weights_model{i+1}.h5")
    y_pred_probs = model.predict(X_valid_fold)
    y_pred_class = np.argmax(y_pred_probs, axis=1)
    y_true_class = np.argmax(y_valid_fold, axis=1)
    # 檢查當前模型的驗證損失
    current_val_loss = min(history.history['val_loss'])
    best_val_loss = current_val_loss
    if current_val_loss < best_val_loss :

```

```

best_weights = model.get_weights() # 保存當前模型的權重
# 循環結束後，使用最佳權重
    model.set_weights(best_weights)
# 定義指標
def precision_np(y_true, y_pred) :
    TP = np.sum(np.logical_and(y_pred == 1, y_true == 1))
    FP = np.sum(np.logical_and(y_pred == 1, y_true == 0))
    return TP / (TP + FP) if (TP + FP) != 0 else 0
def recall_np(y_true, y_pred) :
    TP = np.sum(np.logical_and(y_pred == 1, y_true == 1))
    FN = np.sum(np.logical_and(y_pred == 0, y_true == 1))
    return TP / (TP + FN) if (TP + FN) != 0 else 0
def f1_score_np(y_true, y_pred) :
    precision = precision_np(y_true, y_pred)
    recall = recall_np(y_true, y_pred)

    # 確保分母不為 0
    denominator = precision + recall
    if denominator == 0 :
        return 0.0
    else :
        return 2 * ((precision * recall) / denominator)
def sensitivity_np(y_true, y_pred) :
    TP = np.sum(np.logical_and(y_pred == 1, y_true == 1))
    FN = np.sum(np.logical_and(y_pred == 0, y_true == 1))
    # 避免除以零的情況
    return TP / (TP + FN) if (TP + FN) != 0 else 0
def specificity_np(y_true, y_pred) :
    TN = np.sum(np.logical_and(y_pred == 0, y_true == 0))
    FP = np.sum(np.logical_and(y_pred == 1, y_true == 0))
    return TN / (TN + FP) if (TN + FP) != 0 else 0
def ppv_np(y_true, y_pred) :
    TP = np.sum(np.logical_and(y_pred == 1, y_true == 1))
    FP = np.sum(np.logical_and(y_pred == 1, y_true == 0))
    return TP / (TP + FP) if (TP + FP) != 0 else 0
def npv_np(y_true, y_pred) :
    TN = np.sum(np.logical_and(y_pred == 0, y_true == 0))
    FN = np.sum(np.logical_and(y_pred == 0, y_true == 1))
    return TN / (TN + FN) if (TN + FN) != 0 else 0

```

```

# 計算 roc_auc
roc_auc = dict()
for i in range(8) :
    if len(np.unique(y_valid[ : , i])) == 2 : # Check if there are two unique classes
        roc_auc[i] = roc_auc_score(y_valid[ : , i], y_pred_probs[ : , i])
    else :
        roc_auc[i] = np.nan # Set to NaN or some other placeholder value if there's only one class
# 定義混淆矩陣繪製函數
def plot_confusion_matrix(cm, classes, normalize=False, title='Confusion Matrix', cmap=plt.cm.Purples) :
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
    if normalize :
        cm = cm.astype('float') / cm.sum(axis=1)[ : , np.newaxis]
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])) :
        plt.text(j, i, cm[i, j], horizontalalignment="center", color="white" if cm[i, j] > thresh else "black")
    plt.tight_layout()
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")
# 繪製多類別 ROC 曲線的函數
def plot_roc_curve(y_true, y_pred_probs, n_classes) :
    plt.figure(figsize=(10, 8))
    fpr = dict()
    tpr = dict()
    roc_auc = dict() # 這裡只是初始化 roc_auc 為空字典
    for i in range(n_classes) :
        if len(np.unique(y_true[ : , i])) > 1 :
            fpr[i], tpr[i], _ = roc_curve(y_true[ : , i], y_pred_probs[ : , i])
            roc_auc[i] = auc(fpr[i], tpr[i])
            plt.plot(fpr[i], tpr[i], color=plt.cm.Purples(i / n_classes), label='Class {0} AUC = {1 : 0.2f}'.format(i,
roc_auc[i]))
        else :
            # Set ROC AUC to NaN or other default value if preferred
            roc_auc[i] = np.nan

```

```

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Multi-class ROC Curve')
plt.legend(loc="lower right")
return roc_auc
# 在呼叫混淆矩陣和 ROC 曲線的函數之前，您需要先計算混淆矩陣
cm = confusion_matrix(y_true_class, y_pred_class)

# 呼叫混淆矩陣和 ROC 曲線的函數
plot_confusion_matrix(cm, classes=['class0', 'class1', 'class2', 'class3', 'class4', 'class5', 'class6', 'class7'])
plot_roc_curve(y_valid, y_pred_probs, 8)

roc_auc = dict()
for i in range(8):
    if len(np.unique(y_valid[:, i])) == 2: # 檢查是否存在兩個獨立類別
        roc_auc[i] = roc_auc_score(y_valid[:, i], y_pred_probs[:, i])
    else:
        roc_auc[i] = np.nan # 若只有一個類別，則設為 NaN 或其他表示無法計算的值
# 計算每個類別的 AUC 值並儲存到列表中
results = []
# 定義 total_samples
total_samples = y_valid.shape[0]

for i in range(8):
    y_true_i = y_valid[:, i]
    y_pred_i = np.round(y_pred_probs[:, i])
    correct_preds = np.sum(y_true_i == y_pred_i)
    acc = correct_preds / total_samples # 使用 total_samples 而不是 class_samples
    results.append({
        '類別': i,
        'Accuracy': acc,
        'AUC': roc_auc[i],
        'sensitivity': sensitivity_np(y_true_i, y_pred_i),
        'specificity': specificity_np(y_true_i, y_pred_i),
        'PPV': ppv_np(y_true_i, y_pred_i),
        'NPV': npv_np(y_true_i, y_pred_i),
    })

```

```

    'F1_score' : f1_score_np(y_true_i, y_pred_i),
    'precision' : precision_np(y_true_i, y_pred_i),
    'recall' : recall_np(y_true_i, y_pred_i)
})
# 將 results 轉換為 DataFrame
df = pd.DataFrame(results)
# 計算總平均指標
mean_accuracy = df['Accuracy'].mean()
mean_auc = df['AUC'].mean()
mean_sensitivity = df['sensitivity'].mean()
mean_specificity = df['specificity'].mean()
mean_ppv = df['PPV'].mean()
mean_npv = df['NPV'].mean()
mean_f1 = df['F1_score'].mean()
mean_precision = df['precision'].mean()
mean_recall = df['recall'].mean()
# 將總平均指標加入到 DataFrame
df = df.append({
    '類別' : '總平均',
    'Accuracy' : mean_accuracy,
    'AUC' : mean_auc,
    'sensitivity' : mean_sensitivity,
    'specificity' : mean_specificity,
    'PPV' : mean_ppv,
    'NPV' : mean_npv,
    'F1_score' : mean_f1,
    'precision' : mean_precision,
    'recall' : mean_recall
}, ignore_index=True)
# 將結果保存到 CSV 文件中
df.to_csv('Mel_results.csv', index=False)

```

CNN 調整後

```

# 使用帶有學習速率衰減的 Adam 優化器
optimizer = Adam(learning_rate=lr_schedule)
# 定義 CNN 模型結構
def cnn_model() :
    model = Sequential()

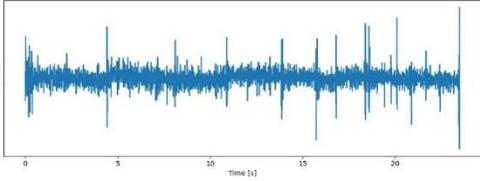
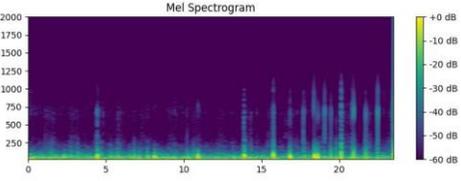
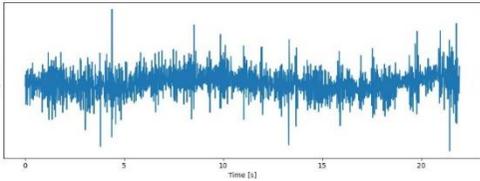
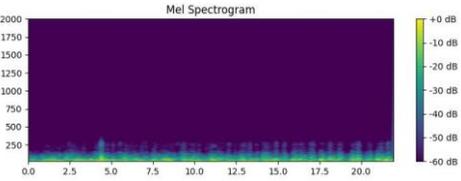
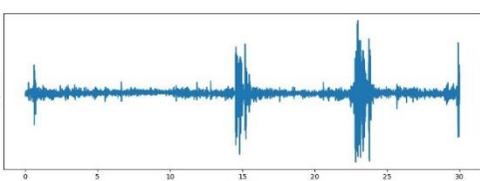
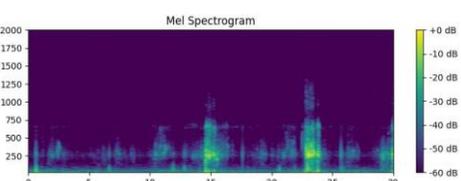
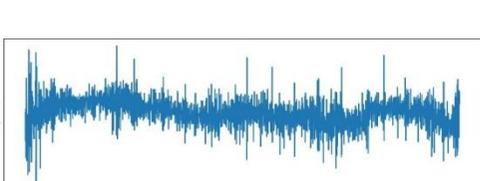
```

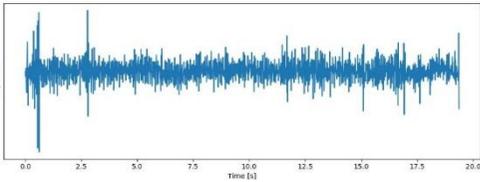
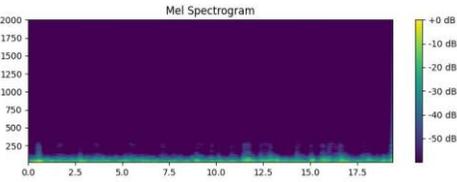
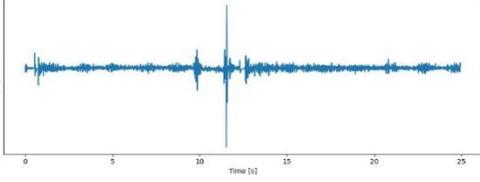
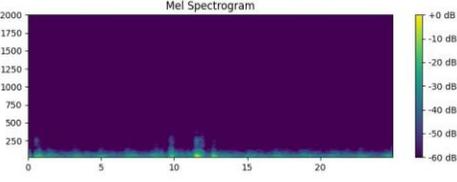
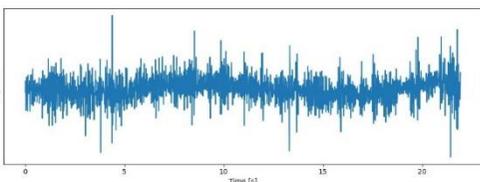
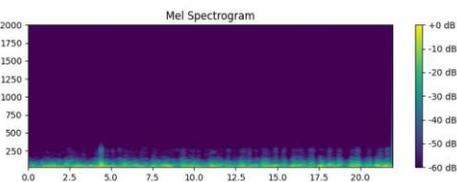
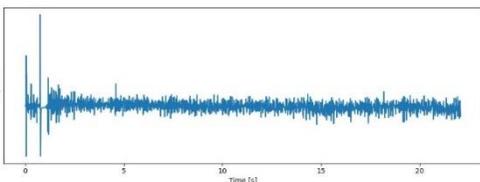
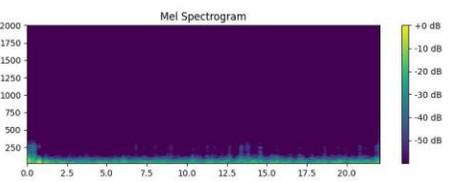
```

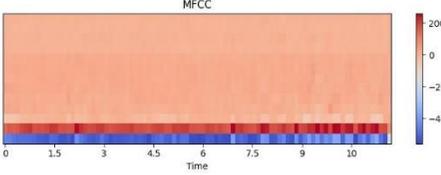
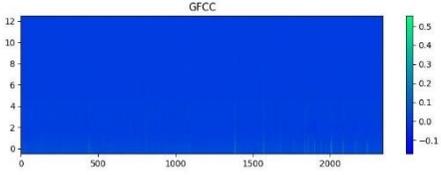
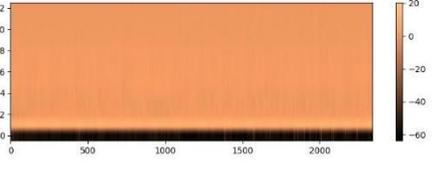
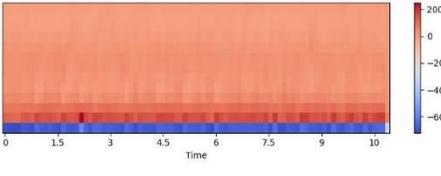
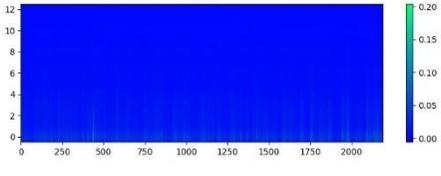
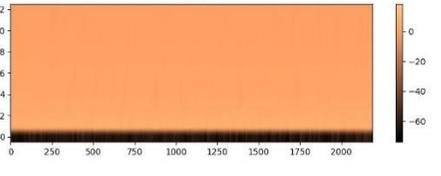
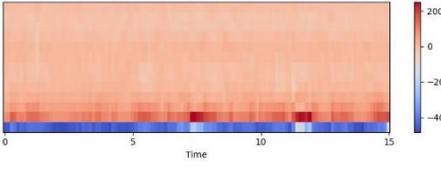
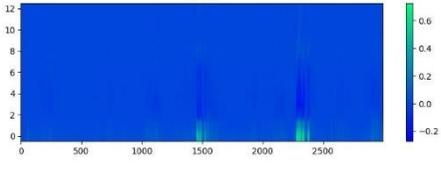
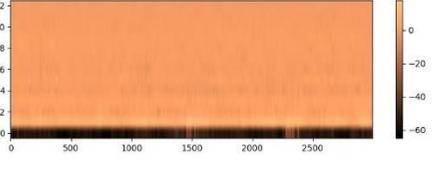
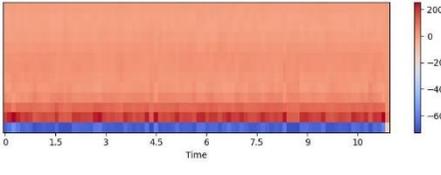
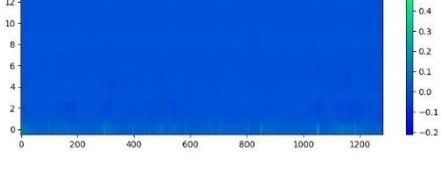
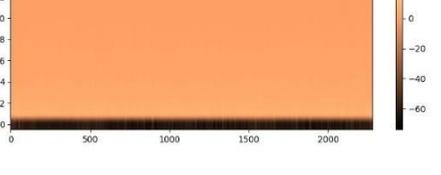
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(300, 800, 3),
kernel_regularizer=l2(0.001)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.5)) # 提高 Dropout 比例
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_regularizer=l2(0.001)))
model.add(Dropout(0.5)) # 提高 Dropout 比例
model.add(Dense(8, activation='softmax'))
# 使用已經定義好帶有學習速率衰減的優化器
optimizer = Adam(learning_rate=lr_schedule)
optimizer.build(model.trainable_variables) # 添加這行代碼
model.compile(loss='categorical_crossentropy', optimizer=optimizer,
              metrics=['accuracy', Precision(name='precision'), Recall(name='recall')])
return model
# 創建模型
model = cnn_model()
print(model.summary())
# 早停策略
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, restore_best_weights=True)
# 學習率衰減策略
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-7, verbose=1)
# 使用整個訓練集進行預訓練
print("進行模型的預訓練...")
history_pretrain = model.fit(
    X_train, y_train,
    epochs=100,
    batch_size=16,
    validation_data=(X_valid, y_valid),
    callbacks=[early_stop, reduce_lr] # 在這裡添加 reduce_lr 回調
)

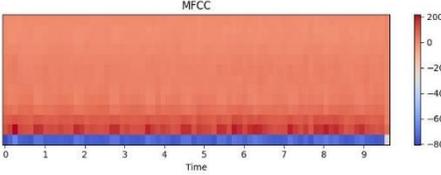
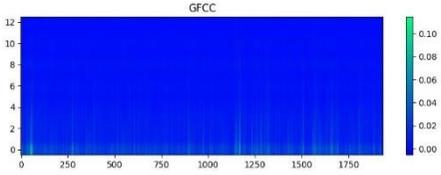
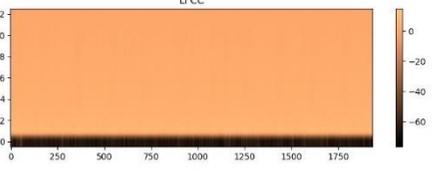
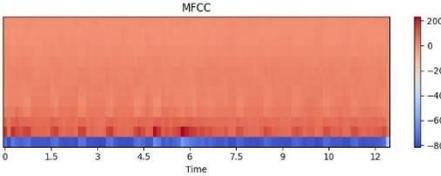
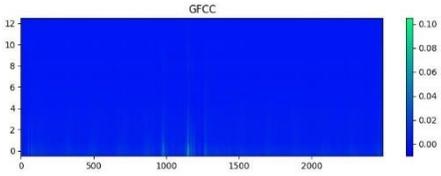
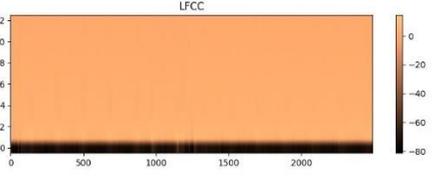
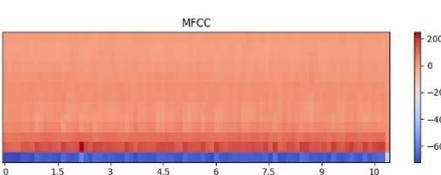
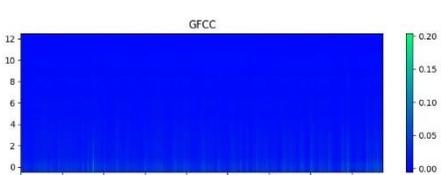
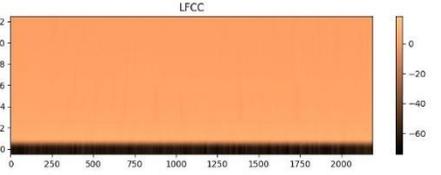
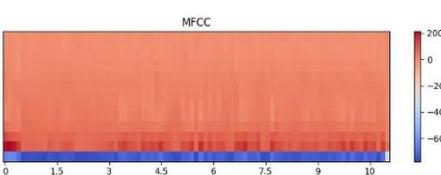
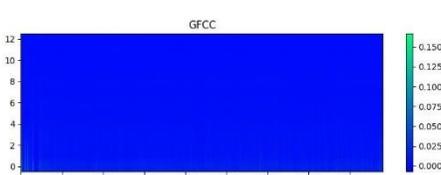
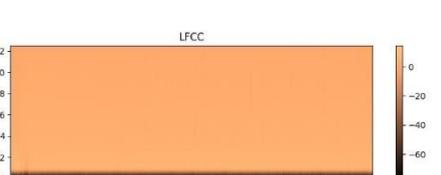
```

附錄五：診斷分類圖檔範例

聲音	Waveform	Mel 頻譜圖
BP75_N,N,PL U,73,F.wav Normal(0.jpg)		
BP97_Asthma,EW,PRU,24,M.wav Asthma,(303.jpg)		
DP36_pneumonia,Crep,P RM,36,F.wav Pneumonia(69.jpg)		
BP108_COPD,EW,PRL,63,M.wav COPD(304.jpg)		

聲音	Waveform	Mel 頻譜圖
BP21_BRON,Crep,PR L ,20,M.wav BRON(318.jpg)		
BP4_Heart FailureI C,P L R,72,F.wav Heart Failure (273.jpg)		
BP23 Lung Fibrosis BP97_Asthma,EW,PRU,24,M.wav (303.jpg)		
BP8_Plueral Effusion,I C B,P L L,70,M.wav Plueral Effusion (208.jpg)		

聲音	MFCCs	GFCCs	LFCCs
BP75_N,N,PL U,73,F.wav Normal(0.jpg)			
BP97_Asthma,EW,PRU,24,M .wav Asthma,(303.jpg)			
DP36_pneumonia,Crep,P RM,36,F.wav Pneumonia(69.jpg)			
BP108_COPD,EW,PRL,63,M. wav COPD(304.jpg)			

聲音	MFCCs	GFCCs	LFCCs
BP21_BRON,Crep,PR L ,20,M.wav BRON(318.jpg)			
BP4_Heart FailureI C,P L R,72,F.wav Heart Failure (273.jpg)			
BP23 Lung Fibrosis BP97_Asthma,EW,PRU,24,M .wav (303.jpg)			
BP8_Plueral Effusion,I C B,P L L,70,M.wav Plueral Effusion (208.jpg)			

附錄六【新型申請專利範圍】：

- 1.一種具肺音圖像智慧解碼功能的個人健康監控設備，係包含有：一監控主機，係內設有一應用處理器、一記憶單元、一通信模組、一語音模組、一顯示單元及一傳輸介面，其中該記憶單元、該通信模組、該語音模組、該顯示單元及該傳輸介面均電性連接該應用處理器；其中該記憶單元預存由該監控主機外部設備所輸入至少一肺音音訊電子檔；及一肺音圖像智慧解碼應用軟體，係更包含有一使用者介面並預存於該記憶單元內，且該肺音圖像智慧解碼應用軟體提供對該至少一肺音音訊電子檔進行自動智慧解碼並轉換為至少一圖像，以利判讀患病或健康。
- 2.如請求項 1 所述具肺音圖像智慧解碼功能的個人健康監控設備，其中係藉由該監控主機的應用處理器呼叫該肺音圖像智慧解碼應用軟體並執行之，使該使用者介面輸出於該監控主機的顯示單元以提供用戶進行操作與參數設定，透過該使用者介面之操作，進而藉由該肺音圖像智慧解碼應用軟體所設一 AI 解碼轉換模組將該至少一肺音音訊電子檔進行解碼並轉換為該至少一圖像，再將該至少一肺音音訊電子檔與該至少一圖像輸出於該監控主機的顯示單元。
- 3.如請求項 2 所述具肺音圖像智慧解碼功能的個人健康監控設備，其中另包含一肺音採集器，且該肺音採集器係以有線或無線連接於該監控主機，並提供置放於人體的胸壁以採集肺音，且將所採集到的肺音轉換成該至少一肺音音訊電子檔來預存於該監控主機所設該記憶單元內。
- 4.如請求項 3 所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該 AI 解碼轉換模組係採用卷積神經網路模型，將涵蓋哮喘、心臟衰竭、肺炎、支氣管炎、胸腔積液、肺纖維化、慢性阻塞肺疾病、健康的肺音音訊、圖像、標籤之資料作成訓練集與驗證集，並以精準度、敏感度、特异性來作為評估指標，再以 ROC 曲線圖及混淆矩陣圖來進行評估，以對該 AI 解碼轉換模組進行訓練及建立。
- 5.如請求項 4 所述具肺音圖像智慧解碼功能的個人健康監控設備，其中提供判讀患病或健康之圖像係更包含有一波形圖、一梅爾頻率倒譜係數圖、一梅爾頻譜或其組合。
- 6.如請求項 3 所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該肺音圖像智慧解碼應用軟體被執行時，令該語音模組、該顯示單元提供詳細的指引和圖示，幫助用戶了解應將該肺音採集器應放在胸壁的哪些位置上。
- 7.如請求項 6 所述具肺音圖像智慧解碼功能的個人健康監控設備，在其進行檢測時，

該肺音圖像智慧解碼應用軟體可以提供語音或文字之提示，以幫助用戶確保正確地進行檢測。

8.如請求項3所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該肺音圖像智慧解碼應用軟體具有錄音和重播功能，用戶可以錄下其呼吸聲音、肺音或其組合，並有選擇地以當面或遠距方式重播給醫生聽。

9.如請求項3所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該肺音圖像智慧解碼應用軟體提供分析用戶的呼吸聲音、肺音或其組合，並根據預設的病因分析對照表給出初步分析或建議。

10.如請求項3所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該肺音圖像智慧解碼應用軟體令該監控主機的通信模組，提供用戶可以選擇地將檢測結果同步上傳至雲端伺服器，並與用戶的醫療提供者共享以便於遠程診斷和健康管理。

11.如請求項3所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該肺音圖像智慧解碼應用軟體的用戶介面係更包含有一般、一捻發聲、一哮鳴聲、一爆裂聲、一支氣管聲之聲音類型選項，以及一般、一支氣管肺炎、一慢性阻塞性肺疾病、一心臟衰竭、一肺纖維化、一哮喘、一胸腔積水之疾病判別選項。

12.如請求項1所述具肺音圖像智慧解碼功能的個人健康監控設備，其中該監控主機設為移動通信設備或桌上型電腦，而該移動通信設備係指智慧型手機、平板電腦或筆記型電腦。

13.一種具肺音圖像智慧解碼功能的個人健康監控系統，係使用請求項1至12任一項所述具肺音圖像智慧解碼功能的個人健康監控設備，其包含有：數台具肺音圖像智慧解碼功能的個人健康監控設備，係在位於不同場所且具上網功能的數個桌上型電腦、數個筆記型電腦、數個智慧型手機、數個平板電腦或其組合之數個監控主機，並在該數個監控主機分別安裝有一肺音圖像智慧解碼應用軟體與一肺音採集器，以藉由各肺音圖像智慧解碼應用軟體所設一AI解碼轉換模組將相應各肺音採集器所採集到的各肺音音訊電子檔進行解碼並轉換為至少一圖像，再將其上傳到一雲端伺服器；及該雲端伺服器，係接收用戶端的該數台具肺音圖像智慧解碼功能的個人健康監控設備所上傳的各肺音音訊電子檔及其經解碼轉換後的該至少一圖像，與用戶的醫療提供者共享以便於遠程診斷和健康管理。

【具肺音圖像智慧解碼功能的個人健康監控系統及其設備】

A:個人健康監控設備

1:監控主機

11:應用處理器

12:記憶單元

13:通信模組

14:語音模組

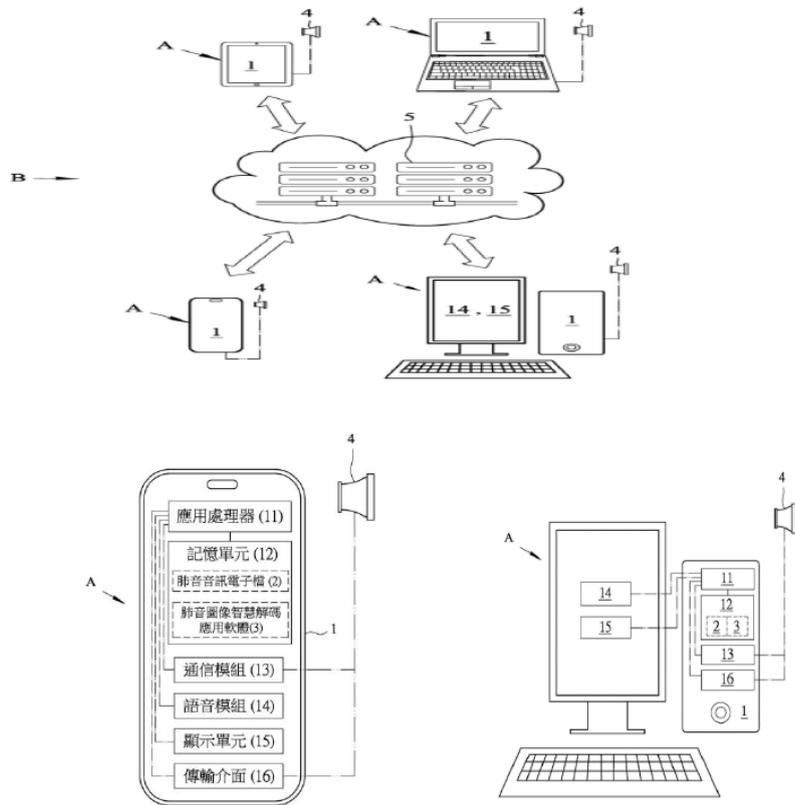
15:顯示單元

16:傳輸介面

2:肺音音訊電子檔

3:肺音圖像智慧解碼應用軟體

4. 肺音採集器



該系統包括以下組成部分：

1. 監控主機：內部設有應用處理器、記憶單元、通信模組、語音模組、顯示單元和傳輸介面。

- 記憶單元：預存至少一個肺音音訊電子檔。
- 肺音圖像智慧解碼應用軟體：包含使用者介面和 AI 解碼轉換模組，預存於記憶單元中。

2. 功能描述：

- 肺音圖像智慧解碼應用軟體能對預存的肺音音訊電子檔進行自動智慧解碼，並轉換為圖像，方便用戶進行健康或疾病的判讀。

附錄七：由圖形轉回聲音程式碼

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from scipy.io import wavfile
from IPython import display

# 定義繪製頻譜圖的函數
def plot_spectrogram(spectrogram) :
    if len(spectrogram.shape) > 2 :
        assert len(spectrogram.shape) == 3
        spectrogram = np.squeeze(spectrogram, axis=-1)

    plt.figure(figsize=(12, 8))
    # 使用 Purples 色彩映射
    plt.imshow(np.log(spectrogram.T + 1e-9), aspect='auto', origin='lower', extent=[0, 16000, 0, 8000],
cmap='Purples')
    plt.xlim(0, 16000)
    plt.ylim(0, 8000)
    plt.title('Spectrogram')
    plt.ylabel('Frequency [Hz]')
    plt.xlabel('Time [s]')
    plt.show()

# 定義讀取 wav 檔案的函數
def read_mel(wav_fname) :
    samplerate, data = wavfile.read(wav_fname)
    length = data.shape[0] / samplerate
    return samplerate, data, length

# 定義獲取音訊頻譜圖的函數
def get_spectrogram(melform) :
    input_len = 16000
    melform = melform[ : input_len]
    zero_padding = tf.zeros(input_len - tf.shape(melform), dtype=tf.float32)
```

```

melform = tf.cast(melform, dtype=tf.float32)
equal_length = tf.concat([melform, zero_padding], 0)
spectrogram = tf.signal.stft(equal_length, frame_length=255, frame_step=128)
spectrogram = tf.abs(spectrogram)
spectrogram = spectrogram[..., tf.newaxis]
return spectrogram

# 指定 wav 檔案的路徑
wav_fname = "/content/drive/MyDrive/LungSounds/input/AudioFiles/BP102_N,N,P L L,41,M.wav"

# 讀取 wav 檔案
samplerate, data, length = read_mel(wav_fname)

# 繪製波形圖
time = np.linspace(0., length, data.shape[0])
plt.figure(figsize=(12, 8))
plt.plot(time, data[:, :], label="Left channel")
plt.legend()
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")
import os

directory = "/content/drive/MyDrive/LungSounds/dataset/mel_spectrogram/"
if not os.path.exists(directory) :
    os.makedirs(directory)

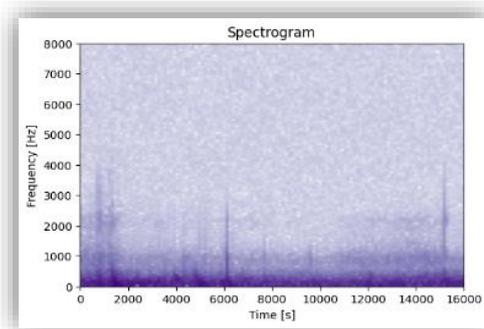
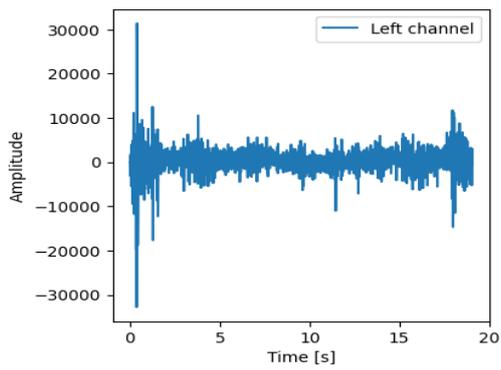
plt.savefig("/content/drive/MyDrive/LungSounds/dataset/output2", facecolor='y', bbox_inches="tight",
pad_inches=0.3, transparent=True)
plt.show()

# 獲取頻譜圖
spectrogram = get_spectrogram(data)
# 使用自定義函數繪製頻譜圖
plot_spectrogram(spectrogram.numpy())

```

```
# 顯示相關資訊
print('melform shape : ', data.shape)
print('Spectrogram shape : ', spectrogram.shape)
print('Audio playback')

# 播放音訊
display.display(display.Audio(data, rate=samplerate))
```



```
melform shape: (76224,)
Spectrogram shape: (124, 129, 1)
Audio playback
```

An audio playback control interface showing a play button, a progress bar at 0:00 / 0:19, a volume icon, and a settings icon.

附錄八：音訊圖檔 mapping

Image File	Audio File
0.jpg	EP75_N,N,P L U,73,F.wav
1.jpg	EP93_N,N,P R M,75,M.wav
2.jpg	EP89_N,N,P L M,70,M.wav
3.jpg	EP98_N,N,P L L,53,M.wav
4.jpg	EP60_heart failure,Crep,P L L,83,F.wav
5.jpg	EP97_Asthma,E W,P R U,24,M.wav
6.jpg	EP67_heart failure,Crep,P R L ,24,F.wav
7.jpg	EP64_asthma,E W,P L U,60,M.wav
8.jpg	EP91_asthma,E W,P R U,43,F.wav
9.jpg	EP61_COPD,E W,P R L ,53,M.wav
10.jpg	EP86_N,N,P R U,68,F.wav
11.jpg	EP76_N,N,A L U,31,M.wav
12.jpg	EP96_N,N,P R L ,66,F.wav
13.jpg	EP71_N,N,P R U,36,M.wav
14.jpg	EP62_COPD,E W,P L M,53,M.wav
15.jpg	EP81_N,N,P L U,33,M.wav
16.jpg	EP92_N,N,P L L ,43,M.wav
17.jpg	EP84_N,N,A R U,65,F.wav
18.jpg	DP35_Asthma,E W,A R L,38,M.wav
19.jpg	EP68_asthma,E W,P R L ,21,F.wav
20.jpg	EP6_Plueral Effusion,I C B,P L R,81,M.wav
21.jpg	EP77_Asthma,E W,A R L,40,M.wav
22.jpg	EP80_asthma,E W,A L U,38,M.wav
23.jpg	EP90_heart failure,Crep,P L M,74,F.wav
24.jpg	EP65_asthma,E W,P R L ,60,F.wav
25.jpg	EP87_N,N,P R M,72,M.wav
26.jpg	EP88_N,N,A R U,29,M.wav
27.jpg	EP94_N,N,P L L,24,M.wav
28.jpg	EP78_Asthma,E W,P R M,20,M.wav
29.jpg	EP72_N,N,P R U,24,F.wav
30.jpg	EP83_N,N,A R U,18,F.wav
31.jpg	EP74_N,N,P R M,74,M.wav
32.jpg	EP99_N,N,P R M,50,M.wav
33.jpg	EP79_asthma,E W,A R U,46,M.wav
34.jpg	EP82_N,N,P L U,38,M.wav
35.jpg	EP8_Plueral Effusion,I C B,P L L,70,M.wav
36.jpg	EP69_pneumonia,Bronchial,P R L ,64,M.wav
37.jpg	EP7_Heart Failure + COPD,I C E W,P L R,65,M.wav

38.jpg	EP66_heart failure,Crep,P R L ,43,M.wav
39.jpg	EP9_Asthma,E W,P R L ,59,M.wav
40.jpg	DP58_asthma,E W,P L M,40,M.wav
41.jpg	DP43_asthma,E W,P L M,15,M.wav
42.jpg	EP85_N,N,A R U,33,M.wav
43.jpg	DP57_COPD,E W,P R L ,42,M.wav
44.jpg	EP51_N,N,P L M,30,M.wav
45.jpg	EP73_N,N,P L L,32,M.wav
46.jpg	EP106_Asthma,E W,P L U,45,F.wav
47.jpg	DP32_N,N,A R L,30,M.wav
48.jpg	DP46_asthma,E W,P L U,41,F.wav
49.jpg	DP44_asthma,E W,P R U,40,M.wav
50.jpg	EP95_N,N,P L M,18,M.wav
51.jpg	DP76_N,N,A L U,31,M.wav
52.jpg	EP12_Asthma,E W,P L L,50,F.wav
53.jpg	EP24_heart failure,Crep,P R L ,76,F.wav
54.jpg	EP50_N,N,P R L ,27,M.wav
55.jpg	EP48_N,N,P L U,31,M.wav
56.jpg	EP43_asthma,E W,P L M,15,M.wav
57.jpg	EP70_N,N,P L U,52,F.wav
58.jpg	EP109_N,N,P L M,26,M.wav
59.jpg	DP33_Asthma,E W,P R M,43,F.wav
60.jpg	EP27_asthma,E W,P R M,53,F.wav
61.jpg	EP63_COPD,E W,P R L ,58,F.wav
62.jpg	DP40_heart failure,Crep,A R M,26,M.wav
63.jpg	EP25_copd,E W,P L L,76,M.wav
64.jpg	EP102_N,N,P L L,41,M.wav
65.jpg	DP85_N,N,A R U,33,M.wav
66.jpg	EP13_Heart Failure,C,P L L,55,M.wav
67.jpg	EP19_heart failure,C,P R U,70,F.wav
68.jpg	EP29_heart failure,Crep,P L L,57,M.wav
69.jpg	DP36_pneumonia,Crep,P R M,36,F.wav
70.jpg	DP49_N,N,P R M,21,M.wav
71.jpg	EP2_Asthma,E W,P L L R,52,F.wav
72.jpg	DP62_COPD,E W,P L M,53,M.wav
73.jpg	DP95_N,N,P L M,18,M.wav
74.jpg	EP105_Lung Fibrosis,Crep,A U R,44,M.wav
75.jpg	EP110_COPD,E W,P L L,62,M.wav
76.jpg	EP55_asthma,E W,P R U,72,F.wav
77.jpg	EP21_BRON,Crep,P R L ,20,M.wav

78.jpg	DP63_COPD,E W,P R L ,58,F.wav
79.jpg	DP89_N,N,P L M,70,M.wav
80.jpg	DP82_N,N,P L U,38,M.wav
81.jpg	EP11_Heart Failure,C,P L L,53,M.wav
82.jpg	DP78_Asthma,E W,P R M,20,M.wav
83.jpg	EP1_Asthma,I E W,P L L,70,M.wav
84.jpg	EP103_N,N,P R U,81,F.wav
85.jpg	DP99_N,N,P R M,50,M.wav
86.jpg	DP98_N,N,P L L,53,M.wav
87.jpg	DP47_asthma,E W,P L M,60,M.wav
88.jpg	EP14_Heart Failure,C,A R M,54,M.wav
89.jpg	DP38_Asthma,E W,P R M,38,M.wav
90.jpg	DP60_heart failure,Crep,P L L,83,F.wav
91.jpg	DP91_asthma,E W,P R U,43,F.wav
92.jpg	EP107_Asthma,E W,P L U,59,F.wav
93.jpg	EP44_asthma,E W,P R U,40,M.wav
94.jpg	EP17_COPD,E W,A R M,57,M.wav
95.jpg	DP88_N,N,A R U,29,M.wav
96.jpg	EP57_COPD,E W,P R L ,42,M.wav
97.jpg	DP68_asthma,E W,P R L ,21,F.wav
98.jpg	DP54_heart failure,Crep,P R L ,73,F.wav
99.jpg	DP80_asthma,E W,A L U,38,M.wav
100.jpg	DP74_N,N,P R M,74,M.wav
101.jpg	DP70_N,N,P L U,52,F.wav
102.jpg	DP75_N,N,P L U,73,F.wav
103.jpg	DP64_asthma,E W,P L U,60,M.wav
104.jpg	EP58_asthma,E W,P L M,40,M.wav
105.jpg	EP30_N,N,P R M,18,F.wav
106.jpg	DP72_N,N,P R U,24,F.wav
107.jpg	DP52_Lung Fibrosis,Crep,P L L,76,F.wav
108.jpg	DP66_heart failure,Crep,P R L ,43,M.wav
109.jpg	EP41_pneumonia,Crep,P R M,51,M.wav
110.jpg	DP3_Asthma,I E W,P L L R,50,F.wav
111.jpg	EP49_N,N,P R M,21,M.wav
112.jpg	EP22_BRON,Crep,P R U,24,M.wav
113.jpg	DP6_Plueral Effusion,I C B,P L R,81,M.wav
114.jpg	EP3_Asthma,I E W,P L L R,50,F.wav
115.jpg	DP69_pneumonia,Bronchial,P R L ,64,M.wav
116.jpg	DP61_COPD,E W,P R L ,53,M.wav
117.jpg	EP36_pneumonia,Crep,P R M,36,F.wav

118.jpg	DP79_asthma,E W,A R U,46,M.wav
119.jpg	DP97_Asthma,E W,P R U,24,M.wav
120.jpg	EP20_Asthma and lung fibrosis,C,A R M,90,M.wav
121.jpg	DP55_asthma,E W,P R U,72,F.wav
122.jpg	DP83_N,N,A R U,18,F.wav
123.jpg	EP35_Asthma,E W,A R L,38,M.wav
124.jpg	EP42_heart failure,Crep,P L L,56,M.wav
125.jpg	EP5_Heart Failure + COPD,I C E W,P L R ,71,M.wav
126.jpg	EP56_asthma,E W,P R M,56,F.wav
127.jpg	DP5_Heart Failure + COPD,I C E W,P L R ,71,M.wav
128.jpg	DP93_N,N,P R M,75,M.wav
129.jpg	DP8_Plueral Effusion,I C B,P L L,70,M.wav
130.jpg	EP45_heart failure,Crep,P R L ,30,M.wav
131.jpg	EP33_Asthma,E W,P R M,43,F.wav
132.jpg	DP73_N,N,P L L,32,M.wav
133.jpg	DP50_N,N,P R L ,27,M.wav
134.jpg	EP34_heart failure,Crep,P R M,78,F.wav
135.jpg	DP77_Asthma,E W,A R L,40,M.wav
136.jpg	DP45_heart failure,Crep,P R L ,30,M.wav
137.jpg	DP48_N,N,P L U,31,M.wav
138.jpg	DP71_N,N,P R U,36,M.wav
139.jpg	DP4_Heart Failure + Lung Fibrosis ,I C,P L R,72,F.wav
140.jpg	DP51_N,N,P L M,30,M.wav
141.jpg	EP18_pneumonia,C,P R U,57,M.wav
142.jpg	EP38_Asthma,E W,P R M,38,M.wav
143.jpg	EP26_Lung Fibrosis,Crep,P,90,F.wav
144.jpg	EP108_COPD,E W,P R L ,63,M.wav
145.jpg	EP39_heart failure,Crep,P R L ,67,M.wav
146.jpg	EP15_Asthma,E W,P R U,49,F.wav
147.jpg	EP16_heart failure,C,P R U,56,M.wav
148.jpg	EP54_heart failure,Crep,P R L ,73,F.wav
149.jpg	EP46_asthma,E W,P L U,41,F.wav
150.jpg	EP47_asthma,E W,P L M,60,M.wav
151.jpg	EP104_Asthma,E W,P L U,45,F.wav
152.jpg	DP86_N,N,P R U,68,F.wav
153.jpg	DP81_N,N,P L U,33,M.wav
154.jpg	EP52_Lung Fibrosis,Crep,P L L,76,F.wav
155.jpg	EP112_N,N,P L M,30,M.wav
156.jpg	DP41_pneumonia,Crep,P R M,51,M.wav
157.jpg	DP59_heart failure,Crep,P R L ,83,F.wav

158.jpg	DP65_asthma,E W,P R L ,60,F.wav
159.jpg	EP59_heart failure,Crep,P R L ,83,F.wav
160.jpg	DP94_N,N,P L L,24,M.wav
161.jpg	DP42_heart failure,Crep,P L L,56,M.wav
162.jpg	EP31_N,N,P L M,27,F.wav
163.jpg	EP23_Lung Fibrosis,Crep,P R L ,50,M.wav
164.jpg	DP87_N,N,P R M,72,M.wav
165.jpg	DP39_heart failure,Crep,P R L ,67,M.wav
166.jpg	EP4_Heart Failure + Lung Fibrosis ,I C,P L R,72,F.wav
167.jpg	DP9_Asthma,E W,P R L ,59,M.wav
168.jpg	EP101_Asthma,E W,P L M,12,F.wav
169.jpg	DP67_heart failure,Crep,P R L ,24,F.wav
170.jpg	DP53_asthma,E W,P R M,72,F.wav
171.jpg	EP53_asthma,E W,P R M,72,F.wav
172.jpg	EP40_heart failure,Crep,A R M,26,M.wav
173.jpg	EP32_N,N,A R L,30,M.wav
174.jpg	DP7_Heart Failure + COPD,I C E W,P L R,65,M.wav
175.jpg	DP34_heart failure,Crep,P R M,78,F.wav
176.jpg	DP90_heart failure,Crep,P L M,74,F.wav
177.jpg	DP56_asthma,E W,P R M,56,F.wav
178.jpg	EP37_pneumonia,Crep,A R L,70,F.wav
179.jpg	DP84_N,N,A R U,65,F.wav
180.jpg	DP37_pneumonia,Crep,A R L,70,F.wav
181.jpg	EP111_COPD,E W,P R L ,51,M.wav
182.jpg	BP83_N,N,A R U,18,F.wav
183.jpg	EP100_N,N,P R M,70,F.wav
184.jpg	BP93_N,N,P R M,75,M.wav
185.jpg	BP27_asthma,E W,P R M,53,F.wav
186.jpg	EP28_BRON,Crep,P L U,68,F.wav
187.jpg	BP76_N,N,A L U,31,M.wav
188.jpg	DP96_N,N,P R L ,66,F.wav
189.jpg	BP44_asthma,E W,P R U,40,M.wav
190.jpg	BP10_Asthma,E W,P R U,59,M.wav
191.jpg	DP31_N,N,P L M,27,F.wav
192.jpg	BP58_asthma,E W,P L M,40,M.wav
193.jpg	EP10_Asthma,E W,P R U,59,M.wav
194.jpg	BP87_N,N,P R M,72,M.wav
195.jpg	BP62_COPD,E W,P L M,53,M.wav
196.jpg	BP77_Asthma,E W,A R L,40,M.wav
197.jpg	BP42_heart failure,Crep,P L L,56,M.wav

198.jpg	BP66_heart failure,Crep,P R L ,43,M.wav
199.jpg	BP32_N,N,A R L,30,M.wav
200.jpg	BP12_Asthma,E W,P L L,50,F.wav
201.jpg	BP55_asthma,E W,P R U,72,F.wav
202.jpg	DP20_Asthma and lung fibrosis,C,A R M,90,M.wav
203.jpg	BP79_asthma,E W,A R U,46,M.wav
204.jpg	DP10_Asthma,E W,P R U,59,M.wav
205.jpg	BP86_N,N,P R U,68,F.wav
206.jpg	DP111_COPD,E W,P R L ,51,M.wav
207.jpg	DP92_N,N,P L L ,43,M.wav
208.jpg	BP8_Plueral Effusion,I C B,P L L,70,M.wav
209.jpg	BP81_N,N,P L U,33,M.wav
210.jpg	BP73_N,N,P L L,32,M.wav
211.jpg	BP99_N,N,P R M,50,M.wav
212.jpg	BP112_N,N,P L M,30,M.wav
213.jpg	DP2_Asthma,E W,P L L R,52,F.wav
214.jpg	BP69_pneumonia,Bronchial,P R L ,64,M.wav
215.jpg	BP57_COPD,E W,P R L ,42,M.wav
216.jpg	BP35_Asthma,E W,A R L,38,M.wav
217.jpg	BP38_Asthma,E W,P R M,38,M.wav
218.jpg	BP2_Asthma,E W,P L L R,52,F.wav
219.jpg	BP23_Lung Fibrosis,Crep,P R L ,50,M.wav
220.jpg	BP3_Asthma,I E W,P L L R,50,F.wav
221.jpg	DP14_Heart Failure,C,A R M,54,M.wav
222.jpg	BP60_heart failure,Crep,P L L,83,F.wav
223.jpg	BP18_pneumonia,C,P R U,57,M.wav
224.jpg	BP28_BRON,Crep,P L U,68,F.wav
225.jpg	BP94_N,N,P L L,24,M.wav
226.jpg	BP95_N,N,P L M,18,M.wav
227.jpg	BP7_Heart Failure + COPD,I C E W,P L R,65,M.wav
228.jpg	DP18_pneumonia,C,P R U,57,M.wav
229.jpg	BP85_N,N,A R U,33,M.wav
230.jpg	BP17_COPD,E W,A R M,57,M.wav
231.jpg	DP19_heart failure,C,P R U,70,F.wav
232.jpg	DP26_Lung Fibrosis,Crep,P,90,F.wav
233.jpg	BP52_Lung Fibrosis,Crep,P L L,76,F.wav
234.jpg	DP101_Asthma,E W,P L M,12,F.wav
235.jpg	DP107_Asthma,E W,P L U,59,F.wav
236.jpg	BP31_N,N,P L M,27,F.wav
237.jpg	BP110_COPD,E W,P L L,62,M.wav

238.jpg	BP92_N,N,P L L ,43,M.wav
239.jpg	BP29_heart failure,Crep,P L L,57,M.wav
240.jpg	BP68_asthma,E W,P R L ,21,F.wav
241.jpg	BP111_COPD,E W,P R L ,51,M.wav
242.jpg	BP98_N,N,P L L,53,M.wav
243.jpg	BP25_copd,E W,P L L,76,M.wav
244.jpg	DP11_Heart Failure,C,P L L,53,M.wav
245.jpg	BP26_Lung Fibrosis,Crep,P,90,F.wav
246.jpg	BP49_N,N,P R M,21,M.wav
247.jpg	BP41_pneumonia,Crep,P R M,51,M.wav
248.jpg	BP1_Asthma,I E W,P L L,70,M.wav
249.jpg	BP64_asthma,E W,P L U,60,M.wav
250.jpg	DP30_N,N,P R M,18,F.wav
251.jpg	BP88_N,N,A R U,29,M.wav
252.jpg	DP12_Asthma,E W,P L L,50,F.wav
253.jpg	BP71_N,N,P R U,36,M.wav
254.jpg	BP11_Heart Failure,C,P L L,53,M.wav
255.jpg	DP112_N,N,P L M,30,M.wav
256.jpg	BP20_Asthma and lung fibrosis,C,A R M,90,M.wav
257.jpg	BP89_N,N,P L M,70,M.wav
258.jpg	BP46_asthma,E W,P L U,41,F.wav
259.jpg	BP24_heart failure,Crep,P R L ,76,F.wav
260.jpg	BP14_Heart Failure,C,A R M,54,M.wav
261.jpg	BP90_heart failure,Crep,P L M,74,F.wav
262.jpg	DP100_N,N,P R M,70,F.wav
263.jpg	BP43_asthma,E W,P L M,15,M.wav
264.jpg	BP91_asthma,E W,P R U,43,F.wav
265.jpg	BP61_COPD,E W,P R L ,53,M.wav
266.jpg	DP104_Asthma,E W,P L U,45,F.wav
267.jpg	BP72_N,N,P R U,24,F.wav
268.jpg	BP37_pneumonia,Crep,A R L,70,F.wav
269.jpg	DP29_heart failure,Crep,P L L,57,M.wav
270.jpg	BP47_asthma,E W,P L M,60,M.wav
271.jpg	DP106_Asthma,E W,P L U,45,F.wav
272.jpg	DP24_heart failure,Crep,P R L ,76,F.wav
273.jpg	BP4_Heart Failure + Lung Fibrosis ,I C,P L R,72,F.wav
274.jpg	DP17_COPD,E W,A R M,57,M.wav
275.jpg	BP9_Asthma,E W,P R L ,59,M.wav
276.jpg	BP30_N,N,P R M,18,F.wav
277.jpg	BP16_heart failure,C,P R U,56,M.wav

278.jpg	BP45_heart failure,Crep,P R L ,30,M.wav
279.jpg	BP6_Plueral Effusion,I C B,P L R,81,M.wav
280.jpg	DP21_BRON,Crep,P R L ,20,M.wav
281.jpg	DP102_N,N,P L L,41,M.wav
282.jpg	BP15_Asthma,E W,P R U,49,F.wav
283.jpg	BP96_N,N,P R L ,66,F.wav
284.jpg	BP40_heart failure,Crep,A R M,26,M.wav
285.jpg	DP13_Heart Failure,C,P L L,55,M.wav
286.jpg	DP16_heart failure,C,P R U,56,M.wav
287.jpg	DP23_Lung Fibrosis,Crep,P R L ,50,M.wav
288.jpg	DP28_BRON,Crep,P L U,68,F.wav
289.jpg	DP105_Lung Fibrosis,Crep,A U R,44,M.wav
290.jpg	BP53_asthma,E W,P R M,72,F.wav
291.jpg	BP54_heart failure,Crep,P R L ,73,F.wav
292.jpg	BP39_heart failure,Crep,P R L ,67,M.wav
293.jpg	DP25_copd,E W,P L L,76,M.wav
294.jpg	BP36_pneumonia,Crep,P R M,36,F.wav
295.jpg	BP56_asthma,E W,P R M,56,F.wav
296.jpg	BP65_asthma,E W,P R L ,60,F.wav
297.jpg	BP13_Heart Failure,C,P L L,55,M.wav
298.jpg	BP78_Asthma,E W,P R M,20,M.wav
299.jpg	BP19_heart failure,C,P R U,70,F.wav
300.jpg	BP82_N,N,P L U,38,M.wav
301.jpg	BP75_N,N,P L U,73,F.wav
302.jpg	BP22_BRON,Crep,P R U,24,M.wav
303.jpg	BP97_Asthma,E W,P R U,24,M.wav
304.jpg	BP108_COPD,E W,P R L ,63,M.wav
305.jpg	DP22_BRON,Crep,P R U,24,M.wav
306.jpg	BP59_heart failure,Crep,P R L ,83,F.wav
307.jpg	DP103_N,N,P R U,81,F.wav
308.jpg	BP33_Asthma,E W,P R M,43,F.wav
309.jpg	BP103_N,N,P R U,81,F.wav
310.jpg	BP67_heart failure,Crep,P R L ,24,F.wav
311.jpg	BP5_Heart Failure + COPD,I C E W,P L R ,71,M.wav
312.jpg	DP15_Asthma,E W,P R U,49,F.wav
313.jpg	DP27_asthma,E W,P R M,53,F.wav
314.jpg	BP84_N,N,A R U,65,F.wav
315.jpg	DP109_N,N,P L M,26,M.wav
316.jpg	BP51_N,N,P L M,30,M.wav
317.jpg	DP1_Asthma,I E W,P L L,70,M.wav

318.jpg	BP21_BRON,Crep,P R L ,20,M.wav
319.jpg	BP34_heart failure,Crep,P R M,78,F.wav
320.jpg	BP48_N,N,P L U,31,M.wav
321.jpg	BP106_Asthma,E W,P L U,45,F.wav
322.jpg	BP104_Asthma,E W,P L U,45,F.wav
323.jpg	BP105_Lung Fibrosis,Crep,A U R,44,M.wav
324.jpg	BP107_Asthma,E W,P L U,59,F.wav
325.jpg	BP102_N,N,P L L,41,M.wav
326.jpg	BP101_Asthma,E W,P L M,12,F.wav
327.jpg	BP109_N,N,P L M,26,M.wav
328.jpg	BP80_asthma,E W,A L U,38,M.wav
329.jpg	DP110_COPD,E W,P L L,62,M.wav
330.jpg	DP108_COPD,E W,P R L ,63,M.wav
331.jpg	BP50_N,N,P R L ,27,M.wav
332.jpg	BP63_COPD,E W,P R L ,58,F.wav
333.jpg	BP100_N,N,P R M,70,F.wav
334.jpg	BP74_N,N,P R M,74,M.wav
335.jpg	BP70_N,N,P L U,52,F.wav