



基於可視化頻譜圖應用於音樂曲風分類之研究

Research of Music Genre Classification Based on Visual
Spectrum

指導教授：鄭煜輝 博士

研究生：張邦慶

中華民國 111 年 9 月



基於可視化頻譜圖應用於音樂曲風分類之研究

Research of Music Genre Classification Based on Visual Spectrum

指導教授：鄭煜輝 (Yu-Huei Cheng)

研究生：張邦慶 (Pang-Ching Chang)

中華民國 111 年 9 月

September, 2022

摘要

近年來由於音樂製作越來越容易，人們使用電腦就可以輕鬆的製作音樂並上傳至串流媒體，例如 YouTube、Spotify、KKBOX 等，而龐大的串流媒體卻也增加了人們在搜尋音樂的困難，就如同圖書館會依據不同的書本內容進行分類，音樂資料庫也需要依據曲風、使用的樂器、創作者等標籤進行分類。許多研究者提出使用音頻分析結合人工智慧與機器學習的方式進行音樂曲風分類，但僅有少數研究者使用可視化方式進行音樂曲風分類之研究且成效都不佳，因此我們在本文中提出可視化梅爾頻譜與可視化 STFT 頻譜，並與傳統使用音頻訊號處理的方式如：卷積神經網路、卷積循環神經網路、長短期記憶模型等方法進行比較。實驗結果得出了可視化梅爾頻譜獲得 94.5% 的準確率，而可視化 STFT 頻譜則獲得 68.8% 的準確率，並與其他研究者使用傳統音頻分析方法進行比較有著較高的準確率，此實驗結果表明可視化頻譜圖方法的可行性。

關鍵字：人工智慧、梅爾頻譜、短時距傅立葉變換頻譜、卷積神經網路、卷積循環神經網路、長短期記憶模型、可視化頻譜圖

Abstract

In recent years, music production is becoming easier and easier. People can easily make music and upload it to streaming media using computers, such as YouTube, Spotify, KKBOX, etc. However, the huge streaming media also increases the difficulty for people to search for music, just like the library will classify books according to different contents. Music databases also need to be categorized by genre, instrument used, creator, etc. Many researchers have proposed using audio analysis combined with artificial intelligence and machine learning approach to music genre classification, but only a few researchers use visual way music genre classification research and results are poor, so we put forward in this article a visualization Mel frequency spectrum and visualization STFT spectrum, and compared with traditional way using audio signal processing such as: Convolutional Neural Network (CNN), Convolutional and Recurrent Neural Network (CRNN), Long Short-Term Memory Model (LSTM) and other methods were compared. The experimental results show that the accuracy of visualized Mel frequency spectrum is 94.5%, and that of visualized STFT spectrum is 68.8%. Compared with other researchers using traditional audio analysis methods, the experimental results show that the visualized spectrum method is feasible.

Keywords: Artificial intelligence, Mel frequency spectrum, Short-time Fourier transform spectrum, Convolutional neural network, Convolutional and recurrent neural network, Long Short-term memory model, Visual spectrum

致謝

能夠順利完成此篇論文，首先需先謝謝我的指導教授鄭煜輝老師，在就讀研究所並在老師門下學習的這段時間裡老師以他專業的角度與耐心指導我，使我能夠完成這篇論文，非常感謝老師的諄諄教誨。

我也非常感謝詹承祐、蔡潮慶、阮泓凱、陳柏允、彭彥愷、王育維等同學，在我感到徬徨無助時是你們依然陪伴並鼓勵我繼續向前，沒有你們我根本無法完成這篇論文甚至走到這裡。

我也需要感謝王士豪、王遠智、王駿恭、寇竹筠等同學，是你們在我剛入學時完全手足無措時陪伴我，即使最後無法走到最後，但這段時間依然感謝你們的陪伴。

非常感謝我的家人，在我讀碩士期間義無反顧的支持我就讀，就算偶有抱怨也是繼續支持，即使因為想嘗試完成夢想而延畢仍然支持我的選擇，在我需要鼓勵時給我加油打氣，在我需要支持時義無反顧，在我需要解惑時分享他們的人生經驗，甚至在我沒有錢時還會偷偷塞錢到我外套口袋，謝謝我親愛的家人。

最後我想謝謝每個出現在我生命裡的每個人，不論我們是情深意厚又或者理念不合，我都非常感謝你們的出現，謝謝大家。

目錄

摘要	I
Abstract	II
致謝	III
目錄	IV
壹、緒論	1
一、研究動機	1
二、研究目的	2
貳、文獻探討	3
一、音樂曲風	3
二、音頻分析與曲風分類	11
參、研究方法	16
一、使用之資料集介紹	16
二、取樣與資料集建置方法	17
(一) 梅爾頻率倒譜系數	17
(二) 短時距傅立葉變換	23
(三) 可視化梅爾頻譜	24
(四) 可視化短時距傅立葉變換頻譜	28
三、使用的方法	31
(一) 卷積神經網路	31
(二) 卷積循環神經網路	44
(三) 長短期記憶模型	47
(四) YOLOv4	50
四、評分指標	64
(一) 混淆矩陣	64
(二) 精確率	65
(三) 召回率	65
(四) F1 分數	65
(五) 準確率	65
(六) mAP	66
肆、結果與討論	67
一、卷積神經網路實驗結果	67
二、卷積循環神經網路實驗結果	72

三、長短期記憶模型實驗結果	76
四、可視化短時距傅立葉變換實驗結果	80
五、可視化梅爾頻譜實驗結果	102
六、所有方法之結果與優缺點	123
伍、結論	127
陸、未來研究方向	128
參考文獻	130



圖目錄

圖 1. 梅爾刻度示意圖	18
圖 2. 音框化	20
圖 3. 漢明窗	21
圖 4. 可視化梅爾頻譜圖	26
圖 5. 建置可視化梅爾頻譜圖資料集	27
圖 6. 可視化短時距傅立葉變換頻譜圖	29
圖 7. 建置可視化短時距傅立葉變換頻譜圖資料集	30
圖 8. 卷積運算	32
圖 9. Sigmoid 函數圖	33
圖 10. tanh 函數圖	34
圖 11. ReLU 函數圖	35
圖 12. Leaky ReLU 函數圖	36
圖 13. Swish 函數圖	37
圖 14. Mish 函數圖	38
圖 15. 最大池化與平均池化示意圖	40
圖 16. 全連接層示意圖	41
圖 17. Dropout 示意圖	42
圖 18. CNN 架構圖	43
圖 19. RNN 示意圖	45
圖 20. CRNN 示意圖	46
圖 21. 單一 LSTM 示意圖	47
圖 22. 多層 LSTM 示意圖	48
圖 23. LSTM 架構圖	49
圖 24. CBM 與 CBL 架構圖	50
圖 25. Res unit 架構圖	50
圖 26. CSPX 架構圖	51
圖 27. SPP 架構圖	51
圖 28. Darknet53 架構圖	53
圖 29. CBN 架構圖	56
圖 30. Dropblock 示意圖	57
圖 31. Mosaci 資料增強示意圖	57
圖 32. Sigmoid 與 Differential Sigmoid	59
圖 33. Cosine Annealing 函數圖	60
圖 34. Cosine Annealing LR 函數圖	61
圖 35. 修改版 SPP 模組示意圖	62
圖 36. 修改版 SAM 模組示意圖	63

圖 37. 修改版 PAN 與一般 PAN 差異示意圖	63
圖 38. 混淆矩陣示意圖	64
圖 39. CNN 方法訓練與測試的準確率曲線圖	69
圖 40. CNN 方法訓練與測試的損失曲線圖	69
圖 41. CNN 方法之混淆矩陣圖	70
圖 42. CRNN 訓練與測試的準確率曲線圖	73
圖 43. CRNN 訓練與測試的損失曲線圖	73
圖 44. CRNN 方法之混淆矩陣圖	74
圖 45. LSTM 訓練準確率與損失曲線圖	77
圖 46. LSTM 驗證準確率與損失曲線圖	78
圖 47. LSTM 方法之混淆矩陣圖	78
圖 48. 可視化 STFT 頻譜圖方法第一次實驗之 mAP 與損失曲線圖	81
圖 49. 可視化 STFT 頻譜圖方法第一次實驗之混淆矩陣圖	81
圖 50. 可視化 STFT 頻譜圖方法第二次實驗之 mAP 與損失曲線圖	83
圖 51. 可視化 STFT 頻譜圖方法第二次實驗之混淆矩陣圖	83
圖 52. 可視化 STFT 頻譜圖方法第三次實驗之 mAP 與損失曲線圖	85
圖 53. 可視化 STFT 頻譜圖方法第三次實驗之混淆矩陣圖	85
圖 54. 可視化 STFT 頻譜圖方法第四次實驗之 mAP 與損失曲線圖	87
圖 55. 可視化 STFT 頻譜圖方法第四次實驗之混淆矩陣圖	87
圖 56. 可視化 STFT 頻譜圖方法第五次實驗之 mAP 與損失曲線圖	89
圖 57. 可視化 STFT 頻譜圖方法第五次實驗之混淆矩陣圖	89
圖 58. 可視化 STFT 頻譜圖方法第六次實驗之 mAP 與損失曲線圖	91
圖 59. 可視化 STFT 頻譜圖方法第六次實驗之混淆矩陣圖	91
圖 60. 可視化 STFT 頻譜圖方法第七次實驗之 mAP 與損失曲線圖	93
圖 61. 可視化 STFT 頻譜圖方法第七次實驗之混淆矩陣圖	93
圖 62. 可視化 STFT 頻譜圖方法第八次實驗之 mAP 與損失曲線圖	95
圖 63. 可視化 STFT 頻譜圖方法第八次實驗之混淆矩陣圖	95
圖 64. 可視化 STFT 頻譜圖方法第九次實驗之 mAP 與損失曲線圖	97
圖 65. 可視化 STFT 頻譜圖方法第九次實驗之混淆矩陣圖	97
圖 66. 可視化 STFT 頻譜圖方法第十次實驗之 mAP 與損失曲線圖	99
圖 67. 可視化 STFT 頻譜圖方法第十次實驗之混淆矩陣圖	99
圖 68. 可視化梅爾頻譜圖方法第一次實驗之 mAP 與損失曲線圖	103
圖 69. 可視化梅爾頻譜圖方法第一次實驗之混淆矩陣圖	103
圖 70. 可視化梅爾頻譜圖方法第二次實驗之 mAP 與損失曲線圖	105
圖 71. 可視化梅爾頻譜圖方法第二次實驗之混淆矩陣圖	105
圖 72. 可視化梅爾頻譜圖方法第三次實驗之 mAP 與損失曲線圖	106
圖 73. 可視化梅爾頻譜圖方法第三次實驗之混淆矩陣圖	107
圖 74. 可視化梅爾頻譜圖方法第四次實驗之 mAP 與損失曲線圖	108

圖 75. 可視化梅爾頻譜圖方法第四次實驗之混淆矩陣圖	108
圖 76. 可視化梅爾頻譜圖方法第五次實驗之 mAP 與損失曲線圖	110
圖 77. 可視化梅爾頻譜圖方法第五次實驗之混淆矩陣圖	110
圖 78. 可視化梅爾頻譜圖方法第六次實驗之 mAP 與損失曲線圖	112
圖 79. 可視化梅爾頻譜圖方法第六次實驗之混淆矩陣圖	112
圖 80. 可視化梅爾頻譜圖方法第七次實驗之 mAP 與損失曲線圖	114
圖 81. 可視化梅爾頻譜圖方法第七次實驗之混淆矩陣圖	114
圖 82. 可視化梅爾頻譜圖方法第八次實驗之 mAP 與損失曲線圖	116
圖 83. 可視化梅爾頻譜圖方法第八次實驗之混淆矩陣圖	116
圖 84. 可視化梅爾頻譜圖方法第九次實驗之 mAP 與損失曲線圖	118
圖 85. 可視化梅爾頻譜圖方法第九次實驗之混淆矩陣圖	118
圖 86. 可視化梅爾頻譜圖方法第十次實驗之 mAP 與損失曲線圖	120
圖 87. 可視化梅爾頻譜圖方法第十次實驗之混淆矩陣圖	120

表目錄

表 1. GTZAN 資料集	16
表 2. 可視化梅爾頻譜參數設定	25
表 3. 可視化短時距傅立葉變換頻譜參數設定	28
表 4. CNN 方法轉換梅爾頻譜之參數設定	67
表 5. CNN 方法結果統計	70
表 6. CRNN 方法轉換梅爾頻譜之參數設定	72
表 7. CRNN 方法結果統計	74
表 8. LSTM 方法轉換 MFCC 之參數設定	77
表 9. LSTM 方法結果統計	79
表 10. 可視化 STFT 方法轉換之參數設定	80
表 11. 可視化 STFT 頻譜圖方法結果統計	101
表 12. 可視化 STFT 頻譜圖方法 10 次實驗結果	101
表 13. 可視化梅爾頻譜方法轉換之參數設定	102
表 14. 可視化梅爾頻譜圖方法結果統計	122
表 15. 可視化梅爾頻譜圖方法 10 次實驗結果	122
表 16. 各式方法實驗結果統計表	125
表 17. 本研究結果與其他方法之比較	126
表 18. 各音樂資料集相關資訊	129

壹、緒論

一、研究動機

音樂，在當今社會是一種隨手可得釋放壓力的管道[10]，且音樂在現今社會是非常普遍的，有研究顯示音樂具備使緊繃的神經得到舒緩、增加注意力、提升創造力等等功能[8]，因此音樂對於當今高壓的社會是不可或缺的。但近年來科技的飛速發展使得網際網路與電腦得到迅速的發展，電腦的效能相比以前強大許多。在過去製作音樂步驟繁瑣且需要花費大量金錢與時間，從音樂取樣開始，需要某種音色就需要借用甚至購入某種樂器，發行時則需要有唱片公司錄製黑膠。而現代由於電腦的高效能，僅透過電腦的合成器就可以發出大部分的樂器聲音，使得音樂製作的門檻也隨之降低，並透過網際網路將做好的音樂上傳至發行商或是如 YouTube 等的串流媒體。但也因為音樂製作方便且快速，使得串流媒體上擁有龐大的音樂資料庫，如此大的音樂資料庫使得人們在搜尋特定曲風音樂時有著一定的困難性，因此找到一個能夠將音樂資料庫中的音樂進行精準分類的工具在當今社會中是一個重要的議題。

二、研究目的

龐大的音樂資料庫就如同雜亂無章的圖書館一樣。在圖書館內為了使讀者能夠更快的找到所需之書本，因此會依照書本種類進行分類，而音樂資料庫也如圖書館一樣需要進行種類的分類。為了增加音樂資料庫中分類的準確率，我們提出使用深度學習將可視化頻譜圖應用於音樂曲風分類中，並與傳統依靠音訊處理的方法進行研究與分析。由於音樂曲風分類屬於冷門研究，因此近些年針對音樂曲風分類之研究大多是將神經網路架構進行串並連之嘗試、融合其他神經網路、修改分類器等等作法，鮮少針對預處理步驟進行實驗，且近年來機器視覺發展迅速，相較於音樂分類，機器視覺的領域發展更成熟且討論度更高，因此本研究將音樂轉換成可視化的頻譜圖後使用機器視覺的方法進行實驗，以驗證將音樂轉換成可視化方法的可行性，其結果在與其他研究者的方法比較準確率後發現，可視化方法在準確率表現可以獲得很好的成績，進而證明可視化方法的可行性。

貳、文獻探討

一、音樂曲風

藍調(Blues)是一種音樂形式，也是一種音樂曲風[37]。藍調之所以叫藍調是因為最初的藍調音樂帶有憂鬱的主題和聲音。藍調發展至今也不僅只有憂鬱的風格，其中也包含了其他主題與情感抒發。藍調主要由特定和弦、貝斯、不和諧的合聲、切分音、音腔和憂鬱的音符所組成。而藍調使用的樂器則由吉他(通常是電吉他)、鼓、低音提琴(撥弦)、薩克斯風、鋼琴與銅管樂器所組成。在藍調中還會使用稱為微分音的技巧，它是比半音還要小的音程，在鋼琴鍵盤中定義是半音與半音之間的聲音，而在電吉他則是使用金屬滑軌來實現。藍調起源於 19 世紀美國內戰後的南方，由非裔美國人的工作歌與靈歌演變而來，因此可以在多數藍調音樂中聽到突然的叫聲與一群人的喊叫聲。藍調音樂是爵士音樂(Jazz)的前身，但與爵士音樂相比，藍調更注重於一定數量的小節和嚴格的和弦的製作，相較於爵士音樂更加嚴謹。一種快速分辨藍調與爵士的方法是，藍調音樂會以歌手為中心，透過歌手領導喊叫的部分可以區別出藍調音樂，而爵士音樂則有較多的樂器演奏表現，甚至純音樂而無人聲。

古典音樂(Classical)可以說是最古老的音樂[30]，一般認為古典音樂是從教會音樂發展而來，在 15 世紀以前通常以天主教堂的聖詩、彌撒曲等人聲清唱為主。在 15 世紀中葉，皇家貴族漸漸脫離教會控制，因此古典樂不再

是教會的音樂，在當時的文藝復興風氣帶領下複音音樂達到全盛時期，複音音樂增加了不同聲部的概念，以各聲部旋律相近的音樂設計方式增加了音樂的層次感，並發展出日後的和弦規則。且在此時期，作曲家將情感以各種音樂手段表達近似文字表達的方式，使得聲樂與器樂漸漸分離獨立，並興起了器樂作曲的風潮。到了 16 世紀的巴洛克時期以華麗為主，在音樂中添加了大量的裝飾音，表現強烈的節奏、短促而律動。16 世紀後期藝術家、作家和建築師開始對希臘和羅馬的古代文化產生興趣，因此誕生了歌劇。到了 17 世紀進入到古典時代，這時候的音樂回歸單純、明朗、樂句簡短、分句清晰的特點，且奏鳴曲與交響樂也從這個時候出現，著名的作曲家如：巴赫、亨德爾、莫札特、海頓即是這個時期的作曲家。19 世紀進入到浪漫時代，浪漫時代以不拘形式的樂曲與自由的演奏法著稱，並且音樂往往賦予豐富的想像力，受到非現實主義的薰陶。浪漫時代主要人物有貝多芬、華格納、李斯特等人。到了現代，古典音樂從原本的音階及大小調性系統發展到多調性或無調性的音樂，並漸漸跳脫原本古典音樂的框架，使得近代古典音樂有著特殊色彩。

鄉村音樂(Country)可以說是最具有美國民族特色的音樂，他是美國現代流行音樂最早的曲風之一[21][55]。鄉村音樂起源於 20 世紀初生活在南方的美國工人階級，特別是在阿帕拉契山脈。鄉村音樂受到許多音樂曲風的影響，包括英國民謡、凱爾特人和愛爾蘭小提琴歌曲、爵士樂、藍調，因此可

以在鄉村音樂中看到這些曲風的身影。普遍認為鄉村音樂開始流行是在 1920 年由卡特家族所錄製的老式音樂，老式音樂由吉他、曼陀林、自鳴箏、小提琴和班卓琴所組成。到了 1930 年代，在全美 AM 電台播放以穀倉舞 (Barn Dance) 為特色的節目，其中最受歡迎的是一種名為 Grand Ole Opry 的鄉村音樂綜藝節目，當時除了播放老式音樂外，還有一種帶有西部牛仔的風格的西方音樂，這種風格通常有像馬一樣奔跑的節奏和關於牛仔、槍戰、不法分子等特色的歌曲，而西方音樂也在當時隨著西部題材電影一同風靡美國。與此同時，一種稱為西部搖擺樂 (Western Swing) 在德克薩斯州、俄克拉荷馬州、加利福尼亞州流行起來。這種風格的音樂使用了如鋼棒吉他這種樂器來產生足夠響亮的音樂，初期西部搖擺樂是西方音樂與搖擺樂的結合，而後期則是鄉村音樂與節奏藍調 (Rhythm and Blues, R&B) 結合，這時則稱為搖滾樂 (Rock)。早期的搖滾樂與鄉村音樂所使用的樂器和節奏非常相似。鄉村音樂通常以吉他和貝斯為基礎，但與電吉他相比，它通常更喜歡原聲吉他，再加上一些小提琴、滑音管吉他、鋼棒吉他和班卓琴。在早期的鄉村音樂中，不經常使用鼓，而是通過敲打吉他來演奏節奏，而搖滾音樂則以電吉他為基礎，配以電貝斯和鼓的打擊樂，一些樂器如風琴、合成器等相結合。另一種區分鄉村音樂與搖滾音樂的方法是，鄉村音樂較著墨於歌詞，就像講故事一樣，在配上簡單的和弦和節奏，這使得鄉村音樂更易於傳唱[17]。搖滾音樂則多以電吉他的操作為主，與貝斯和鼓的配合。

迪斯可音樂(Disco)是一種快節奏舞曲，在 1970 年代達到顛峰，這種音樂常在夜總會或舞廳播放[40]。迪斯可起源於 1960 年代的地下俱樂部，在那個年代由於戰爭、政治醜聞、幫派暴力、失業和飆升的犯罪率導致許多人到夜總會或舞廳逃避現實。時髦的聲音、迷幻的燈光、反光的迪斯科球和積極的氛圍使得迪斯科越來越受歡迎，然而由於那個時代的社會情緒，種族主義和同性戀恐懼症使得迪斯可一瞬間跌落神壇，廣播電台也逐漸不撥放迪斯可樂曲。但不可否認，迪斯可還是持續影響了流行音樂的發展，這要歸功於它的感染力、可跳舞的節拍。迪斯可節奏是用一種在 4/4 時間的節奏模式，稱為 Four-on-the-floor，這種節奏模式由低音鼓均勻的敲擊每個四分音符，這種如同心跳節奏般的風格搭配上切分音，使得迪斯可更適合跳舞。在前期，迪斯可音樂通常由喇叭、弦樂、長笛和其他各種古典樂器所組成，而到了 1970 年代中期，更多製作人則偏向使用電子樂器與合成器而不使用實體樂器，這也是迪斯可被歸類在電子音樂的列表中。

嘻哈音樂(Hip-hop)是一種具備強烈節奏節拍和說唱的音樂[41]。嘻哈是一種文化，包含了唱片騎師(Disc Jockey, DJ)、說唱(MCing)、霹靂舞(Breaking)或 b-boying)、塗鴉。嘻哈音樂起源於 1970 年代紐約市布朗克斯區，由黑人、拉丁裔和加勒比裔美國青年以 DJ 演奏靈魂音樂和放克音樂為特色，透過各式 breakbeats、轉盤技術、刮擦、基於牙買加的即興說唱等方式，而這些特色被認為是現代嘻哈和說唱音樂的先驅。到了 1980 年代人們將架子鼓、

複雜的取樣、隱喻的說唱歌詞，以及電子音樂特色添加到嘻哈音樂中，使得 80 年代嘻哈音樂傳播至世界各地。大部分嘻哈音樂都具備強烈、有節奏的節拍，它可以是快速而激進的，也可以是緩慢而放鬆的，但在嘻哈歌曲中的大多數節拍都不是簡單的鼓聲，而是一種複雜而多樣的形式。在這些節拍中添加說唱與 Breaks，即為一首嘻哈音樂。

爵士音樂(Jazz)的風格很難定義，因為它衍生了許多的子流派，且即興創作是爵士音樂的關鍵元素之一[23][29]。爵士樂的起源可以追溯到 19 世紀下半葉的新奧爾良。新奧爾良是美國唯一一個允許奴隸擁有鼓的地方，因此在新奧爾良可以聽到來自世界各地的音樂，多年來由於來自西非的奴隸致力於保留他們音樂傳統，使得這些黑人將教堂讚美詩、歐洲古典音樂、奴隸歌曲、田野聖歌和具有古巴風格的節奏混合在一起而形成了爵士樂。大部分爵士音樂由薩克斯風、小號、鋼琴、長號、貝斯、鼓和吉他所組成。在 1920 年代爵士音樂開始傳播並達到顛峰，爵士音樂成為咆嘯的二十年代(Roaring Twenties)的代名詞。隨著經濟大蕭條的結束，爵士樂也漸漸從主導地位慢慢退出，但它對流行文化的影響隨著時間的推移而不斷迴響，音樂也隨著新的風格和子流派的形成而繼續發展，到了 1990 年代爵士音樂不再具有主導風格，而是轉變成個人或是樂隊中的特色。

重金屬音樂(Heavy Metal，簡稱為 Metal)是搖滾音樂的子流派[56][16]，通常具備嚴重失真的吉他、有力的鼓聲、超低範圍的低音音符以及激進或沙

啞的人聲等特色。重金屬音樂起源自 1950 年代的藍調，到了 1970 年代達到流行高峰，且擺脫藍調的特色，添加了龐克(Punk)特色，節奏與鼓聲也越來越快。到了 1980 年代分裂出更多的子流派，使得重金屬音樂越來越小眾。

流行音樂(Pop)是一種模糊的曲風，因為流行音樂會隨著時間與地點不同而有不同的分類依據，例如：日本的流行音樂與美國的流行音樂指的是不同曲風的歌曲、1950 年代的美國流行音樂與 1970 年代的美國流行音樂也是不同的曲風。但可以確定的是，流行音樂是一種民間、藝術和古典音樂風格不同的當代音樂形式。通常來說流行音樂是由電子鍵盤、電鋼琴、電子風琴、鍵盤、吉他、鼓、電吉他、原聲吉他和貝斯等樂器所組成。流行音樂起源於 1950 年代的美國與英國，當時的流行音樂泛指爵士樂、鄉村音樂、咆勃爵士樂(Bebop)、說唱和搖滾樂。到了 1960 年代攜帶式收音機的發展使得流行音樂在歐洲和美國獲得了廣泛的影響，當時的流行音樂以巴洛克流行與泡泡糖流行(Bubblegum)為主流。1970 年代巴洛克與泡泡糖的風格被鄉村音樂、龐克搖滾所取代，期間流行音樂發生的最大革命是流行搖滾音樂的興起。1980 年代由於數位錄音的興起使得流行音樂進一步發展，而合成器、電子音樂等被放入流行音樂中。到了 2000 年代，流行音樂已經不是指某個時代的音樂，而是成為一種曲風，流行音樂是一種複雜而奇妙的音樂流派，其中的可能性可謂無窮無盡[36][38]。

雷鬼音樂(Reggae)是早期牙買加的流行音樂之一，並在 1960 年代迅速成為牙買加的主導音樂[42]。雷鬼音樂融合了許多音樂曲風，包括了門托音樂(Mento)、斯卡音樂(Ska)、慢拍搖滾(RockSteady)、卡呂普索音樂(Calypso)、美國靈魂音樂、R&B 等曲風。雷鬼音樂以漸近式打擊樂、催眠式低音線與穩定向上的節奏吉他(Skank Beat)所聞名，且大多數雷鬼音樂都是由牙買加英語、牙買加方言或艾雅爾方言為歌詞。雷鬼音樂通常由鼓、低音吉他、電吉他、鋼琴鍵盤、合成器喇叭所組成。作為牙買加的主導音樂，雷鬼音樂歌詞多與拉斯塔法里運動和社會運動有著密切相關，並為城市底層的人用來表達他們抗議的方法。作為現代音樂發展的始祖，雷鬼音樂影響了包括迪斯可、流行音樂、嘻哈音樂等音樂[25][39][44]。

搖滾音樂(Rock)是全世界最廣為人知的音樂類型，搖滾音樂起源於 1940 年代後期，它融合了古典樂、爵士、藍調、節奏藍調、民謡及鄉村音樂[15][60]。當時由於二戰結束，作為戰後的放縱與新的消費能力出現，搖滾音樂成為戰後青少年的最愛。早期的搖滾音樂是鄉村音樂和 R&B 的融合，所以鄉村和搖滾使用的樂器和節奏非常相似，但鄉村音樂通常是透過簡單的和弦演唱一個故事，而搖滾音樂則是透過各式吉他的操作作為亮點。到了 1970 年代搖滾音樂成為流行音樂的代名詞，並且開創了硬派搖滾(Hard Rock)與重金屬(Heavy Metal)兩種子流派，並在 1970 年代末期簡化搖滾音樂的核心：響亮的吉他、硬朗的旋律和歌唱風格，成為龐克搖滾(Punk)。1980 年代由於主

流搖滾音樂少了創新的發展導致漸漸淡出人們的視野，直到 1990 年代硬派搖滾融合了說唱成為饒舌搖滾(Rap Rock)才重新復出，如今的搖滾音樂也成為各式搖滾的代名詞。



二、音頻分析與曲風分類

在音頻分析與曲風分類相關研究中，Paul Mermelstein 致力於發展梅爾頻譜[47]，他參考了 Bridle 和 Brown [31]的研究。他們將一組由 19 個餘弦轉換出的頻譜型係數，經過一組非均勻分布的帶通濾波器的輸出，來做為轉換後的輸入，由於濾波器的間隔是呈現對數分布，因此又稱為梅爾頻率倒譜系數，正因為梅爾頻譜的出現，推動了音樂範疇和風格分類研究的進展。由於機器學習和深度學習的發展，從以前的支持向量機到現在的卷積神經網路都有了長足的進步。傳統的音訊分析和音樂曲風分類都是基於高斯混合模型(Gaussian Mixture Model, GMM) 和支持向量機(Support Vector Machine, SVM) 構建音訊事件分類。然而，這種方法在高斯混合模型中往往依賴高斯混合數，因此不適合於少量的數據集，因此近年來的音訊分析和音樂曲風分類開始朝著人工智慧和深度學習的方向發展。陳智偉等人[32]利用深度神經網路(Deep Neural Network, DNN) 對聲音相互干擾時的聲音事件進行分離和檢測。他們利用多組 DNN 建立音訊事件的檢測模型，並以 MFCC 作為特徵輸入模式。結果表明，神經網路的識別效率高於傳統的聲學識別方法。易洋等人[4]探討了基於卷積神經網路的支持向量機、K-近鄰算法、高斯混合模型和多層感知器，並通過分層分類器獲得了最高的分類精度。吳紹敬等人[3]採用階層式分類器，結合音樂的時態和非時態特徵，提出了一種基於典型驅動的改進演算法，驗證了他的算法可以將準確率提高 4%。牛璽翔等

人[1]分析了音樂事件密度(Onset Density)、音量(Volume)、調性(Mode)和不和諧度(Dissonance)五種音樂特徵方法，計算特徵值和聽者情緒相關的壓力能量值來分析音樂情緒，並在可攜式設備上實現。張育瑞等人[6]提出了一種在 ACC 壓縮域翻唱歌曲的快速檢索方法，該方法利用解碼後的 MDCT 對應色度特徵(Chroma feature)，將多個語音框組合成音段作為輸入，並構建深度信任網絡(Deep Belief Network, DBN)，使檢索效率 MRR 達到 0.505。吳振銘等人[2]提出了協同過濾(Collaborative Filter)模型和內容式過濾(Content-Based Filter)來實現基於音樂的推薦系統，並提出了改良式 K-means 分群技術。實驗結果表明，改良式 K-means 分群技術對雲端音樂推薦系統的準確率有很好的提高。Panwar 等人[57]將卷積神經網路和遞迴神經網路(Recurrent Neural Network, RNN)融合成 CRNN。目的是利用卷積神經網路選取不同層次的局部特徵，並利用 RNN 分析全域特徵的時間上下文，他們將提出的 CRNN 應用於 MagnaTagATune 數據集，其 AUC-ROC 得分為 0.893。Lokhande 等人[48]將 DBN 應用於印度古典音樂(Indian Classical Music, ICM)。他們把印度古典音樂中的情感稱為 Rasa，並將其分為九種情感，包括 Shringar(愛)、Hasya(幽默)、Karuna(傷感)、Rudra(憤怒)、Veer(英雄主義)、Bhayanaka(恐怖)、Vebhatsa(厭惡)、Adbhuta(奇蹟)、Shanta(平靜)並構建了一個系統。他們的實驗結果表明，該系統可以幫助識別古典音樂中所表現的情感。Sigtia 等人[58]提出了一種用於和弦琴轉錄的監督式神經網路模型，該

模型類似於語音辨識模型。他們建立了五個模型進行比較，包括 Benetos、Vincent、DNN (Deep Neural Networks, 深度神經網路)、RNN 和 CNN。實驗結果表明，CNN 在五種模型中效果最好，因此，將 CNN 模型應用於 MAPS 資料集中，該資料集包含了相應的隔音注釋、和弦注釋和完整的鋼琴音樂注釋。實驗結果表明，該模型可以應用於即時的應用。Vishnupriya 等人[59]比較了基於 GTZAN 資料集的 CNN 上梅爾頻譜和 MFCC 的精確度，精確度分別為 0.76 和 0.47。他們的實驗表明，MFCC 速度較快但準確度較低，梅爾頻譜速度較慢，但準確度較高。Kim 等人[62]提出了一種用於音樂分類的 SampleCNN 架構。該方法與傳統方法步驟不同。傳統的音樂曲風辨識是將原始音訊轉換成梅爾頻譜或 MFCC，然後發送給模型進行訓練。他們的方法是直接使用原始音訊作為訓練輸入，並用一個很小的濾波器代替傳統的濾波器，這種濾波器大大降低了在時域不同階段學習相同濾波器形狀的可能性。他們的實驗是將所提出的樣本 CNN 應用於 MagnaTagATune 並使用 AUC-ROC 計分法表示為 0.9054，這有助於音樂曲風的辨識。Kim 等人[27]提出了一種基於智能手機的移動系統，可以根據用戶當前的活動模式推薦適合該活動的音樂。他們採用基於雙向門控遞迴單元(Bidirectional Gated Recurrent Unit, BGRU)的深度殘差雙向門控遞迴神經網路(DRBGRNN)構建模型，基於識別準確率(Recognition accuracy)的準確率為 0.961，不僅可以改善活動與音樂的關係，同時也提高了音樂類型和情感分類的表現。Yang 等

人[52]提出了一種並行神經網路結構，稱為並行遞迴卷積神經網路(Parallel Recurrent Convolutional Neural Networks , PRCNN)。與傳統的 CRNN 不同的是，傳統的 CRNN 是將 CNN 和 RNN 進行串聯處理，而 PRCNN 對 CNN 和 Bi-RNN 進行並行處理，然後將兩個神經網路的輸入進行融合再進行分類，該結構應用於 GTZAN 資料集和擴展舞廳資料集(Extended Ballroom Dataset)的最高準確率分別為 0.902 和 0.925，說明了 PRCNN 的可行性。Zheng 等人[63]提出了一種用於非監督學習的正規化特徵選擇方法。由於傳統的特徵選擇方法是通過平均樣本來選擇重要的特徵，一般來說樣本是多樣的，這使得傳統的特徵選擇方法缺乏彈性。為了減少離群點對特徵選擇的影響，他們在稀疏特徵選擇模型中添加了一個可自訂進度的正規化方法，來改善傳統方法的缺點。Hassen 等人[34]探討了針對特定領域的音樂曲風分類網路與一般的影像分類網路是否有效能上的差異，他們使用 ResNet18 與 NNet2 分別針對 GTZAN 資料集與他們自己建立的 10GenreGram 資料集進行實驗。他們的實驗證明了影像分類網路與針對特定領域設計的 CNN 架構有著差不多的效能，且影像分類網路不易過度擬合，訓練速度較快，對硬體要求較低，是可以作為針對特定領域的網路的替代。Fu 等人[67]則統整了基於音頻分類的方法並總結了音樂分類的最新技術。Nirmal[43]等人提出了使用 STFT 結合 CNN 方法進行音樂曲風分類，在他們的研究中僅使用三種音樂曲風，分別為 Blues、Classical、Rock 等而非原始 GTZAN 資料集的 10 個曲風，而

他們的研究所得出的最高準確率為 67%。Benetos 等人[19]在 2008 時開發了非負張量分解算法(non-negative tensor factorization, NTF)並提出了一種監督式 NTF 分類器，他們的實驗得到了 75%的準確率，證明了他們的方法優於一般的矩陣分類器，並在 2010 年提出了完整的論文[20]，並在該論文中使用監督式 NTF 分類器得到了 78.9%的準確率。我們也使用神經網路的方式進行音樂曲風分類，在[64]中我們使用卷積神經網路建構音樂曲風分類的模型。我們使用 5 層基礎的卷積神經網路建構我們的模型，並將 GTZAN 資料集應用在模型上。我們實驗的結果顯示了在該資料集中得出的準確率為 84%。在[65]中我們則將 CNN 與 RNN 融合，目的是為了試著找出兩種不同模型融合的可能性。我們使用 5 層的卷積神經網路並在最後添加一層的遞迴神經網路來達到融合的目的，我們實驗結果顯示了該模型應用在 GTZAN 資料集中的準確率為 43%，其原因是訓練到 Epochs 等於 12 時出現了停止訓練。在[66]中提出了使用長短期記憶模型(Long Short Term Memory, LSTM))結合梅爾頻率倒譜系數 (Mel-Frequency Cepstral Coefficients, MFCC)進行音樂曲風分類，在該研究中我們使用 5 層 LSTM 層建置模型，並嘗試社頂不同 Epochs 所得到的準確率。實驗結果表明，在 Epochs 為 2,500 次時得到了最高準確率為 68%，以證明 Epochs 次數並非越多越好。

一、使用之資料集介紹

我們使用 GTZAN 資料集作為我們實驗的資料集來源。該資料集由 Tzanetakis 和 Cook 建立[24]。該資料集總共包含 10 種曲風，分別為 Blues、Classic、Country、Disco、Hip-hop、Jazz、Metal、Pop、Reggae、Rock。每種曲風有 100 首歌曲，總共 1,000 首歌曲，每首歌曲長度為 30 秒，取樣率為 22,050Hz，聲道模式為單聲道模式，音樂格式為 AU 格式，音頻文件為 16 位元文件，資料集的總容量為 1.6GB。表 1 為 GTZAN 資料集的相關資訊。

表1. GTZAN 資料集

項目	資訊
曲風數量	10
歌曲數量	1000
歌曲長度	30 秒
取樣率	22,050Hz
聲道模式	單聲道
音樂格式	AU 格式
音頻文件	16 位元

二、取樣與資料集建置方法

由於在本研究中有兩種不同的實驗方法：傳統音頻訊號與可視化頻譜圖，因此針對不同資料集輸入有不同取樣方式，在傳統音頻訊號中使用梅爾頻譜(Mel Spectrogram)、梅爾頻率倒譜系數(MFCC)與短時距傅立葉變換(STFT)兩種方法。在可視化頻譜圖中採用梅爾頻譜與短時距傅立葉變換兩種取樣方式，但在最後一個步驟時會將頻譜轉換成可視化模式。

(一) 梅爾頻率倒譜系數

梅爾頻率倒譜系數是由 Davis 和 Mermelstein 在 1980 年提出。他們提出了梅爾頻率分析與倒譜分析。他們認為，人類聽覺感知不會感知整個音訊，而只會關注某些特定區域。因此，梅爾頻率分析是人類聽覺感知的一種分析方法。經過實驗發現，人的聽覺感知結構就像一個濾波器，它會特別注意某些特定的頻率成分。也就是說，它只會使某些頻率的聲音通過，而忽略其他不感興趣的頻率訊號。然而，由於這些訊號在頻率坐標軸上分佈不均勻，通常高頻部分的濾波器較少，低頻部分的濾波器較多，因此梅爾刻度的分佈與人耳濾波器的分佈相同，高頻濾波器較少，低頻濾波器較多。圖 1 為梅爾刻度示意圖，其中 X 軸為頻率，Y 軸為振幅。

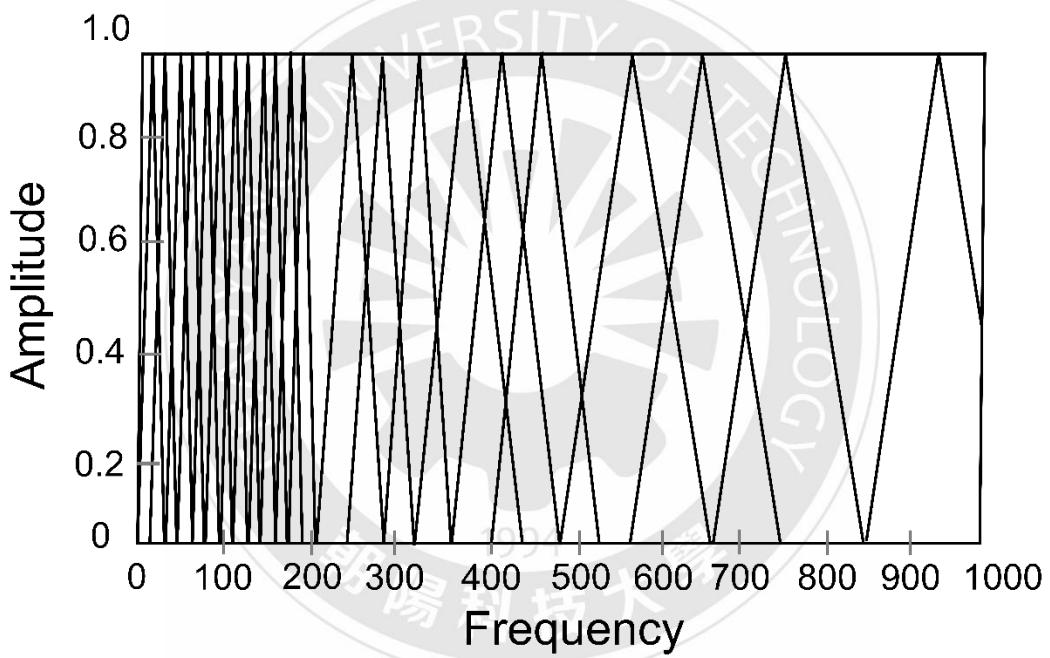


圖1. 梅爾刻度示意圖

梅爾頻率倒譜系數是梅爾頻譜透過倒譜分析後的系數。倒譜分析是音訊經過傅立葉變換後進行對數運算，再經過反傅立葉變換得到的頻譜，這一步稱為倒譜分析(Cepstrum Analysis)。梅爾頻譜主要是將音訊經過傅立葉轉換後將傅立葉的結果再通過梅爾濾波器(Mel filter bank)。該濾波器分佈為高頻濾波器數量較少而低頻濾波器數量較多。梅爾濾波器由 20 個非線性三角帶通濾波器(Triangular Bandpass Filters)組成。由於梅爾頻譜是基於人耳聽覺感知模式，因此更適合用於語音辨識技術、音樂曲風分類、樂器辨識等領域。

要將音訊轉換成梅爾頻譜需要五步驟。第一步是預加重原始音訊，並將訊號 $s(n)$ 通過高通濾波器，如公式(1)所示。如果以時域運算式表示，則預加

重訊號 $s_2(n)$ 如公式(2)所示。其中 a 介於 0.9 和 1.0 之間。使用預加重的目的是補償語音系統抑制高頻的部分。預加重後，聲音會變得尖銳清脆但音量會變小[28]。

$$H(z) = 1 - a * z^{-1} \quad (1)$$

$$s_2(n) = s(n) - a * s(n - 1) \quad (2)$$

第二步是音框化，進行音框化時會將兩個音框之間進行重疊，重疊部分包括 N 個取樣點，一般來說 N 的數量是兩個聲音幀中取樣點的一半或 1/3。將音訊做音框化的目的是方便對音訊做處理。在一段訊號中每一段的訊號變化都很大，因此必須將訊號分成小段處理，使得小段之內訊號的變化變小，這些小段的訊號稱之為幀。而由於音訊經過切割後會造成訊號不連續，因此音框化會將所有幀中的頭尾訊號重疊，使訊號保持連續，如圖 2 所示。漢明窗公式如公式(3)所示。不同的 a 值將產生不同的漢明窗。一般來說 a 值會設置為 0.46 [7]。

帧 帧

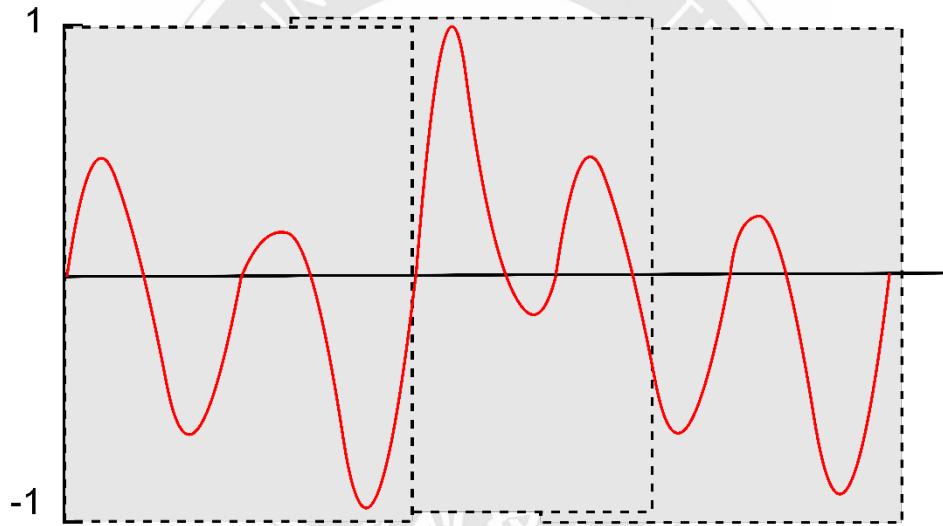


圖2. 音框化

第三步是加窗函數，僅靠音框化的重疊使音訊保持連續性相當不可靠，因為真實的音樂訊號是相當複雜的，一段音框中所包含的變化是相當大的，且在音框化中僅做音訊取樣而沒有做其他音訊處理，因此若僅使用音框化的重疊想保持音訊連續性則會使音訊重疊，聽起來就像一首歌好幾的地方重複播放或混雜的感覺。為了使音訊不出現上述情況，則需添加窗函數，在本研究中使用的窗函數為漢明窗(Hamming Window)，使用窗函數可增加聲音幀左右兩端的連續性，從而避免音訊因為取樣和量化造成的不連續性和丟失重要信息的問題。漢明窗公式如公式(3)所示， α 值會產生不同的漢明窗。圖3為訊號經過漢明窗的結果，其中上圖為原始訊號，中圖為漢明窗，

下圖為訊號經過漢明窗的結果。由圖 3 可以看出，窗函數會使訊號左右兩端抑制，使訊號滿足傅立葉變換對週期性的要求，並減少音訊洩漏[28]。

$$W(n, \alpha) = (1 - \alpha) - \alpha \cos(2\pi n / (N - 1)), \quad 0 \leq n \leq N - 1 \quad (3)$$

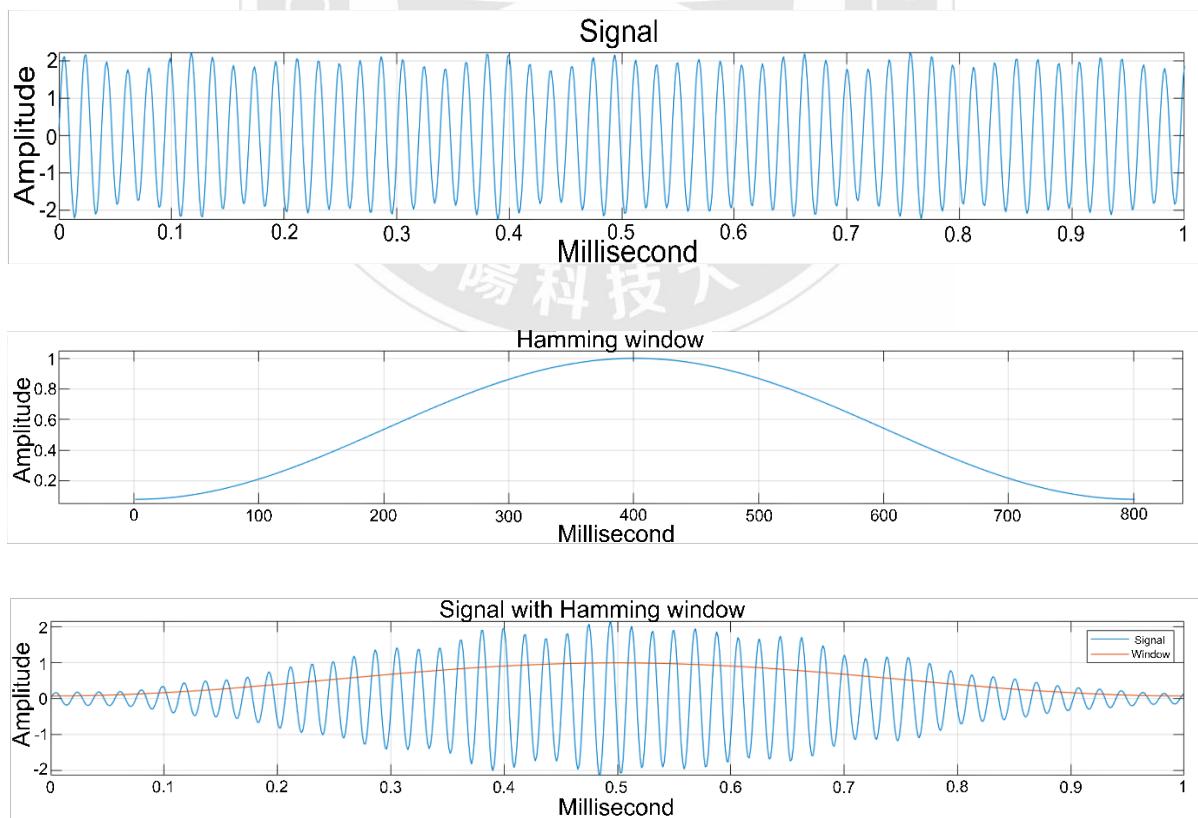


圖3. 漢明窗

第四步是快速傅立葉變換(Fast Fourier Transform, FFT)。由於訊號在時域時很難看出特徵，因此將訊號轉換成頻域的能量分佈型態進行觀察。在FFT中，若音框的左右兩端沒有呈現週期性分佈，會產生原本沒有的能量分佈，這會使新訊號與原始訊號出現差異導致分析誤差，因此需添加漢明窗，以保持原始訊號左右兩側的週期連續性。公式4為傅立葉變換的公式[28]。

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad (4)$$

第五步是將 FFT 後的結果映射到梅爾刻度(Mel scale)上，作法是將 FFT 的結果乘以一組 20 個三角帶通濾波器，會得到 20 個由濾波器輸出的對數能量。這 20 個三角帶通濾波器是平均分佈在梅爾頻率上。公式 5 為三角帶通濾波器的公式，其中 $B_m[k]$ 是梅爾濾波器[28]。

$$Y[m] = \sum_{k=f_{m-1}}^{f_{m+1}} |x[k]|^2 B_m[k] \quad (5)$$

以上五步驟即可將音訊轉換成梅爾頻譜，而若想將音訊轉成 MFCC 則還需執行第六步與第七步。第六步是離散餘弦變換(Discrete Cosine Transform, DCT)[28]。DCT 步驟會將第五步的 20 個對數能量 E_k 帶入運算，以求出 L 階的梅爾刻度倒譜(Mel Scale Cepstrum)，DCT 的公式如公式 6 所示。其中 E_k 為步驟五中求得的三角帶通濾波器和頻譜能量的內積值， N 為三角帶通濾波器個數， L 通常設定為 12。

$$C_m = \sum_{k=1}^{N} \sum_{m=1,2,\dots,L} N \cos[m * (k - 0.5) * \pi / N] * E_k \quad (6)$$

第七步包含了取對數能量(Log Energy)與差量倒頻譜參數(Delta Cepstrum)[28][35]。在音訊中，一個音框的音量也是語音特徵且計算容易，因此通常會加上音框的對數能量，使得每個音框中的特徵數提升到 13 個(1 個對數能量與 12 個倒頻譜參數)。然而即使已經提取 13 個特徵，在真實應

用時會再加上差量倒頻譜參數，其目的是顯示 12 個倒頻譜參數對時間的變化，也就是代表倒頻譜參數在時間上的動態變化，其公式如公式 7 所示。

$$\Delta C_m(t) = \frac{\sum_{\tau=1}^M \tau * (C_m(t + \tau) - C_m(t - \tau))}{2 * \sum_{\tau=1}^M \tau^2} \quad (7)$$

(二) 短時距傅立葉變換

短時距傅立葉變換常用於語音訊號處理，其指定了任意信號隨時間和頻率變化的複數幅度。短時距傅立葉轉換也適用於分析非平穩信號。非平穩信號由於波形的變化沒有規律，也沒有瞬間頻率的概念，不能直接使用快速傅立葉變換。音樂也屬於非平穩信號，因此適合使用短時距傅立葉變換。短時距傅立葉變換的計算過程是把一個較長的時間信號分成相同長度或更短的段落，並在每個段落上計算傅立葉變換，即為短時距傅立葉轉換。短時傅立葉轉換有四個步驟，分別為分幀、加窗函數、傅立葉轉換、合併傅立葉結果。其公式如公式 8 所示。其中 $x(n)$ 為在時間 n 的輸入信號， $w(n)$ 為窗函數 M 的長度， $Xm(\omega)$ 為以時間 mR 為中心的窗函數離散時間傅立葉變換資料， R 為以樣本為中心的連續 DTFT 之間 hop size 大小。公式 9 為離散時間傅立葉轉換(DTFT)公式，其中 ω 為頻率變數， $x(n)$ 為在取樣數 n 處的訊號幅度， $\tilde{\omega}$ 的定義則如公式 10 所示，其中 T 為時間。在實際中，短時距傅立葉轉換是計算一系列加了窗函數幀的快速傅立葉變換，其中窗函數會隨著時間滑動(或稱為跳躍(hop))[33]。

$$\sum_{n=-\infty}^{\infty} x(n)w(n-mR)e^{-j\omega n} \quad (8)$$

$$x(\tilde{\omega}) \equiv \sum_{n=-\infty}^{\infty} x(n)e^{-j\tilde{\omega}n} \quad (9)$$

$$\tilde{\omega} \equiv \omega T \in [-\pi, \pi] \quad (10)$$

(三) 可視化梅爾頻譜

本研究所提出的可視化梅爾頻譜是將音訊透過梅爾頻譜的五步驟轉換成梅爾頻譜後，以橫軸為時間縱軸為頻率的規定，透過 MATLAB 繪製完成。我們將 30 秒的原始音頻透過 MATLAB 轉換成梅爾頻譜後，使用漢明窗作為實驗的窗函數，窗函數尺寸設定為 1,024。重疊區域(Overlap)設定為 512，即為有 512 幀會重疊。傅立葉變換尺寸設定為 4,096 幀。我們的參數設定如表 2 所示。而透過這樣的參數轉換出的可視化梅爾頻譜圖則如圖 4 所示，其中該曲風為 Blues。由於可視化梅爾頻譜圖是使用圖片作為神經網路的輸入，因此我們透過 LabelImg 進行特徵框選，LabelImg 是一套影像標註工具 [26]，它可以根據選定之資料夾內的影像進行特徵標記並給予註釋，且可產生 PascalVOC 與 YOLO 等格式，圖 5 中紅色框表示我們所選定的特徵區域，透過 LabelImg 會根據紅色框輸出 XML 檔案，該檔案內即含有特徵的座標點，模型會根據 XML 檔案與圖片進行訓練。

表2. 可視化梅爾頻譜參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帀
重疊區域數	512 帀
傅立葉變換尺寸	4,096 帀
濾波器個數	64 個
橫軸單位	時間/秒
縱軸單位	頻率

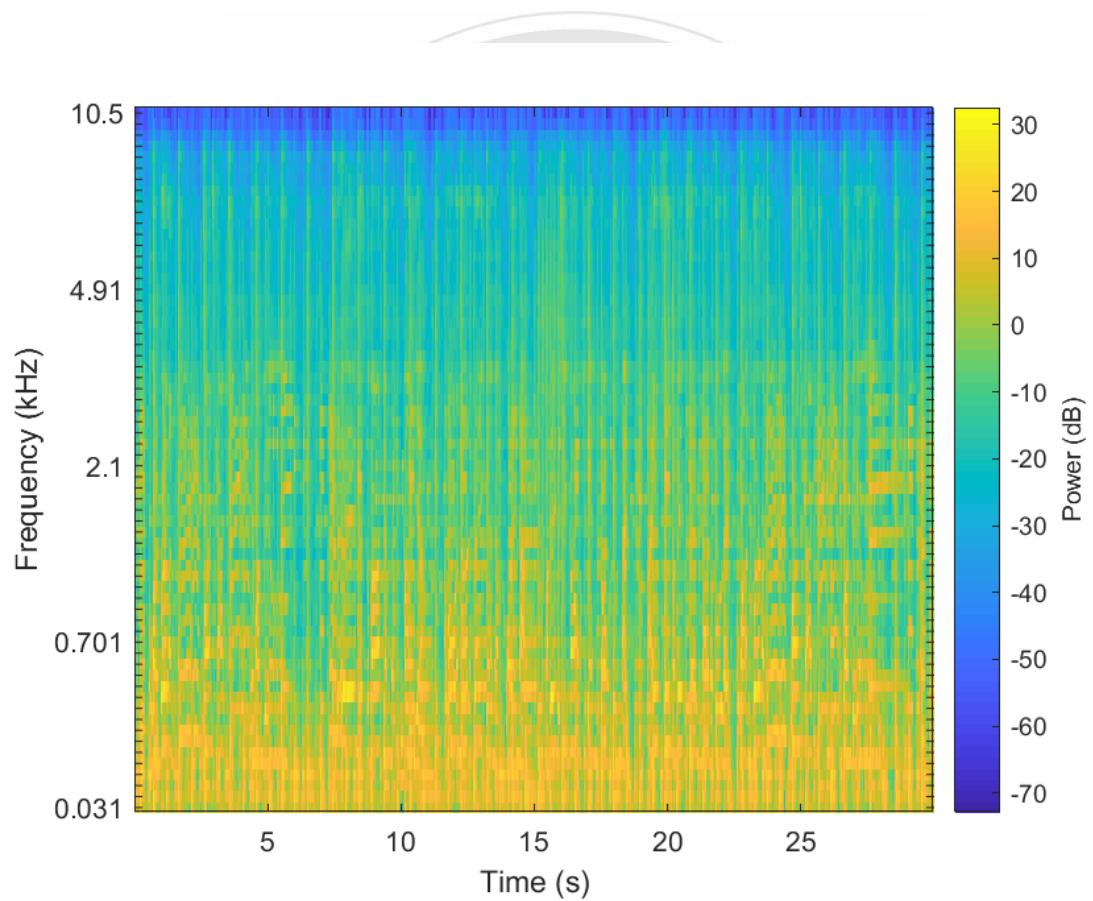


圖4. 可視化梅爾頻譜圖

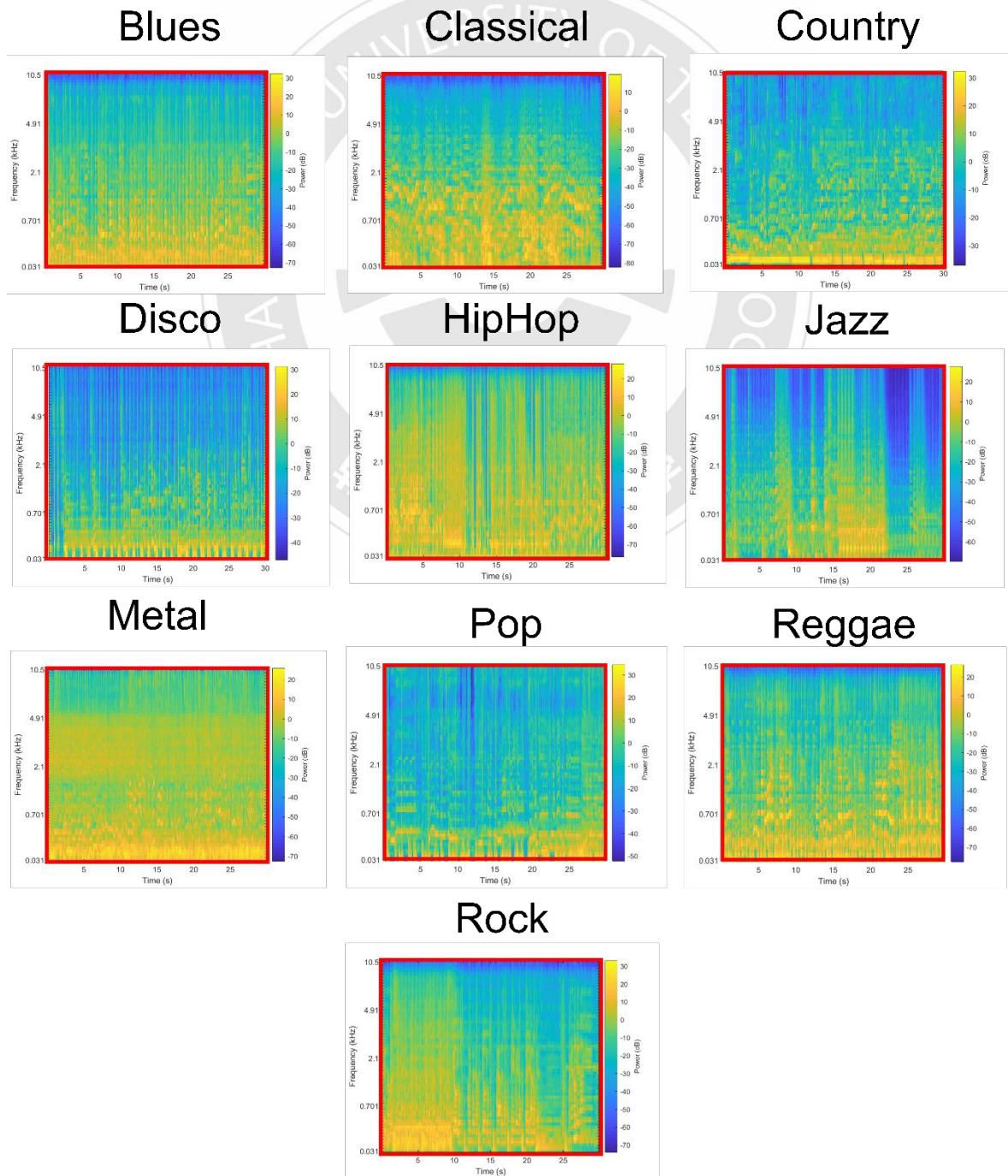


圖5. 建置可視化梅爾頻譜圖資料集

(四) 可視化短時距傅立葉變換頻譜

為了探討不同的可視化頻譜圖做為是否會影響實驗結果，因此我們設定可視化短時距傅立葉變換頻譜的轉換參數會與可視化梅爾頻譜的參數相同。如表 3 所示。而圖 6 則為可視化短時距傅立葉變換頻譜圖，其中該曲風為 Blues。圖 7 為可視化短時距傅立葉變換資料集建置方式。

表3. 可視化短時距傅立葉變換頻譜參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帧
重疊區域數	512 帧
傅立葉變換尺寸	4,096 帧
橫軸單位	時間/秒
縱軸單位	頻率

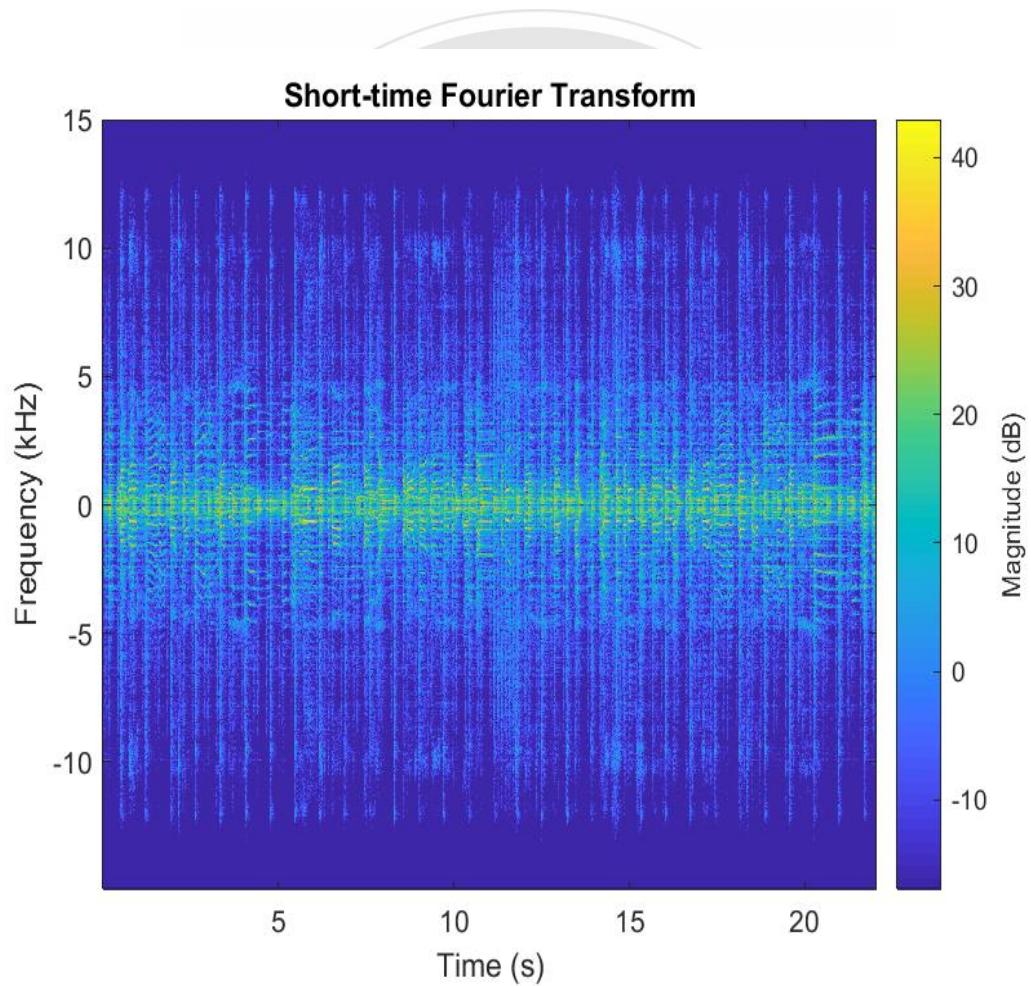


圖6. 可視化短時距傅立葉變換頻譜圖

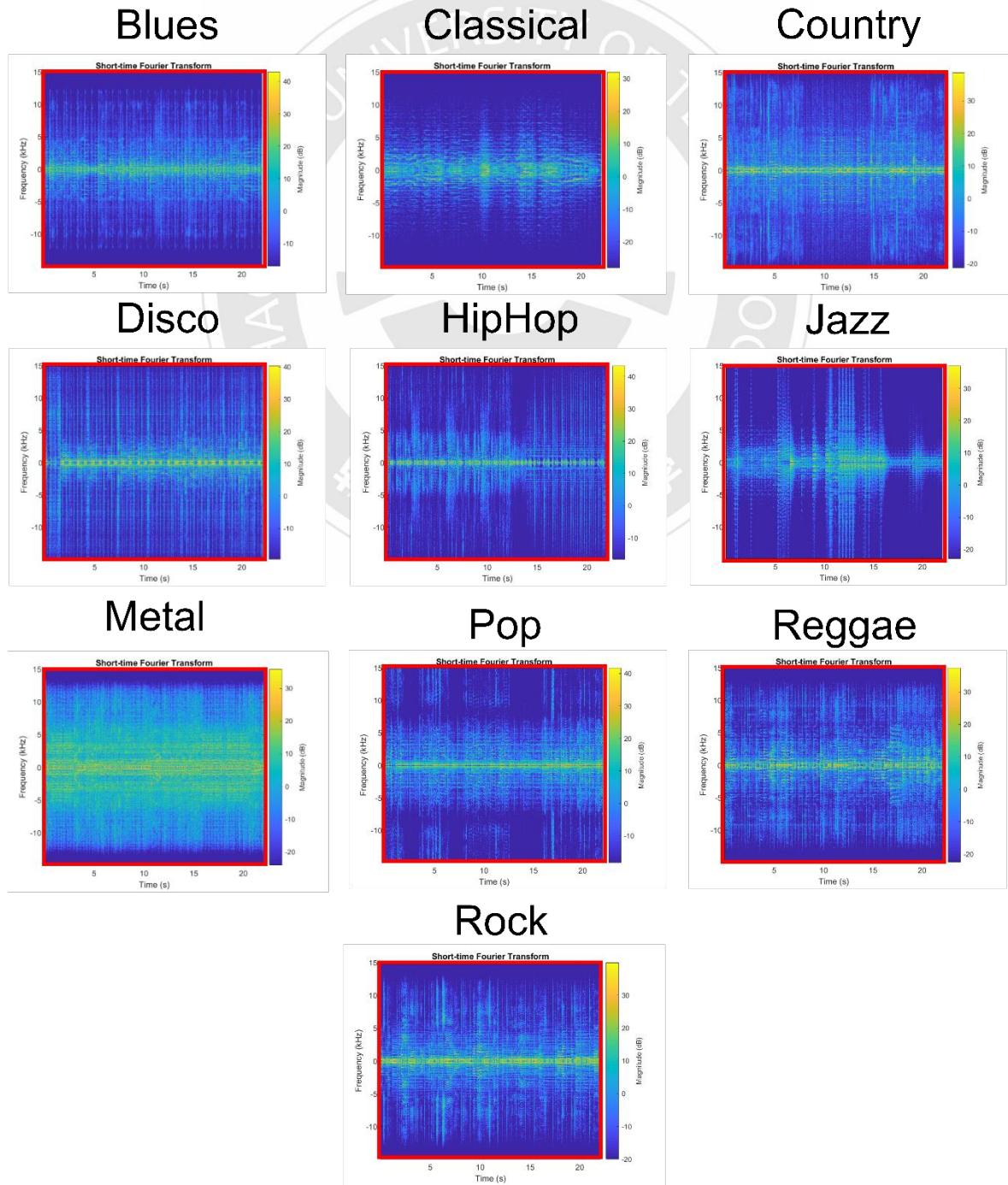


圖7. 建置可視化短時距傅立葉變換頻譜圖資料集

三、使用的方法

本研究為了探討可視化頻譜圖的可行性，參照傳統的音訊處理方法與神經網路方法，並提出使用 YOLOv4 進行模型建置與訓練。使用 YOLOv4 是由於近年來影像處理進步的速度相較於音訊處理來的快，且基於圖像的深度學習發展也較音訊處理來的穩健，因此我們提出使用視覺化的頻譜圖，並與傳統的音訊處理方法進行比較與驗證。在本研究中我們使用卷積神經網路搭配 MFCC、卷積循環神經網路搭配 MFCC、長短期記憶模型搭配 MFCC、YOLOv4 搭配可視化梅爾頻譜圖、YOLOv4 搭配可視化短時距傅立葉變換頻譜圖等五種方法進行比較。

(一) 卷積神經網路

卷積神經網路由卷積層、池化層、全連接層所組成，其中卷積層內的卷積運算由四個部分組成：輸入資料大小、濾波器（Kernel map）大小、步幅(Stride)和輸出資料大小，Kernel map 與 Stride 會直接影響卷積運算的結果。卷積層的原理是透過一個可指定尺寸的窗(Convolutional kernel)，經由左右上下依序的滑動取得訊號的局部特徵，接下來經過激勵函數(Activation function)產生特徵圖(Feature map)當作下一層的輸入，如圖 8 所示。其中綠色方塊為輸入，紅色方塊為 3×3 的 Kernel map，橙色方塊為卷積運算的結果，假設設定步幅為 1×1 ，即紅色方塊會在綠色方塊中按照 1×1 的距離移動，在第一步中將綠色方塊的值與紅色方塊的值相乘，即

$((1*1)+(1*0)+(0*1)+(1*0)+(1*1)+(1*0)+(0*1)+(0*0)+(1*1))=3$ ，則橙色方塊的第一格結果為 3，以此類推則可以得到完整的 3*3 橙色方塊結果。

1	1	0	0	1
1	1	1	1	0
0	0	1	0	1
1	1	1	1	1
1	0	0	1	1

X

1	0	1
0	1	0
1	0	1

=

3	2	4
4	5	3
3	2	4

圖8. 卷積運算

在每層卷積層後會使用激勵函數，其目的是為了增強神經網絡的非線性變化，若是神經網路沒添加激勵函數就只是矩陣運算，即便添加很多層神經網路也還是矩陣運算，這使得深度學習失去意義。常用的激勵函數有六種：Sigmoid、tanh、ReLU、PReLU、Swish、Mish。Sigmoid 函數是將 $(-\infty, +\infty)$ 值的圖像轉移到 $(0,1)$ 。Sigmoid 具有優化穩定、操作簡單的優點。但由於其軟飽和的特性容易造成梯度消失，導致訓練出現問題。梯度消失是指神經網路在反向傳播中，梯度不斷減少直到零，這會使淺層的神經網路權重無法更新，前層的學習速率低於後層的學習速率，而隨著層數增加，分類準確率反而會下降。梯度爆炸則與梯度消失相反，在反向傳播中梯度不斷增大，導致權重

更新太大，前層的學習速率高於後層的學習速率，這種現象即為梯度爆炸。

Sigmoid 公式如公式 11 所示，圖 9 為 Sigmoid 函數圖[46]。

$$g(z) = \frac{1}{1 + e^{-z}} \quad (11)$$

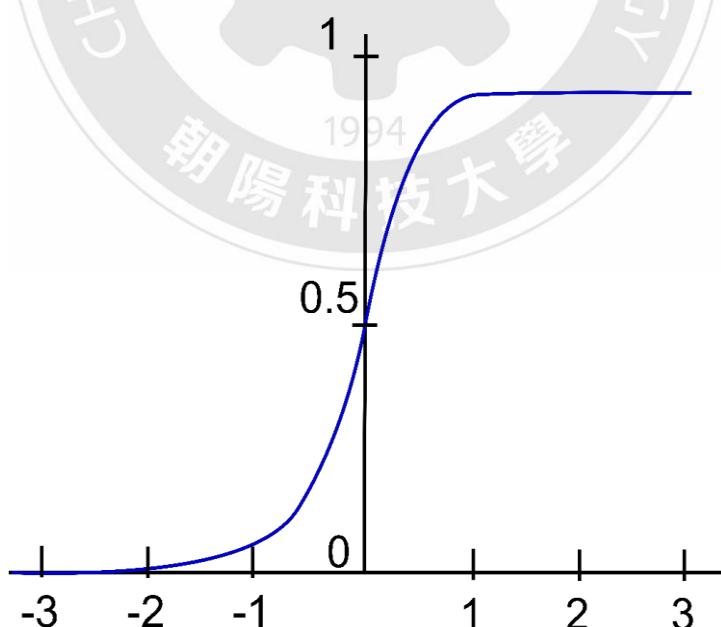


圖9 . Sigmoid 函數圖

\tanh 函數將具有 $(-\infty, +\infty)$ 值的數字映射到 $(-1, 1)$ 。 \tanh 的收斂速度比 Sigmoid 函數快，但它也有一個缺點，就是因為飽和所導致的梯度消失。 \tanh 公式如公式 12 所示，而圖 10 為 \tanh 函數圖[46]。

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (12)$$

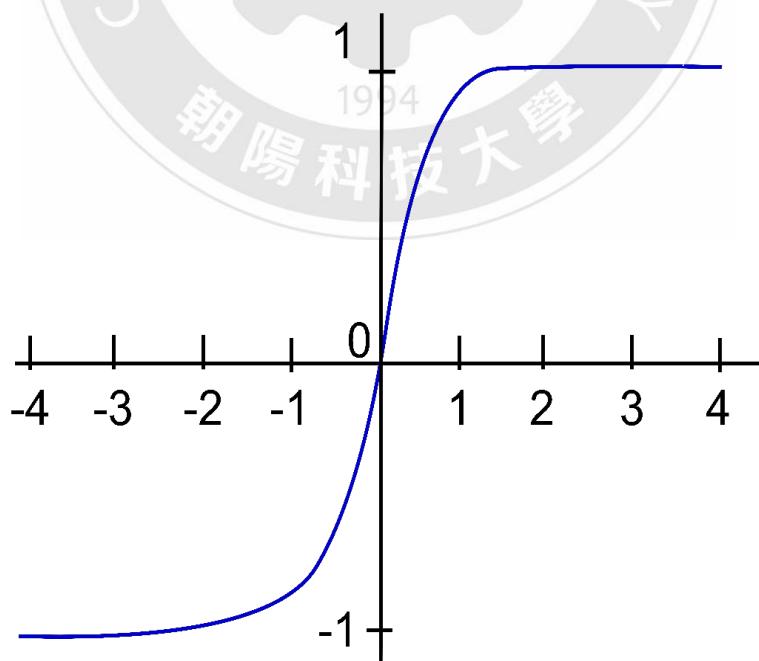


圖 10. \tanh 函數圖

ReLU 函數，又稱為 Rectified Linear Unit，是一個分段線性函數。ReLU 可以在隨機梯度算法(Stochastic gradient descent, SGD)中快速收斂，減少梯度消失問題，且在非監督式預訓練中表現良好，可以提供神經網路的稀疏表達能力。但隨著訓練的進行，可能會出現神經元死亡，這將導致權重無法更新，從這一刻開始神經元的梯度就永遠為 0，即不可逆死亡，ReLU 公式如公式 13 所示，圖 11 為 ReLU 函數圖[14]。

$$g(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{if } z \leq 0 \end{cases} \quad (13)$$

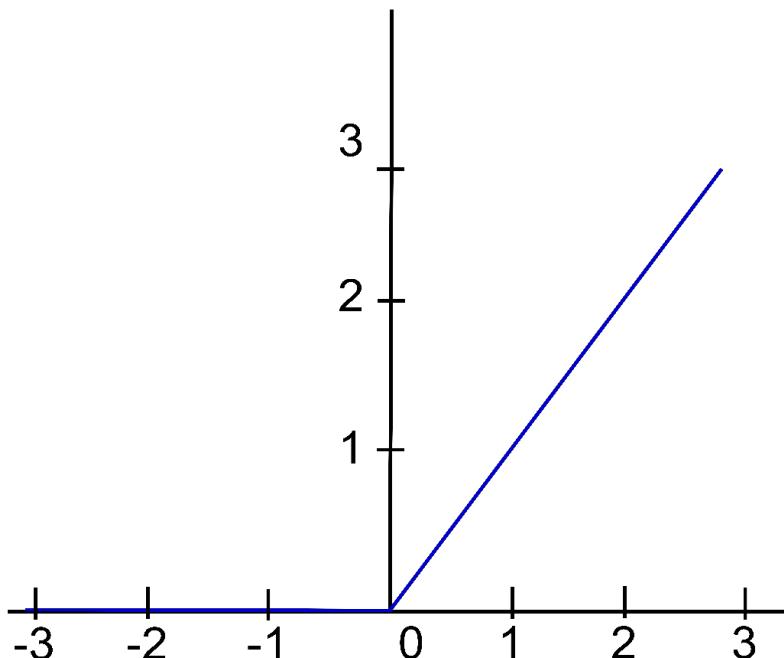


圖 11. ReLU 函數圖

Leaky ReLU 是一個改進 ReLU 的函數，與 ReLU 不同的是，Leaky ReLU 會給負值賦予一個非零斜率，因此即使輸入值為負時也可以進行反向傳播，並且具備收斂速度快，錯誤率低等優點。Leaky ReLU 是為了要解決 ReLU 中神經元死亡的問題，但並非所有情況下 Leaky ReLU 會優於 ReLU。Leaky ReLU 公式如公式 14 所示，圖 12 為 Leaky ReLU 函數圖[14]。

$$g(z) = \begin{cases} z, & \text{if } z > 0 \\ az, & \text{if } z \leq 0 \end{cases} \quad (14)$$

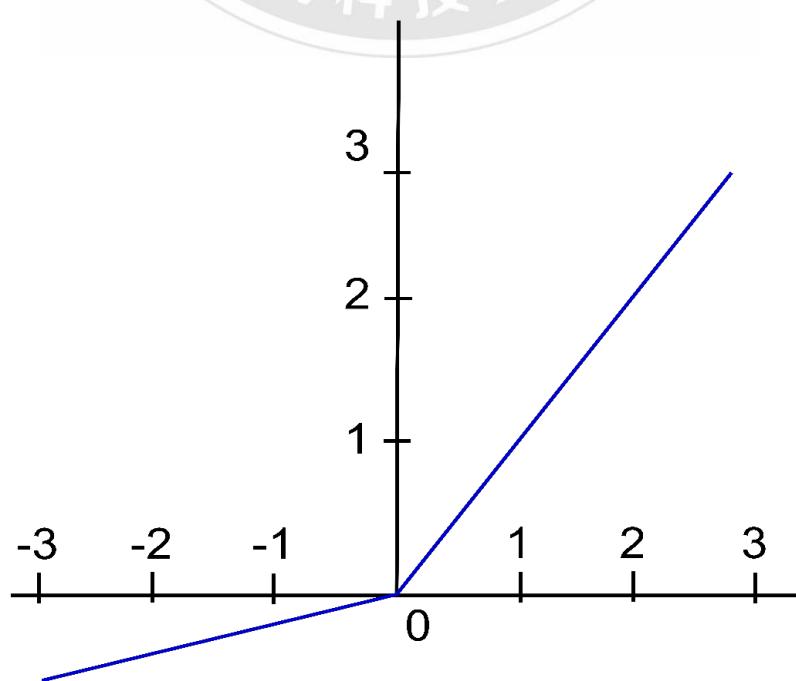


圖 12. Leaky ReLU 函數圖

Swish 是介於線性函數和 ReLU 函數之間的平滑函數，因此在深度模型中優於 ReLU。Swish 沒有上邊界，所以不會有梯度飽和，Swish 同時具備下邊界，可以產生更強的正則化效果。Swish 訓練容易但不穩定，不同的任務中會有不同的結果。Swish 公式如公式 15 所示，不同 β 會產生不同曲線，如圖 13 之 Swish 函數圖所示[5]。

$$f(x) = x * \text{sigmoid}(\beta x) \quad (15)$$

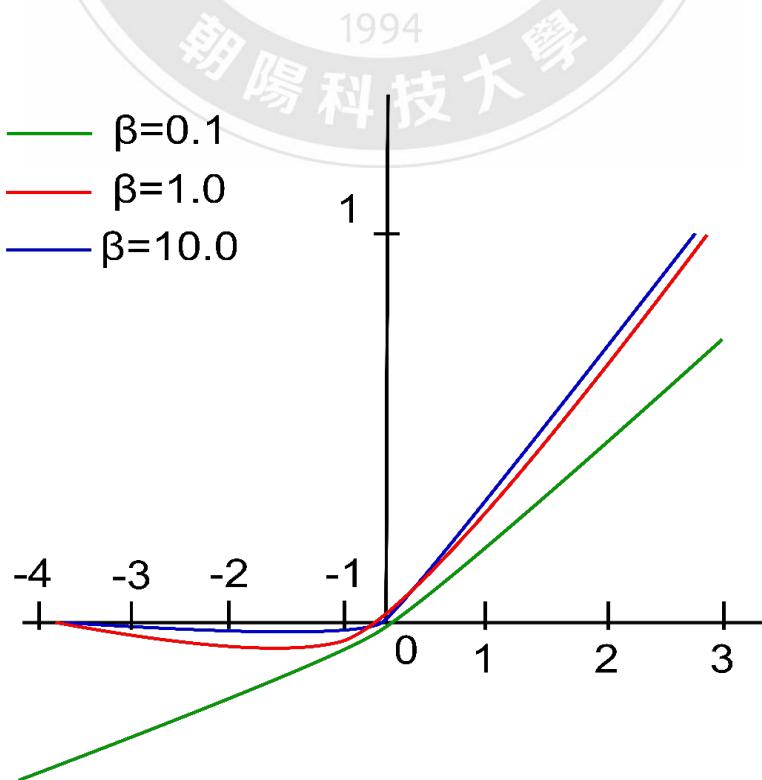


圖13. Swish 函數圖

Mish 是一個自正則化的非單調神經激活函數。Mish 具有成本低、曲線平滑、非單調、無上邊界、有下邊界等特點，但計算量大。沒有上邊界是任何激活函數都需要的特性，因為它避免了梯度飽和導致訓練速度急劇下降。非單調函數有助於保持較小的負值，從而穩定網路梯度。Mish 是一個平滑函數，具有良好的泛化能力和有效的結果優化能力，可以提高結果的質量。Mish 公式如公式 16 所示，圖 14 為 Mish 函數圖[50]。

$$f(x) = x * \tanh(\ln(1 + e^x)) \quad (16)$$

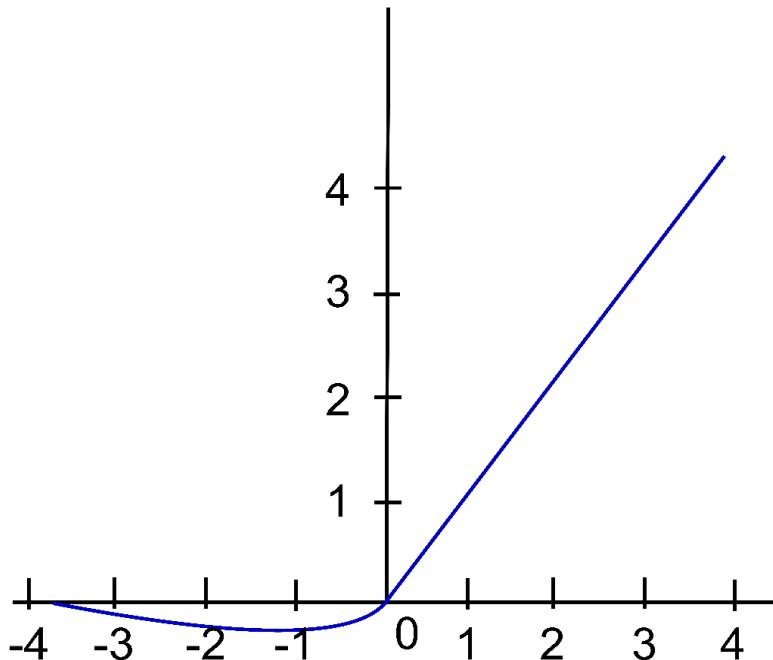


圖 14. Mish 函數圖

池化層主要作用是壓縮資料並保留重要資訊，透過降低每張特徵圖的維度並保留重要特徵，以減少計算資源與時間。常見的池化方法有兩種：最大池化與平均池化。最大池化的方式是通過取矩陣內最大值以進行資料縮減，如圖 15 上方所示，其中在 $4 * 4$ 矩陣中分別於藍色、綠色、紫色與紅色矩陣中取出各自的最大值後重組成 $2 * 2$ 的區塊，即完成最大池化的步驟。平均池化的方式則是通過取矩陣內平均值的方式壓縮資料，如圖 15 下方所示。雖然經過池化層後資料會有所遺失，但保留下來的資料是最符合特徵圖的結果。由於圖片在卷積層時會與卷積核進行運算以找出特徵，因此在圖片中特徵越明顯則卷積後該特徵值會被放大，而透過池化層將該特徵保留並去除其餘較不符合的值後，即可以降低神經網路的運算資源與減少時間。

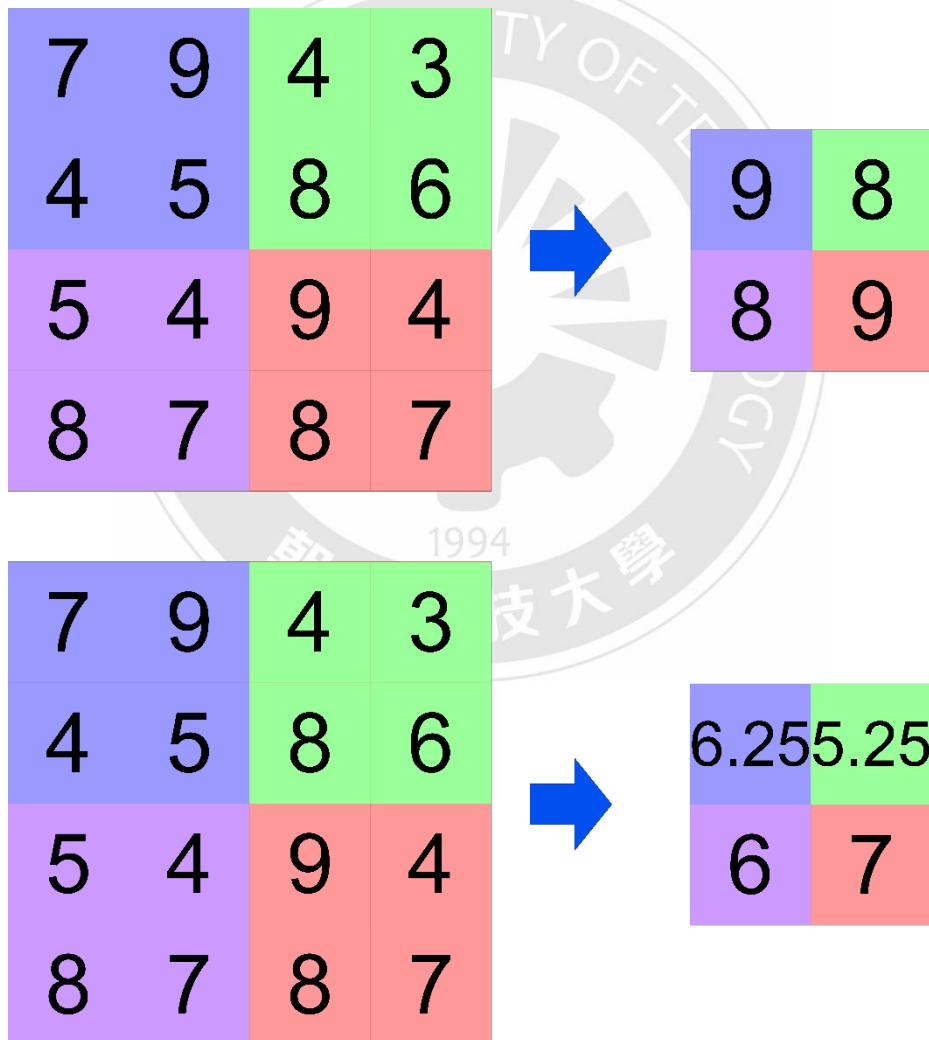


圖15. 最大池化與平均池化示意圖

全連結層的作用是將經過前幾層運算得出的特徵經過權重的運算後獲得最後的分類結果。特徵圖在進入全連接層前會先從二維陣列轉換成一維陣列，並且每個一維陣列中都有數值，即為權重(weight)，這些權重會決定所輸入的圖片應該被分類到哪種類別中，如圖 16 所示。

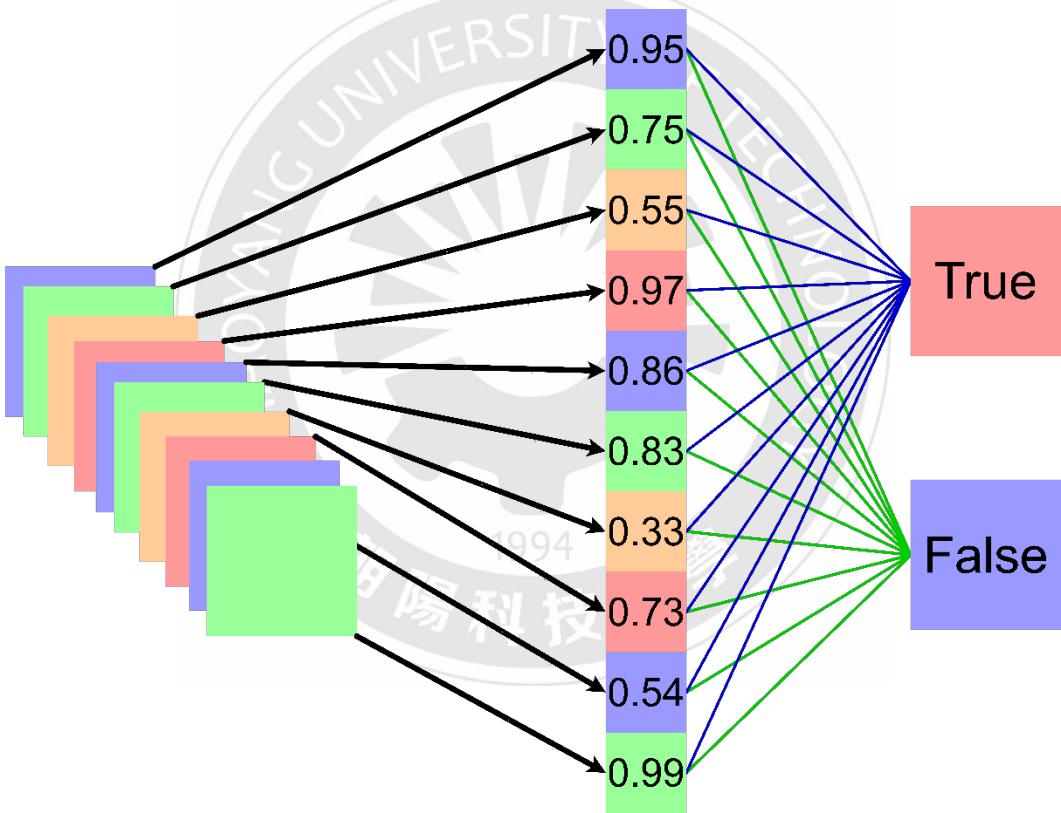


圖16. 全連接層示意圖

在每次迭代中都會加入 Dropout 以防止過擬合的發生。過擬合是指神經網路對特徵的學習過於徹底，學習了所有樣本特徵，無論是好的還是壞的，包括樣本中的雜訊，這導致神經網路的學習精度雖然很高，但模型的泛化和識別精度很低。Hinton 等人[22]在 2012 年提出的論文中提到，在較小的資料集中容易出現過擬合，並在[13]時運用了 Dropout 以證明 Dropout 的可行性。而 Dropout 的方法是在訓練中隨機使神經元不加入本次迭代中，這可以使每個神經元之間的相互作用降低，以防止各個神經元之間過於依賴，因此在順向傳播時使某個神經元以設定的機率停止工作，以此增加模型的泛化

性與降低過擬合的發生機率，一般來說 Dropout 會設定 0.5，即為有 50% 機率會使神經元停止工作，當 Dropout 設定為 0.5 時隨機生成的網路架構最多 [45]。Dropout 示意圖如圖 17 所示。其中左圖為未使用 Dropout 的神經網路架構，右圖為使用 Dropout 的神經網路架構。

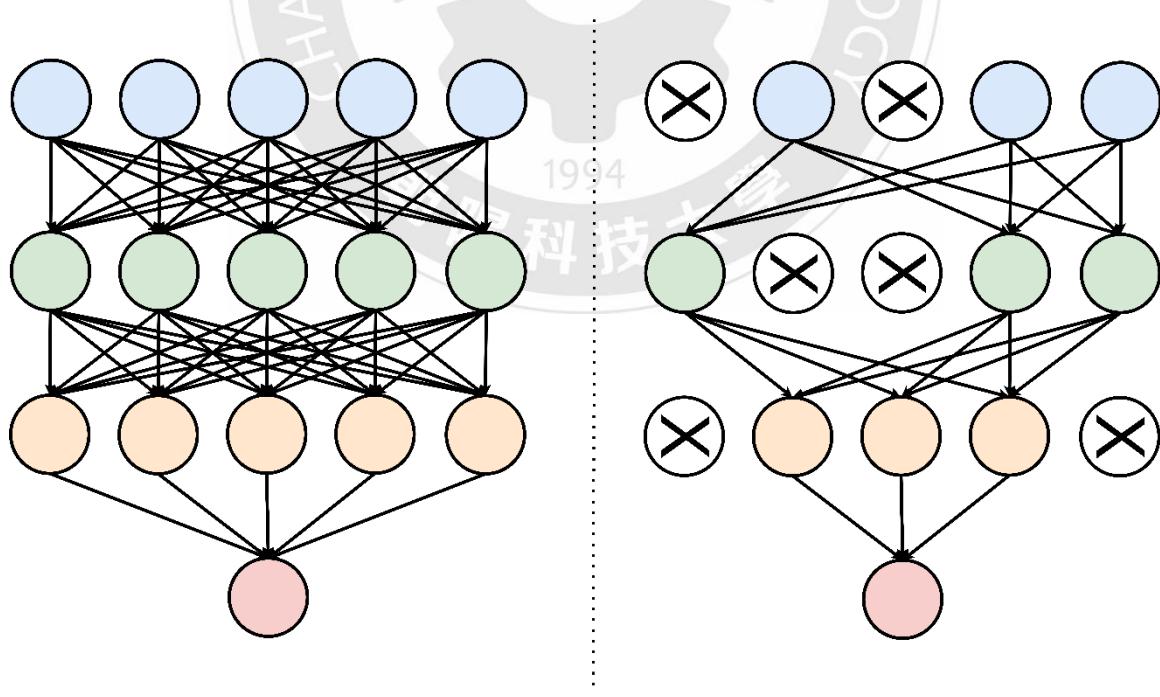


圖 17. Dropout 示意圖

在[64]中所使用的卷積神經網路由 5 層卷積神經網路所組成，並設定卷積核為 3×3 ，步幅為 1×1 ，激勵函數使用 ReLU，Dropout 設定為 0.5，其架構圖如圖 18 所示。

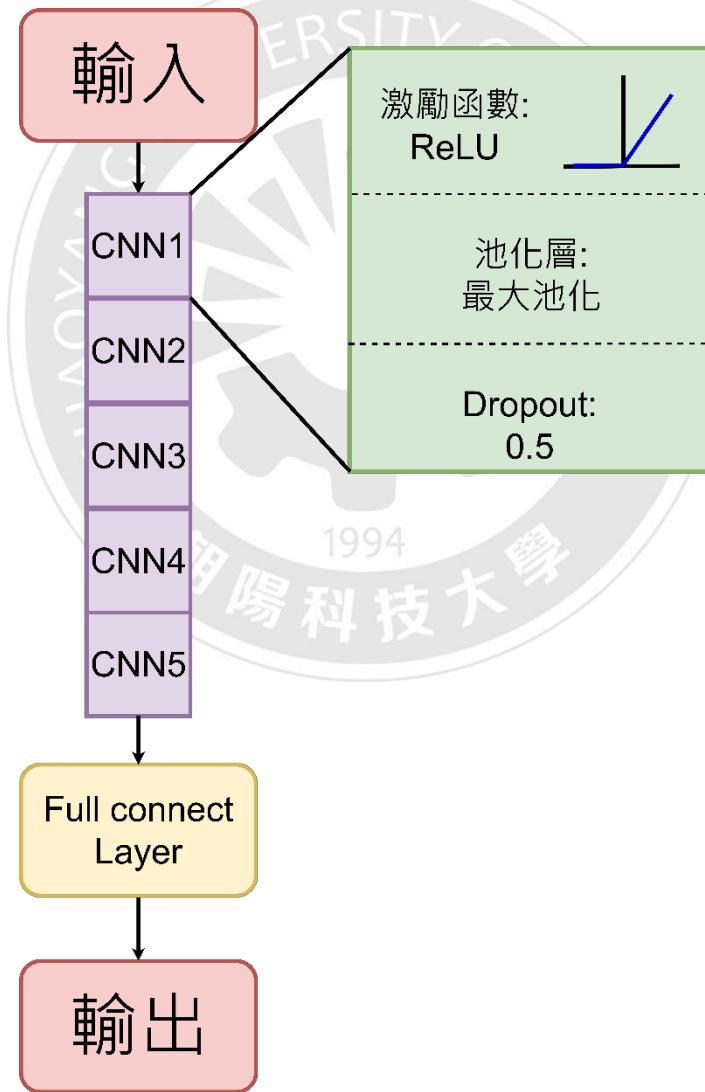


圖18. CNN 架構圖

(二) 卷積循環神經網路

卷積循環神經網路(Convolutional Recurrent Neural Network, CRNN)是由卷積神經網路與循環神經網路(Recurrent Neural Network, RNN)所組成。循環神經網路常用於解決與時間相關的問題，例如：預測股票、語音辨識等。與一般神經網路前後文可以互相呼應，RNN 的輸入與輸出不會是固定關係，這是因為 RNN 具備記憶功能。在 RNN 中加入了記憶空間(Hidden State)，這個記憶空間不能容納太多資訊，但可以保留上一次的輸出，並且每一次輸出之前都會參考在記憶空間內的結果是什麼，使 RNN 可以學習與前文之間的對應關係。也因為 RNN 具備這種特性，使得 RNN 在訓練時較一般神經網路來的困難且容易出現梯度消失。為了解決在 RNN 中的梯度消失問題，我們在 RNN 中添加門控循環單元(Gated recurrent unit, GRU)，GRU 有兩個門：重製門(Reset Gate)與更新門(Update Gate)。重製門是用於決定是否要遺忘之前留下來的記憶空間，更新門則是決定這次的記憶空間是否要保留。透過 GRU 的兩個門之間的運算得到的記憶空間可使兩者之間的歷史資訊變成線性關係，從而避免梯度消失。圖 19 為 RNN 的示意圖，其中 I_t 為在時間 t 的輸入， O_t 為輸出， W 、 V 、 U 是權重，前一個狀態在經過一個 t 時會與權重做加成運算。神經元在每個 t 時接收 I_t 以及上一次的輸出 O_t ，因此會有兩個權重：輸入 I_t 的權重 W_I 與上一次輸出 O_t 的權重 W_O 。

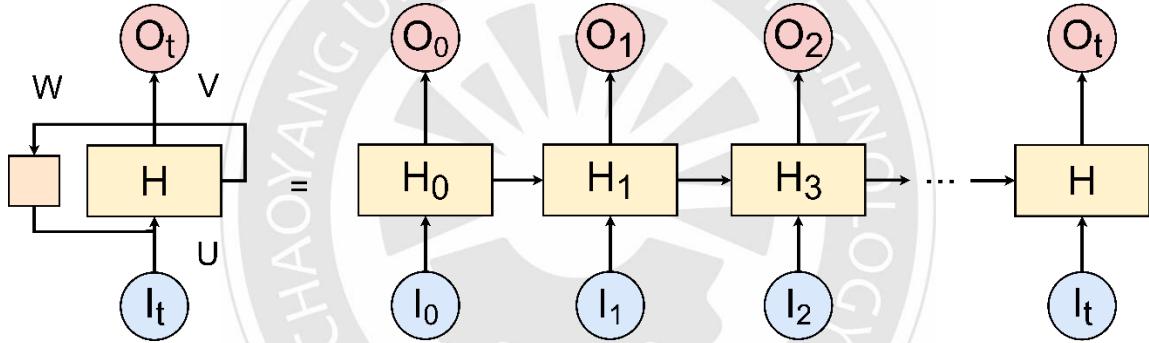


圖19. RNN 示意圖

在[65]中將卷積神經網路與循環神經網路進行結合，其架構如圖 20 所示，方法為在最後一層的 CNN 層之後透過轉換將 CNN 的輸出轉換成 RNN 的輸入並進行 RNN 運算。與前面卷積神經網路架構不同的是，CRNN 中採用 Leaky ReLU 作為激勵函數而非 CNN 中的 ReLU。在[65]所使用的 CRNN 由 4 層 CNN 層與 1 層 RNN 層所組成，並設定卷積核為 3×3 ，步幅為 1×1 ，激勵函數使用 Leaky ReLU，Dropout 設定為 0.5，兩層 GRU 層內都使用 32 個神經元，並在最後分類層進行分類。

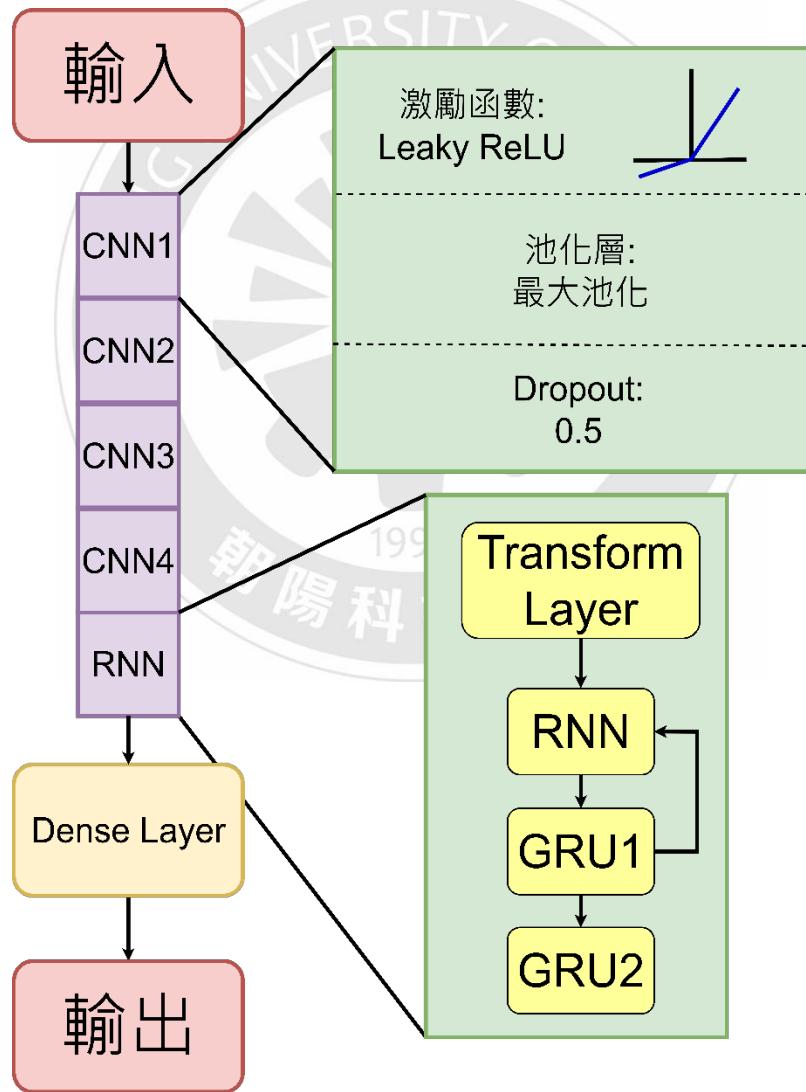


圖20. CRNN 架構圖

(三) 長短期記憶模型

長短期記憶(Long Short Term Memory, LSTM)是循環神經網路的一種改進版本，其是為了解決循環神經網路在長時間序列訓練過程中的梯度消失和梯度爆炸等問題。LSTM 是由 Jürgen Schmidhuber 和 Sepp Hochreiter [54] 在 1997 年提出的模型架構，由於 LSTM 的設計使得它適合用於預測和處理時間序列中間隔與延遲較長的事件。單一 LSTM 結構如圖 21 所示，其運算如公式 17~22 所示。

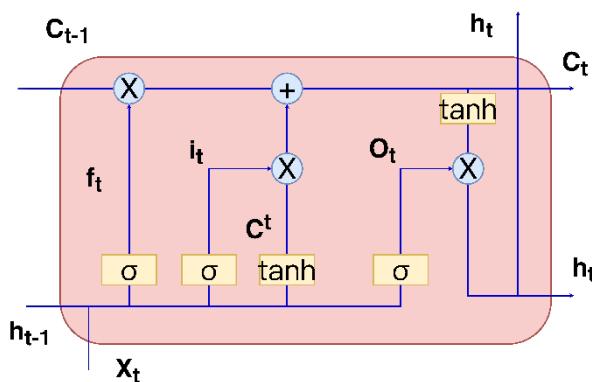


圖21. 單一 LSTM 示意圖

$$f_t = \sigma(W_f * [h_{t-1}, X_t] + b_f) \quad (17)$$

$$i_t = \sigma(W_i * [h_{t-1}, X_t] + b_i) \quad (18)$$

$$C^t = \tanh(W_C * [h_{t-1}, X_t] + b_C) \quad (19)$$

$$C_t = f_t * C_{t-1} + i_t * C^t \quad (20)$$

$$O_t = \sigma(W_O * [h_{t-1}, X_t] + b_O) \quad (21)$$

$$h_t = O_t * \tanh(C_t) \quad (22)$$

LSTM 內部主要有三個階段，第一階段為忘記階段(Forget)，這個階段主要是對上一個節點傳進來的輸入進行選擇性忘記，也就是忘記不重要的資訊，而將重要的資訊保留，該階段會透過計算得到 f_t 作為門控來控制 C_{t-1} 需要保留那些狀態。第二階段為選擇記憶階段，該階段會將本階段的輸入值做選擇性記憶，主要是會對 X_t 選擇記憶，將重要的資訊紀錄下來。階段一與階段二結果相加則可以獲得 C_t ，也就是公式(20)。第三階段則為輸出階段，該階段會決定那些值可以成為當前狀態的輸出，主要是透過第三個 σ 來控制，並可針對第二階段得到的值透過 $tanh$ 激活函數進行變化。由於單一 LSTM 區塊對於音頻訊號來說運算會很吃力，因此我們將 5 個 LSTM 串聯來增加訓練的效能與結果的準確性。多層 LSTM 區塊連接示意圖則如圖 22 所示。

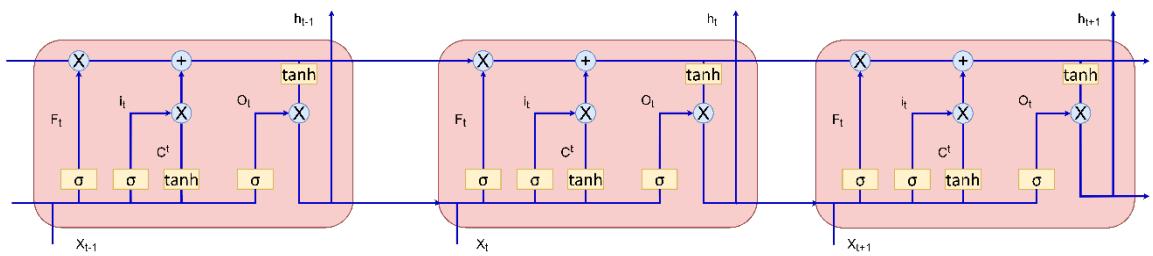


圖22. 多層 LSTM 示意圖

在[66]中所提出了 5 層 LSTM，其架構圖如圖 23 所示，在 LSTM 層中我們設定了 256 個神經元，Dropout 設定為 0.01，在最後的分類區塊中使用 Softmax 激勵函數作為分類器。Softmax 是 Sigmoid 的改良版，Sigmoid 只能用於分類兩種類別，而 Softmax 則可進行多類分類。Softmax 公式如公式 23 所示，其中 $j = 1, \dots, K$ 。

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (23)$$

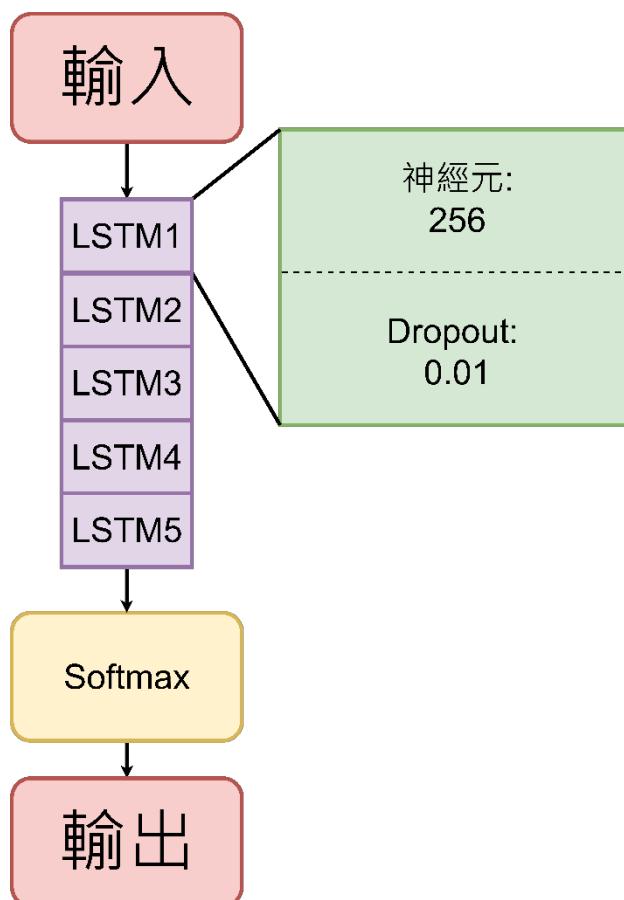


圖23. LSTM 架構圖

(四) YOLOv4

YOLOv4 演算法由 Bochkovskiy 等人[9]所提出，YOLOv4 是由 CBM、CBL、Res unit、CSPX、SPP 這五個模組做為基本組成要素。CBM 是網絡結構中最小的組件，由 Conv+Bn+Mish 激活函數組成，CBL 和 CBM 的區別在於，CBL 激活函數採用 Leaky ReLU，如圖 24 所示。

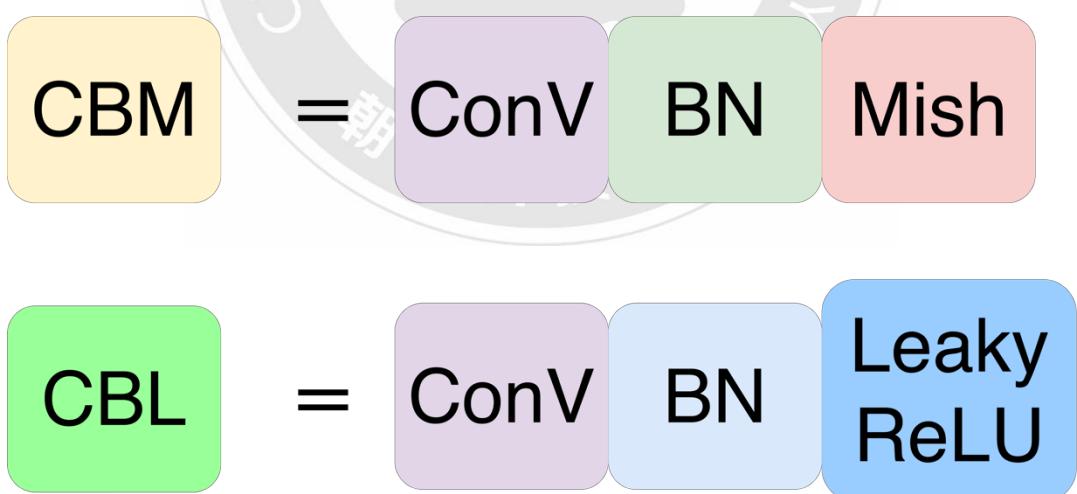


圖24 . CBM 與 CBL 架構圖

Res unit 則是基於 Resnet 網絡中的殘差結構，因此可以更深深地構建網絡，其架構圖如圖 25 所示。

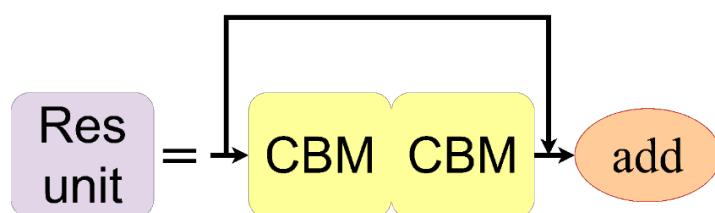
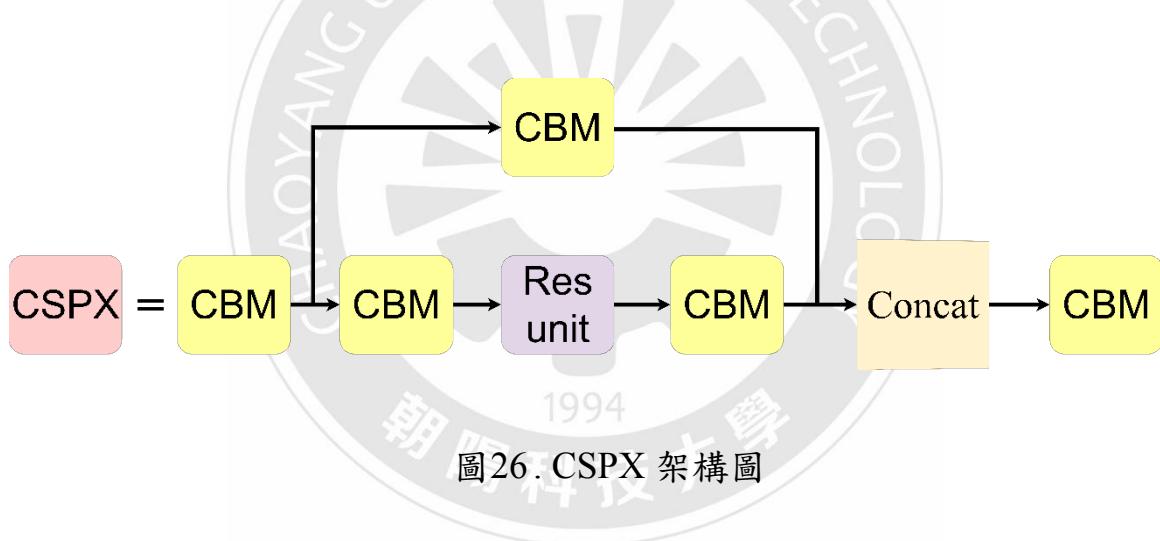


圖25 . Res unit 架構圖

CSPX 是一種基於 CSPNet 的網絡結構，由三個卷積層和 X 個 Res unit 模塊組成，其架構圖如圖 26 所示。



SPP 採用 $1*1$ 、 $5*5$ 、 $9*9$ 和 $13*13$ 的最大池化方式進行多尺度融合，而後將不同內核大小的特徵圖連接在一起作為輸出，如圖 27 所示。

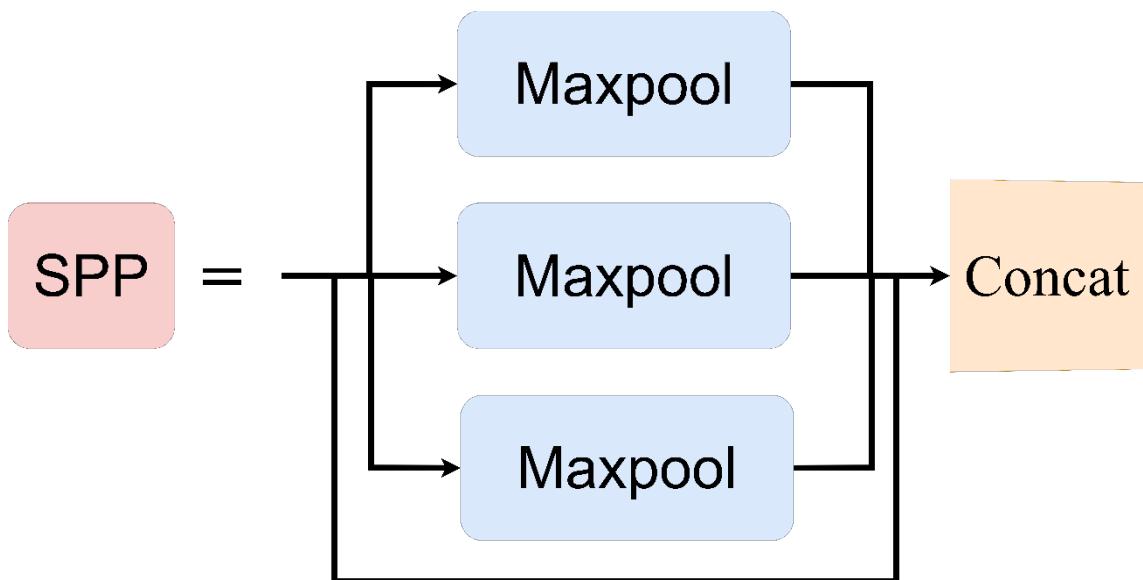


圖27. SPP 架構圖

Concat 是張量拼接，它會擴展兩個張量的維度，例如 $48*48*512$ 和 $48*48*1,024$ 。兩個張量拼接後，輸出結果是 $48*48*1536$ 。Add 是張量的加

法，不會擴展維度。例如，當 $96*96*256$ 和 $96*96*256$ 相加時，結果仍然是 $96*96*256$ 。

YOLOv4 中主幹網路(Backbone)裡卷積層數量則為每個 CSPX 中包含的卷積層數量($CSPX = 3 + 2 * X$ 個卷積層)，總共為 $2 + (3 + 2 * 1) + 2 + (3 + 2 * 2) + 2 + (3 + 2 * 8) + 2 + (3 + 2 * 8) + 2 + (3 + 2 * 4) + 1$ 共 72 層卷積層，該架構稱為 CSPDarknet53，是由 CSP 架構結合 Darknet53 所組成[18]，而 Darknet53 架構如圖 28 所示。

	Type	Filters	Size	Output
1x	Convolutional	32	3x3	256x256
	Convolutional	64	3x3/2	128x128
	Convolutional	32	1x1	
	Convolutional	64	3x3	
	Residual			128x128
2x	Convolutional	128	3x3/2	64x64
	Convolutional	64	1x1	
	Convolutional	128	3x3	
	Residual			64x64
8x	Convolutional	256	3x3/2	32x32
	Convolutional	128	1x1	
	Convolutional	256	3x3	
	Residual			32x32
8x	Convolutional	512	3x3/2	16x16
	Convolutional	256	1x1	
	Convolutional	512	3x3	
	Residual			16x16
4x	Convolutional	1024	3x3/2	8x8
	Convolutional	512	1x1	
	Convolutional	1024	3x3	
	Residual			8x8
	Avgpool		Global	
	Connected		1000	
	Softmax			

圖28 . Darknet53 架構圖

在 YOLOv4 中採用了 Bag of Freebies (BoF)和 Bag of specials (BoS)於目標檢測中。在訓練過程中經常會出現諸如數據增強、類別不平衡、成本函數和軟標記等操作以提高準確性。這些改進對推理速度沒有影響，因此稱為 BoF。BoS 是對推理速度影響較小但性能良好的動作的總稱。YOLOv4 中的 BoF 包括 CIoU-loss、CmBN、Dropblock 正規化、Mosaic 資料增強、Self-Adversarial Training(SAT)、網格靈敏度消除(Eliminate grid sensitivity)、Cosine annealing scheduler、最佳超參數(Optimal hyperparameters)和隨機訓練形狀(Random training shapes)。CIoU-loss 是為了增加標準答案的框與預測框的重疊區域並最小化中間點距離以保持兩者框的高寬一致性，其公式如公式 24~26 所示，其中 b, b^{gt} 分別表示 B, B^{gt} 的中心點， ρ 是歐式距離， c 是最小包圍兩個框的對角線長度， α 是權重函數，依照兩個框之間的距離調整， ν 是衡量長寬比的相似性， \arctan 是反正切，即為 \tan^{-1} [49]。

$$L_{CIoU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\nu \quad (24)$$

$$\nu = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (25)$$

$$\alpha = \frac{\nu}{(1 - IOU) + \nu} \quad (26)$$

CmBN 是一種修改過後的方式，一般的批量歸一化(Batch Normalization)在一個小批量(mini-batch)是以一個白化(Whiten)層當作輸入來收集樣本的均值和方差，然而當小批量的資料量過小的時候則會產出很多的噪聲，雖然有一解決方案是在多個小批量資料去估算他們，但由於權重是會在每個迭代中不斷變化，因此在此權重下收集的統計訊息在新的權重下可能變得不準確，也就是說取所有的 mini-batch 的平均是錯誤的，但也由於權重是會一直變化的，因此在交叉迭代批量歸一化(Cross-Iteration Batch Normalization, CBN)使用了公式 27~29 調整來基於 k 個之前的迭代估計這些統計數據，YOLOv4 中的 CmBN 是基於 CBN 去做修改，在 mini-batch 之間不做更新計算，而是在一個 batch 做完後才去更新網路參數。圖 29 為 CBN 示意圖 [49]。

$$\bar{u}_{t,k}^l(\theta_t) = \frac{1}{k} \sum_{\tau=0}^{k-1} u_{t-\tau}^l(\theta_t) \quad (27)$$

$$\bar{v}_{t,k}^l(\theta_t) = \frac{1}{k} \sum_{\tau=0}^{k-1} \max[u_{t-\tau}^l(\theta_t), u_{t-\tau}^l(\theta_t)^2] \quad (28)$$

$$\bar{\sigma}_{t,k}^l(\theta_t) = \sqrt{\bar{v}_{t,k}^l(\theta_t) - \bar{u}_{t,k}^l(\theta_t)^2} \quad (29)$$

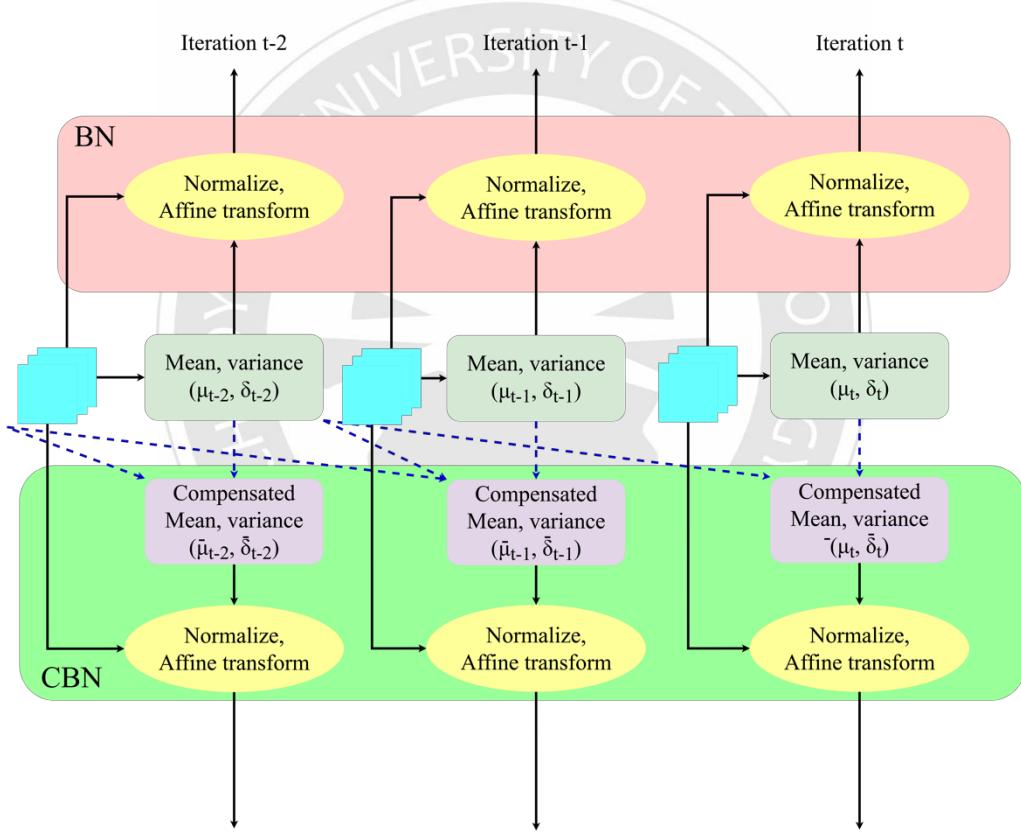
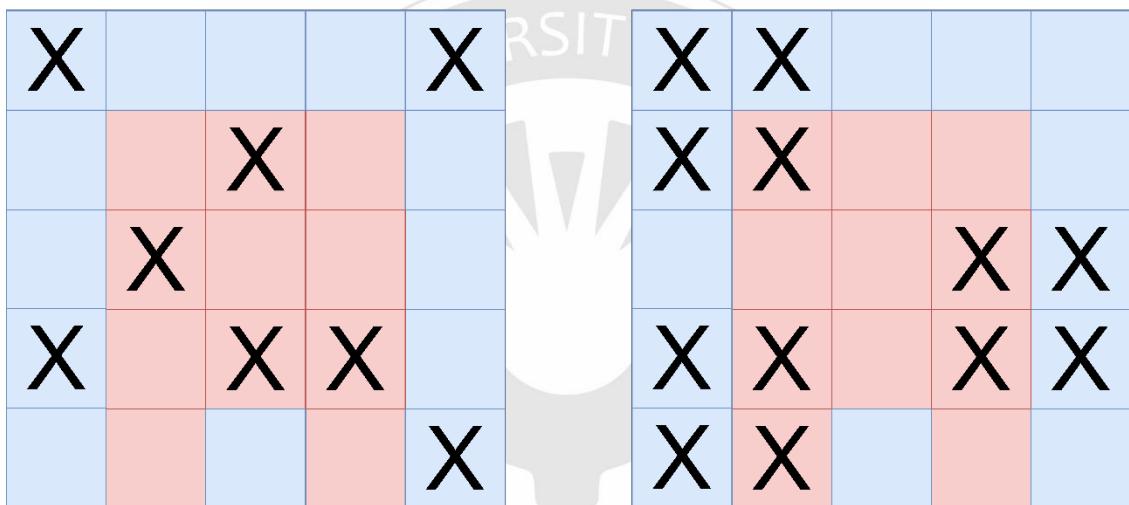


圖29. CBN 架構圖

Dropblock 是用於代替 Dropout 的方式。傳統的 Dropout 會透過隨機的方式使神經元不加入本次迭代的訓練，以此防止過擬合發生。由於卷積神經網路中的卷積層對於隨機丟棄的方式並不敏感，這是因為卷積層通常都與池化層、全連接層同時使用，即使隨機丟棄，卷積層仍然可以從相鄰的激活單元學習到相同的信息，使用傳統 Dropout 效益不大，因此在 YOLOv4 中使用 Dropblock，該方法與傳統 Dropout 作法差不多，差異在於傳統 Dropout 是採用隨機丟棄的方式丟棄神經元，而 Dropb\l Block 則是根據被丟棄的神經元與相鄰神經元一起丟棄，如圖 30 所示[49]。



Dropout 1994 Dropblock

圖30. Dropblock 示意圖

Mosaic 資料增強是隨機取 4 張照片進行翻轉、縮放、色域變化等操作後拼接成 1 張照片，這 4 張照片都有各自對應的框。拼接成 1 張照片後則同時獲得 4 張照片的框，並將這 1 張照片進行訓練，即等同於 1 次放 4 張照片進行訓練，如圖 31 所示[49]。

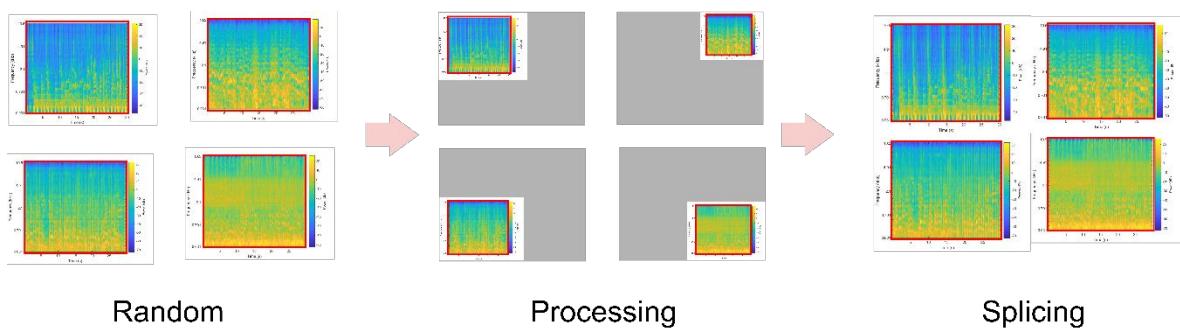


圖31. Mosaci 資料增強示意圖

自對抗訓練(Self Adversarial Training, SAT)是一種資料增強的技術，首先會在訓練樣本上執行一次正向傳播，在傳統步驟中，會在反向傳播過程中調整模型的權重來提升檢測器的能力，但這步驟在 SAT 中則是相反。它會修改圖像使其能在最大程度上降低檢測器的性能，即創建以當前模型為目標的對抗攻擊，並使用這張新圖與原始的邊界框和類別標籤來訓練該模型，這可以提升模型的泛化力並降低過擬合[49]。

網格靈敏度消除(Eliminate grid sensitivity)是使檢測結果中心的若接近網格(Grid)時增加學習準確率的方法。在修正中心點時若需修正至如圖 32 紅點的網格邊緣是較為困難的。由於 Sigmoid 函數的梯度，使得 Sigmoid 在網格邊緣時導數接近 0，容易發生梯度消失，導致中心點在 Grid 邊緣時很難學習到準確的結果，因此調整 Sigmoid 函數，將 Sigmoid 乘以一個大於 1 的參數來解決這個問題，如公式 30 所示，其中 α 是大於並接近 1 的值。

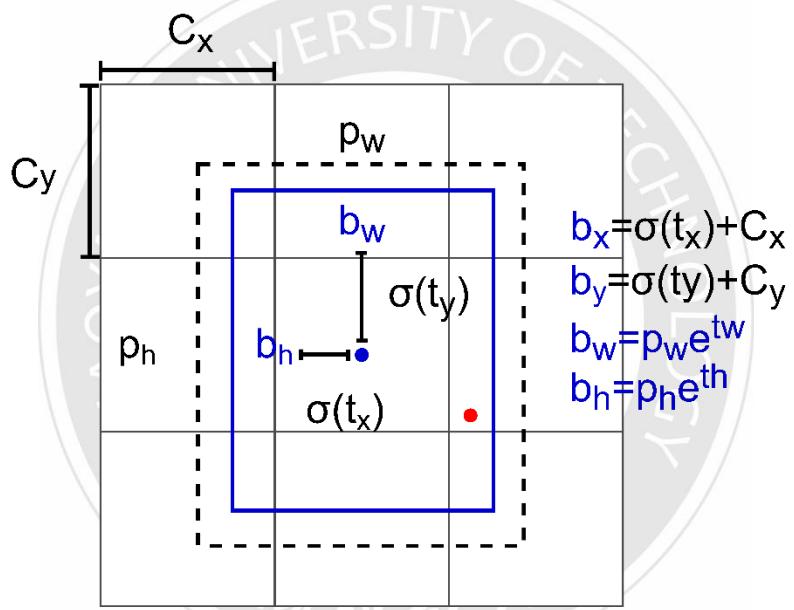


圖 32 . Sigmoid 與 Differential Sigmoid

$$b_x = \alpha\sigma(t_x) - (\alpha - 1) / 2 + C_x \quad (30)$$

餘弦退火調度器(Cosine annealing scheduler)是利用餘弦函數來調整學習率，會先緩降學習率，中途再加速下降，最後再次地緩降。調度器有兩種，分別為 Cosine Annealing 與 Cosine Annealing LR。其中 Cosine Annealing 公式如公式 31 所示，圖 33 為其函數圖，其中 η_t 是指學習率， η_{max} 是指最大學習率， η_{min} 是指最小學習率， T_{cur} 是指當前迭代次數， T_{max} 是指最大迭代次數。

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right) \quad (31)$$

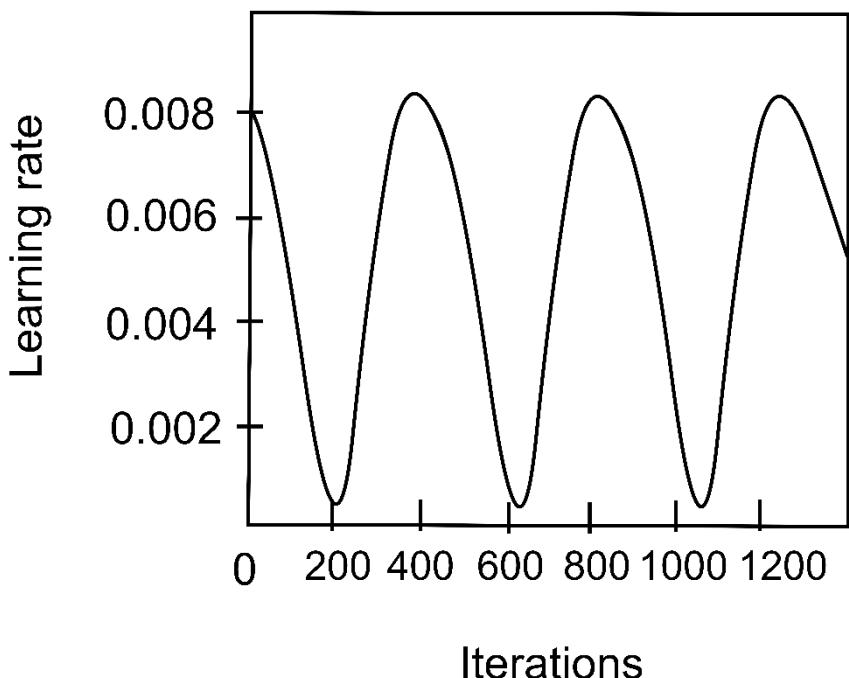


圖33. Cosine Annealing 函數圖

Cosine Annealing LR 公式如公式 32 所示，圖 34 為其函數圖，其中 η_t 是指學習率， η_{max} 是指最大學習率， η_{min} 是指最小學習率， T_{cur} 是指重啟後的迭代次數， T_i 是指第 i 輪的迭代次數。

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right) \quad (32)$$

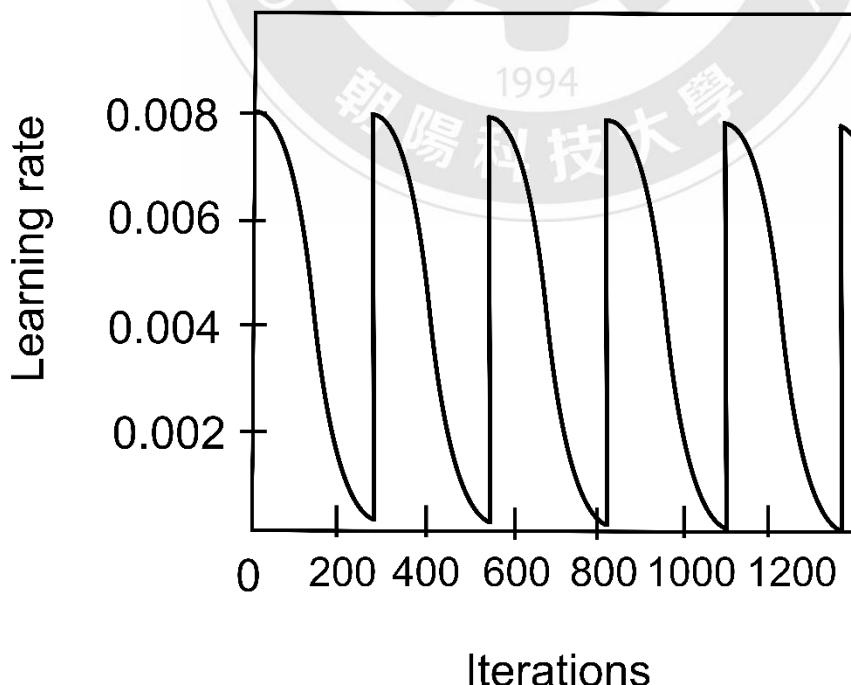


圖 34. Cosine Annealing LR 函數圖

最佳超參數 (Optimal hyperparameters) 是使用基因演算法 (Genetic Algorithms, GA) 選擇超參數。首先隨機組合超參數進行訓練，後挑選出最好的 10% 超參數在隨機組合進行訓練，最後選擇結果最佳的模型。

隨機訓練形狀(Random training shapes)是 YOLOv4 每隔一段 Epoch 會變更輸入圖片大小，以減少過擬合。

BoS 則包含了 Mish 激活、修改版 SPP 模組、修改版 SAM 模組、修改版 PAN 路徑聚合模組、DIOU-NMS。其中修改版 SPP 模組分別在 5×5 、 7×7 、 13×13 內合中進行最大池化(13×13)，在將 3 個結果與原始輸入進行結合，目的是增加視野(Receptive Field)，如圖 35 所示[49]。

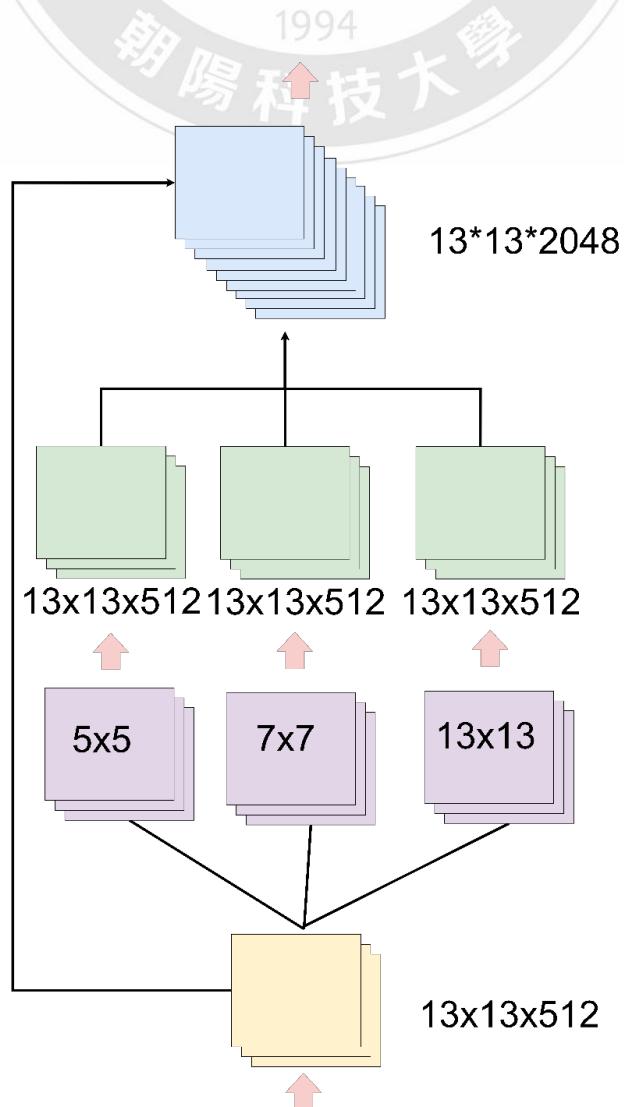


圖 35. 修改版 SPP 模組示意圖

修改版 SAM 模組是使輸入通過 CNN 與 Sigmoid 函數在與原本的輸入相乘，如圖 36 所示。

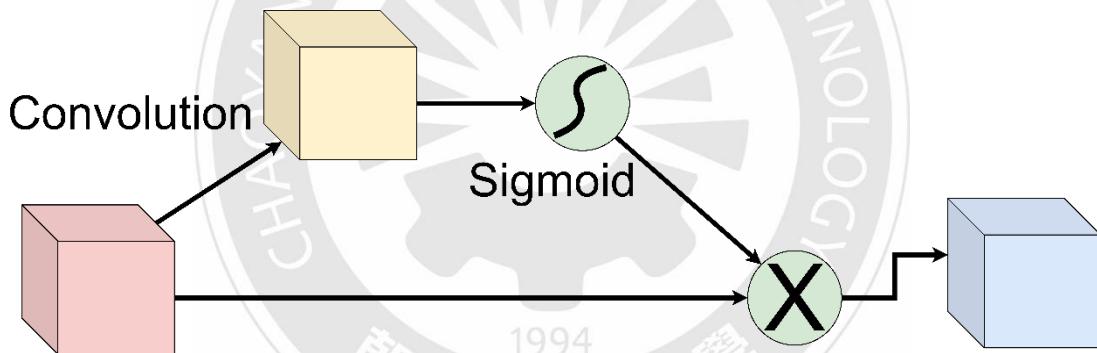


圖36. 修改版 SAM 模組示意圖

修改版 PAN 模組是為了使模型可以保留更多的淺層資訊，與一般的 PAN 相比，YOLOv4 在連接時使用 Concatenation 而非相加的方式進行合併。如圖 37 所示。

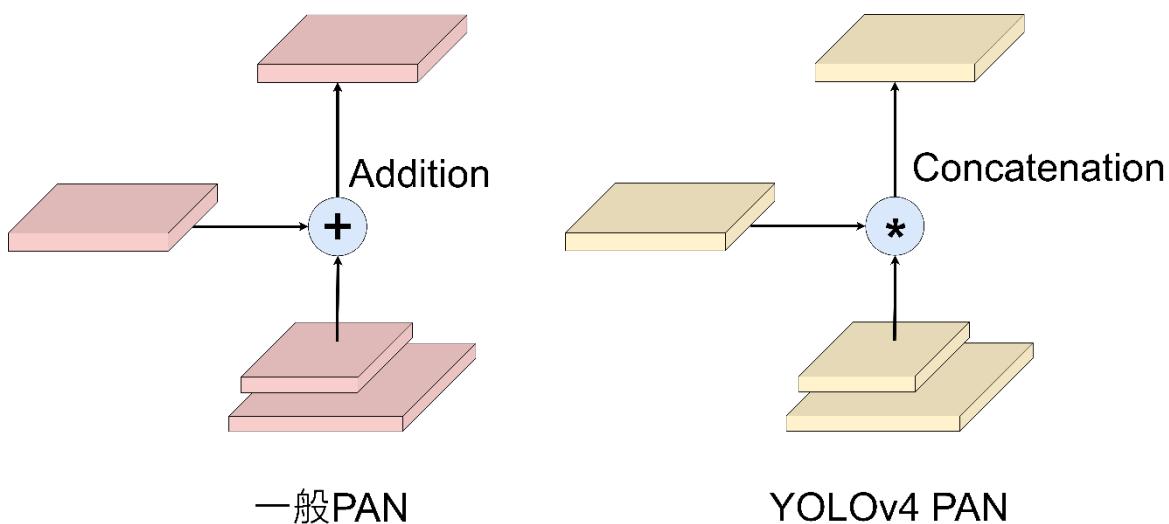


圖37. 修改版 PAN 與一般 PAN 差異示意圖

DIoU-NMS 可以使模型能更加穩健地應對有遮擋的情況，非極大值抑制(Non-Maximum Suppression, NMS) 可以過濾掉其它預測相同目標的邊界框並保留置信度最高的邊界框，而 DIoU 是用做 NMS 的一個因子，該方法在抑制冗餘的邊界框時會使用 IoU 和兩個邊界框的中心點之間的距離[49]。

四、評分指標

(一) 混淆矩陣

混淆矩陣(Confusion Matrix) 是一種可視化的模型分類準確度衡量工具，如圖 38 所示，其中 True Positive(TP)是真陽性，表示實際為真，預測為真。True Negative(TN)是真陰性，表示實際為假，預測為假。False Positive(FP)是偽陽性，表示實際為假，預測為真。False Negative(FN)是偽陰性，表示實際為真，預測為假。

Confusion Matrix		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP	FP
	Negative	FN	TN

圖38. 混淆矩陣示意圖

(二) 精確率

精確率(Precision)是用於判斷模型中預測為真而實際正確的比例，其公式如公式 33 所示。精確率越高代表模型區分陰性樣本的能力越強。

$$Precision = \frac{TP}{TP + FP} \quad (33)$$

(三) 召回率

召回率(Recall)是用於判斷模型中陽性樣本被正確識別為正的比例，其公式如公式 34 所示。召回率越高代表模型區分陽性樣本的能力越強。

$$Recall = \frac{TP}{TP + FN} \quad (34)$$

(四) F1 分數

F1 分數(F1-score)是精確率和召回率的調和平均值(Harmonic mean)，是衡量分類模型準確率的指標，其公式如公式 35 所示。F1 分數越高代表模型越穩健。

$$F1 - score = \frac{2 * (Recall * Precision)}{Recall + Precision} \quad (35)$$

(五) 準確率

準確率(Accuracy)是常見的評分指標，其公式如公式 36 所示。在資料集平衡時可以有效評估模型好壞，但若資料集不平衡則可能出現準確率異常高。因此通常不會只使用準確率作為評分標準。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (36)$$

(六) mAP

mAP 是 mean Average Precision 的縮寫，Average Precision(AP)是 PR 曲線下的面積，若 AP 越高則代表分類器越好。而 mAP 是指每個類別的 AP 的平均，mAP 越高代表分類器越好，其公式如公式 37 所示。其中 $classes$ 表示類別數量， $TP(c)$ 為針對類別 c 的 TP， $FP(c)$ 為針對類別 c 的 FP。

$$mAP = \frac{1}{|classes|} * \sum_{c \in classes}^{1994} \frac{|TP(c)|}{|TP(c)| + |FP(c)|} \quad (37)$$

肆、結果與討論

一、卷積神經網路實驗結果

在 CNN 中我們使用梅爾頻譜作為模型的輸入，首先將 GTZAN 中的音訊轉換成 MFCC 訊號，其中窗函數使用漢明窗並設定尺寸為 1,024 帀。重疊區域設定為 512，傅立葉變換尺寸設定為 4,096 帀，梅爾濾波器設定 64 個，如表 4 所示。

表4.CNN 方法轉換梅爾頻譜之參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帀
重疊區域數	512 帀
梅爾濾波器數量	64 個
傅立葉變換尺寸	4,096 帀

轉換成梅爾頻譜訊號後送入 CNN 模型進行訓練，設定 5 層 CNN 層，每層的激勵函數都使用 ReLU，池化層選用最大池化層，Dropout 設定為 0.5，Epochs 設定為 2,500 次，Batch size 設定為 128，並將資料集拆分為訓練集 70%，驗證集 20%，測試集 10%。圖 39 為 CNN 方法訓練與測試的準確率曲線圖。圖 40 為訓練與測試的損失曲線圖。圖 41 為混淆矩陣。在 CNN 方法中得到的 Accuracy 為 73.1%，Precision 為 65.1%，Recall 為 61.7%，F1 分數為 61.3%，其結果統計在表 5 中。

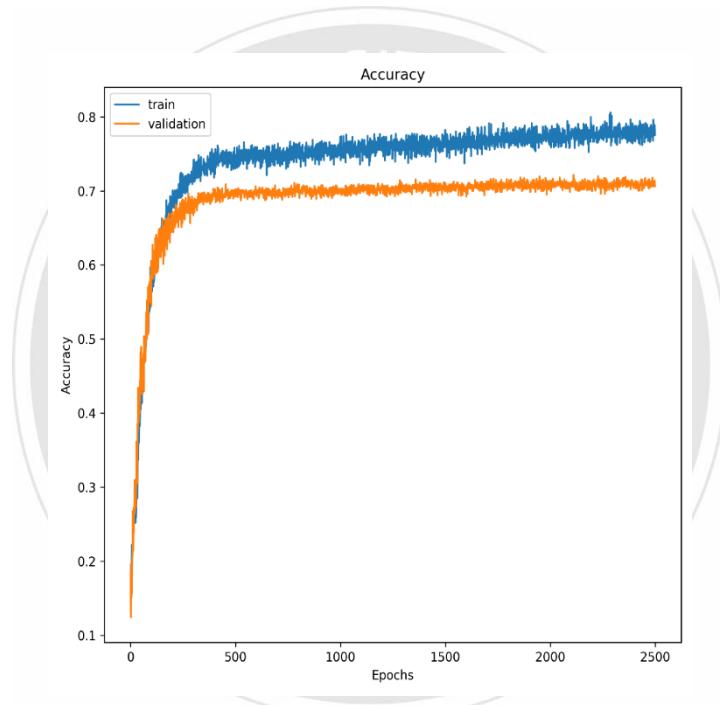


圖39. CNN 方法訓練與測試的準確率曲線圖

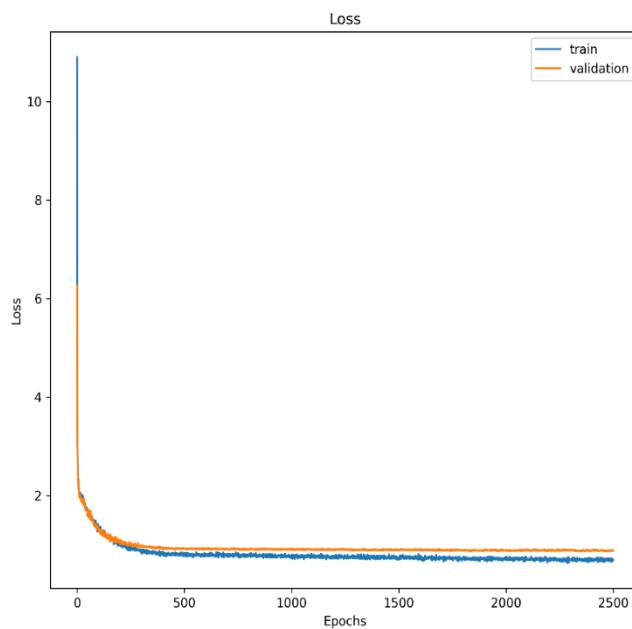


圖40. CNN 方法訓練與測試的損失曲線圖

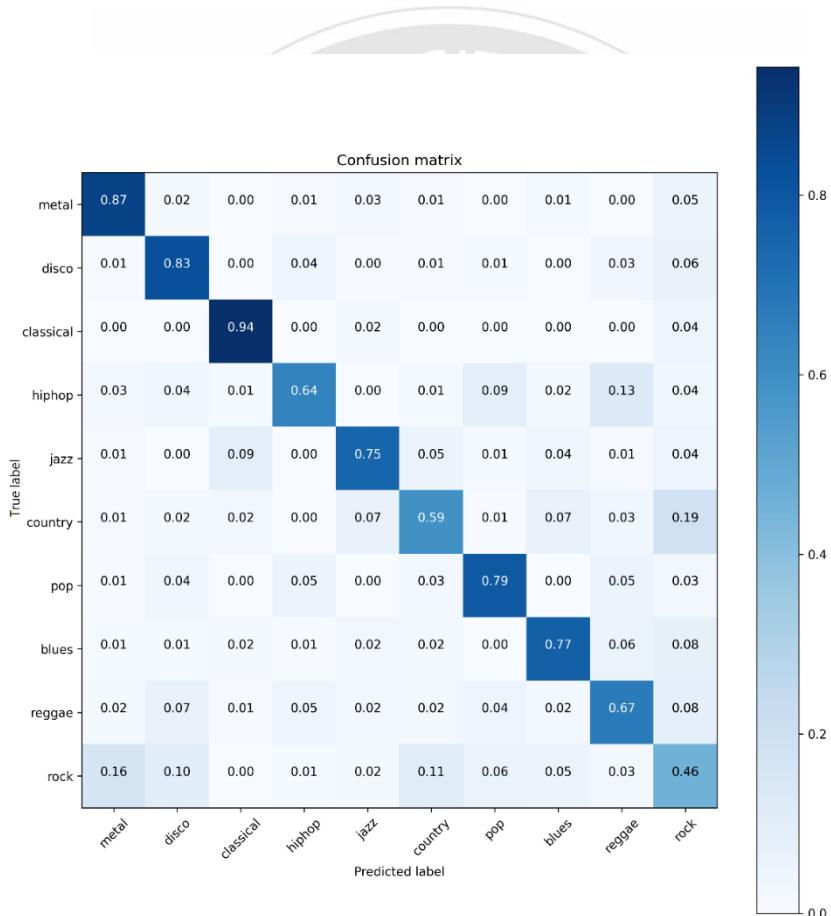


圖41. CNN 方法之混淆矩陣圖

表5. CNN 方法結果統計

評分方式	結果
Accuracy	73.1%
Precision	65.1%
Recall	61.7%
F1-score	61.3%

由圖 39 與圖 40 可以看出，使用基礎的 CNN 層並堆疊至 5 層，搭配 ReLU 以及最大池化的方式可獲得 73.1% 的準確率。從圖 39 可以看出，模型在訓練時的準確率僅接近 80%，而圖 40 的訓練損失曲線下降趨勢已經趨於平穩，這代表使用基礎 CNN 方法的極限位置，因此在圖 39 與圖 40 的測試曲線可以看到，在曲線中最高頂點 70% 附近起伏，而測試的損失曲線趨於平穩且貼近訓練的損失曲線也代表 CNN 方法的極限值已經出現。而從圖 41 的損失曲線可以看出，CNN 方法在分類 Hip-hop、Country、Reggae、Rock 出現較低的正確率，其中 Hip-hop 有較大比例被分類成 Reggae，Country 大部分比例被分類成 Rock，Reggae 大部分比例被分類成 Hip-hop，Rock 大部分比例被分類成 Country，其中原因是因為如文獻探討中提及的 Hip-hop 音樂是起源於 Reggae 且大部分受到 Reggae 音樂的影響，這使得 Hip-hop 與 Reggae 是很相似的。Country 音樂是 Rock 音樂的祖先，早期的 Rock 音樂常常具有 Country 音樂的靈魂，直到近代 Rock 音樂才漸漸脫離 Country 音樂的範圍，區分 Country 音樂與 Rock 音樂除了依照使用樂器外，也可以從歌詞去區分兩者。

二、卷積循環神經網路實驗結果

在 CRNN 中我們使用梅爾頻譜作為模型的輸入，首先將 GTZAN 中的音訊轉換成 MFCC 訊號，其中窗函數使用漢明窗並設定尺寸為 1,024 帀。重疊區域設定為 512，傅立葉變換尺寸設定為 4,096 帀，梅爾濾波器設定 64 個，如表 6 所示。

表6. CRNN 方法轉換梅爾頻譜之參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帀
重疊區域數	512 帀
梅爾濾波器數量	64 個
傅立葉變換尺寸	4,096 帀

轉換成梅爾頻譜訊號後送入如圖 20 之 CRNN 模型進行訓練，設定 4 層 CNN 層，每層的激勵函數都使用 Leaky ReLU，並在最後一層 CNN 層之後添加一層 RNN 層，池化層選用最大池化層，Dropout 設定為 0.5，Epochs 設定為 2,500 次，Batch size 設定為 16，並將資料集拆分為訓練集 70%，驗證集 10%，測試集 20%。圖 42 為 CRNN 方法訓練與測試的準確率曲線圖。圖

43 為訓練與測試的損失曲線圖。圖 41 為 CRNN 方法的混淆矩陣。在 CRNN 方法中得到的 Accuracy 為 68%，Precision 為 69.6%，Recall 為 67.9%，F1 分數為 67.3%，其結果統計在表 7 中。

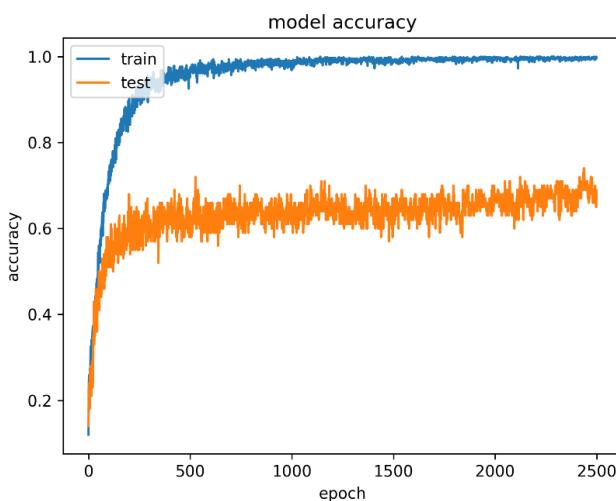


圖42. CRNN 訓練與測試的準確率曲線圖

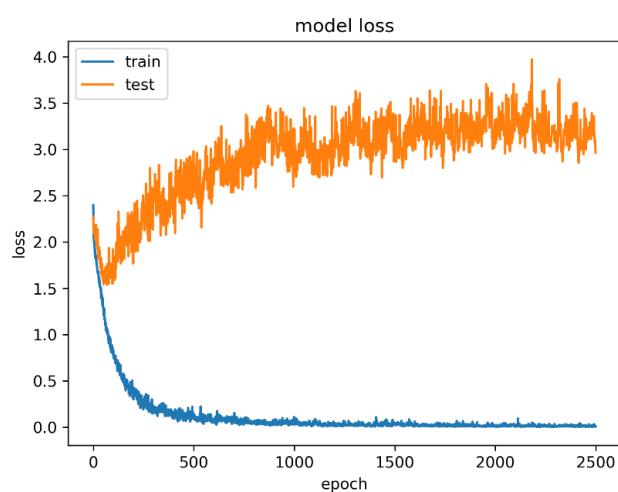


圖43. CRNN 訓練與測試的損失曲線圖

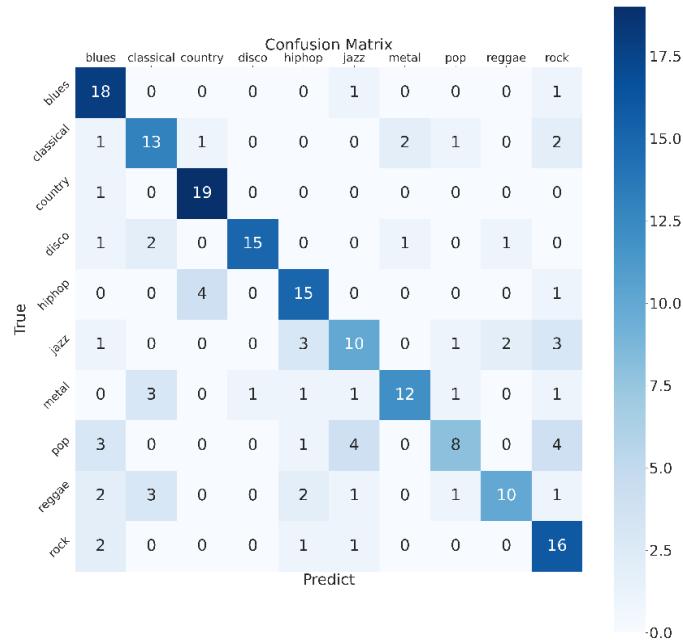


圖44. CRNN 方法之混淆矩陣圖

表7. CRNN 方法結果統計

評分方式	結果
Accuracy	68%
Precision	69.6%
Recall	67.9%
F1-score	67.3%

從圖 42 可以看出，CRNN 方法在訓練時的準確率可以接近 100%，而測試的準確率則為 68%，而圖 43 可以看到在訓練時的損失率呈現一個有弧度的下降曲線，但在測試時的損失率則逐步上升，這樣的結果表示模型出現過擬合，才會使訓練損失曲線下降而測試損失曲線上昇。我們嘗試添加正則化、修改 Dropout、調整 Batch size 都無法降低過擬合的影響，因此在未來將會針對 CRNN 的過擬合狀況進行修正。而在圖 44 的混淆矩陣中可以看出，CRNN 在分類上與 CNN 是差不多的，有差異的地方在於，CNN 方法在分類 Hip-hop、Country、Reggae、Rock 等曲風時較弱，而 CRNN 在分類 Jazz、Pop、Reggae 等曲風時較弱，其共同之處在於 Reggae 曲風有大部分被分類至 Hip-hop 音樂中，這也可以表示 Reggae 音樂與 Hip-hop 音樂的血緣是非常接近的，尤其是 Reggae 音樂對 Hip-hop 音樂的影響之重。

三、長短期記憶模型實驗結果

在 LSTM 中我們使用 MFCC 作為模型的輸入，首先將 GTZAN 中的音訊轉換成 MFCC 訊號，其中窗函數使用漢明窗並設定尺寸為 1,024 帀。重疊區域設定為 512，傅立葉變換尺寸設定為 2,048 帀，梅爾濾波器設定 13 個，如表 8 所示。轉換成 MFCC 訊號後送入如圖 23 之 LSTM 模型進行訓練，設定 5 層 LSTM 層，每層都使用 256 個神經元並設定 Dropout 都為 0.01，Epochs 設定為 2,500 次，Batch size 設定為 128，並將資料集拆分為訓練集 70%，驗證集 26%，測試集 4%。圖 45 為 LSTM 方法訓練的準確率與損失曲線圖。圖 46 為驗證的準確率與損失曲線圖。圖 47 為 LSTM 方法的混淆矩陣。在 LSTM 方法中得到的 Accuracy 為 42.5%，Precision 為 49.78%，Recall 為 42.5%，F1 分數為 41.1%，其結果統計在表 9 中。

表8. LSTM 方法轉換 MFCC 之參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帀
重疊區域數	512 帀
梅爾濾波器數量	13 個
傅立葉變換尺寸	2,048 帀

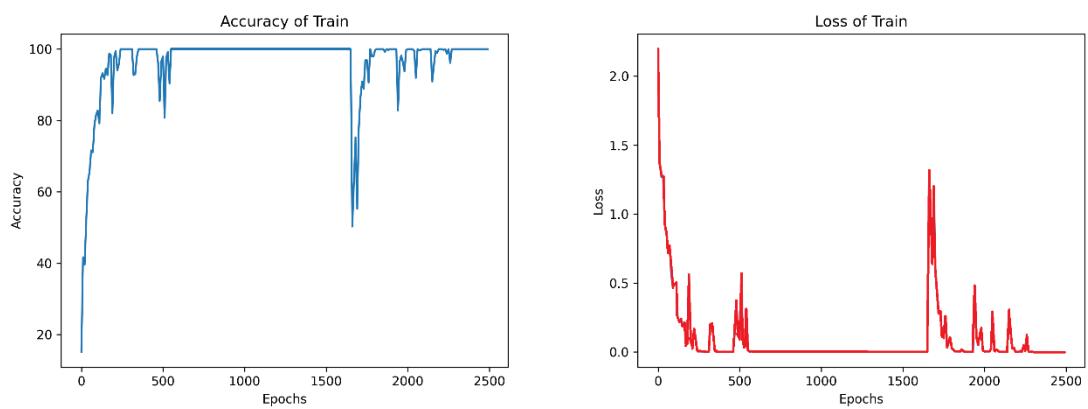


圖45. LSTM 訓練準確率與損失曲線圖

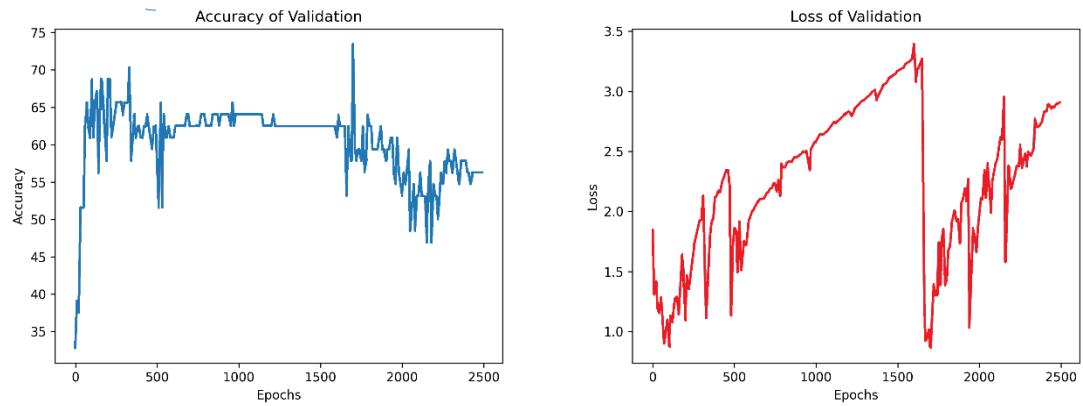


圖46. LSTM 驗證準確率與損失曲線圖

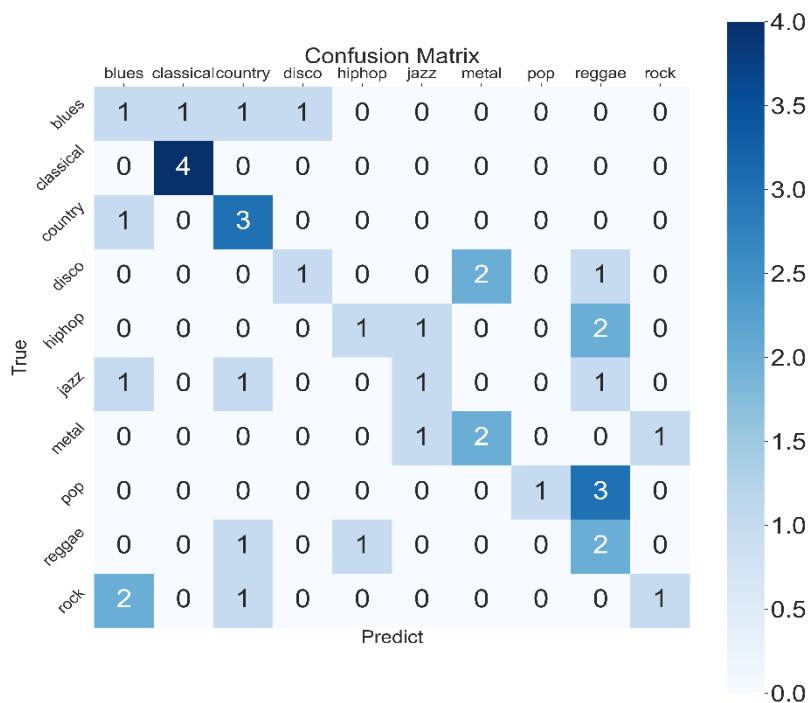


圖47. LSTM 方法之混淆矩陣圖

表9. LSTM 方法結果統計

評分方式	結果
Accuracy	42.5%
Precision	49.78%
Recall	42.5%
F1-score	41.1%

在圖 45 中可以看出，模型在訓練時在前期呈現上漲趨勢至 Epochs 為 500 時，但在 Epochs 持續增加時出現一段平穩後出現跌幅，等於是模型又重新訓練的狀況發生，從損失曲線也可以發現相同的情況發生，因此在訓練時的平均準確率僅 68%。而在驗證時的平均準確率為 68%，從圖 46 的損失曲線可以發現模型出現過擬合，即訓練損失率下降而驗證損失率上升，與 CRNN 方法出現相似的情況，因此此情況成為在未來研究中必須解決的問題。而從圖 47 的混淆矩陣可以看出，僅使用 LSTM 方法在測試集的結果並不理想，其中分類較為精準的曲風為 Classical 與 Country，兩者都有著 75% 以上的分類準確率，而 Metal 與 Reggae 則有 50% 的分類準確率。LSTM 方法一直都存在三個缺點，即並行處理存在劣勢、對於較長序列資料存在缺陷、訓練佔用資源多且訓練時間較長，而由於音樂屬於長時間序列資料，因此若只使用 LSTM 方法而不添加其他神經網路架構或分類器的話，單靠 LSTM 是較為吃力的。

四、可視化短時距傅立葉變換實驗結果

在可視化 STFT 方法中我們使用經過轉換過的圖片進行訓練。首先將 30 秒音訊透過 MATLAB 轉換成可視化頻譜圖，其中窗函數使用漢明窗並設定尺寸為 1,024 帀。重疊區域設定為 512，傅立葉變換尺寸設定為 4,096 帀，如表 10 所示。

表10. 可視化 STFT 方法轉換之參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帀
重疊區域數	512 帀
傅立葉變換尺寸	4,096 帀

將轉換後的可視化 STFT 頻譜送入 YOLOv4 中進行訓練，並設定 Epochs 次數為 20,000 次，Batch size 設定為 64，輸入長寬為 416*416，並將資料集拆分為訓練集 70%，驗證集 10%，測試集 20%，且進行 10 次實驗取得平均時以達到較準確的實驗結果，因此評分方式的結果都是 10 次實驗的平均結果。

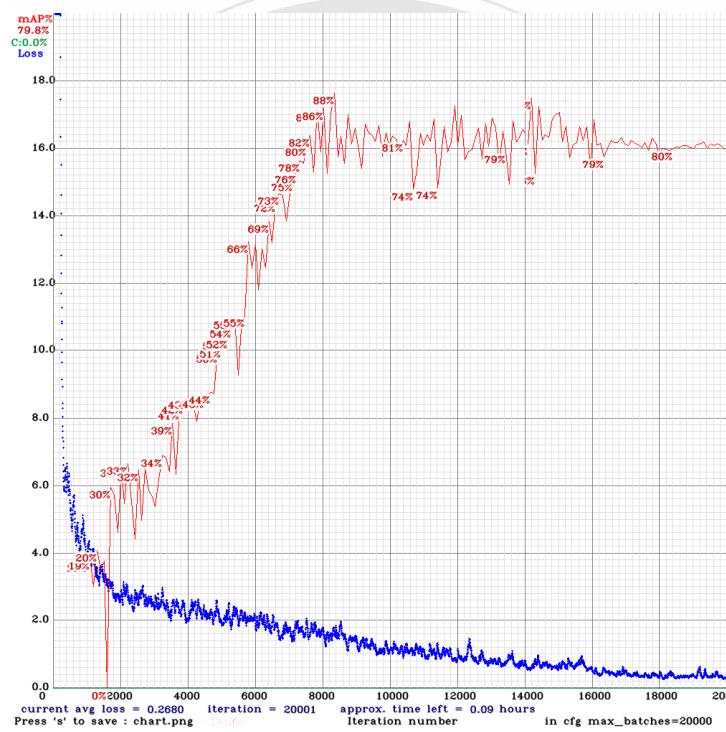


圖48. 可視化 STFT 頻譜圖方法第一次實驗之 mAP 與損失曲線圖

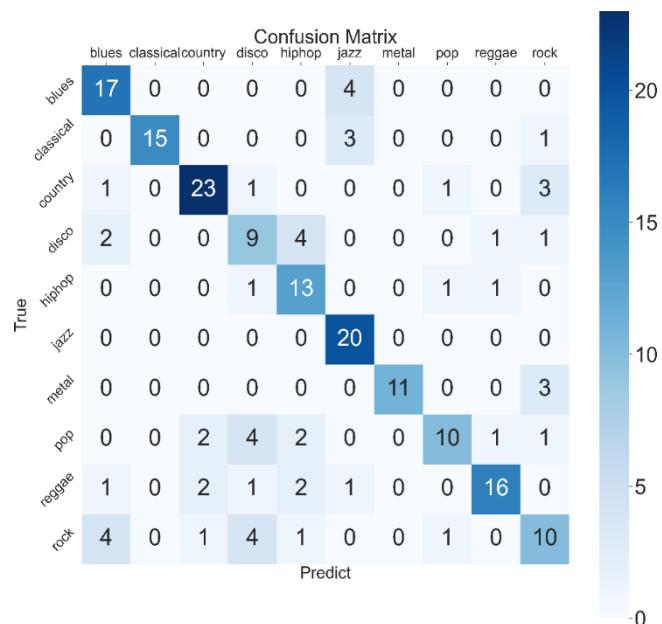


圖49. 可視化 STFT 頻譜圖方法第一次實驗之混淆矩陣圖

圖 48 為第一次可視化 STFT 頻譜圖方法的 mAP 與 Loss 曲線，其中 mAP 為 79.8%，平均損失率為 0.2680。圖 49 為第一次可視化 STFT 頻譜圖方法的混淆矩陣，其中 Accuracy 為 72%，Precision 為 72%，Recall 為 81%，F1 分數為 76%，測試集 mAP 為 87.38%。從圖 48 可以看出 mAP 曲線是非線性上升的，有時會出現下降但又馬上上升，這說明了 Dropblock 起到的作用，使得電腦在學習過程中降低了過擬合的發生機率，且這樣上下波動也表示模型是穩定學習而非一次性全部學完，這也使得模型的泛化性會提升。平均損失率下降到 0.2680，在 Epochs 為 16,000 趨於平緩，同時 mAP 曲線也於趨於平緩，為了驗證該情況是否為偶發狀況，因此我們進行 10 次實驗以進行測試。

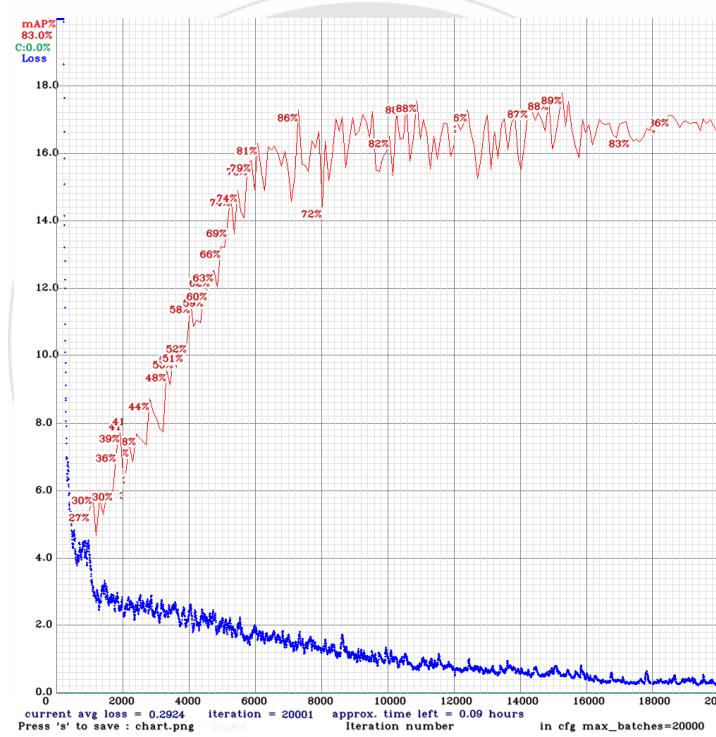


圖50. 可視化 STFT 頻譜圖方法第二次實驗之 mAP 與損失曲線圖

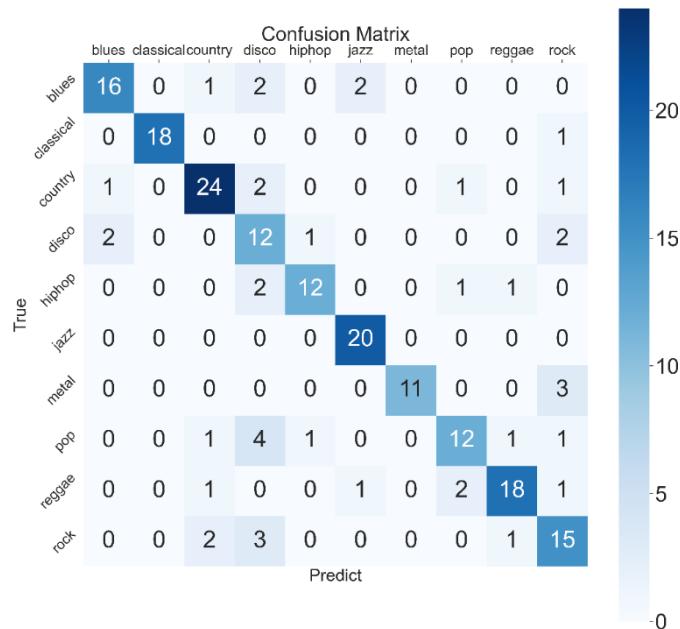


圖51. 可視化 STFT 頻譜圖方法第二次實驗之混淆矩陣圖

圖 50 為第二次實驗 mAP 與 Loss 曲線圖，其中 mAP 為 83%，平均損失率為 0.2924。圖 51 為第二次實驗之混淆矩陣，其中 Accuracy 為 79%，Precision 為 75%，Recall 為 82%，F1 分數為 78%，測試集 mAP 為 85.56%。從圖 50 可以看出，相較於第一次實驗初期的突發下降狀況，在第二次實驗中並沒有出現，也因此第二次實驗 mAP 也相較第一次實驗來的高，也因此使得損失率更低。在圖 51 中可以看出，在測試集中得出的結果與第一次實驗的結果差異不大，在分類 Disco、Metal、Pop 與 Rock 音樂時也較第一次實驗分類來的更精確，這也與第一次實驗初期的突發狀況有關，該狀況導致模型需要花費更多的 Epochs 次數才能將準確率提升，而在 Epochs 次數皆為固定情況下導致第一次實驗的 mAP 較低，其分類精確度也較低。

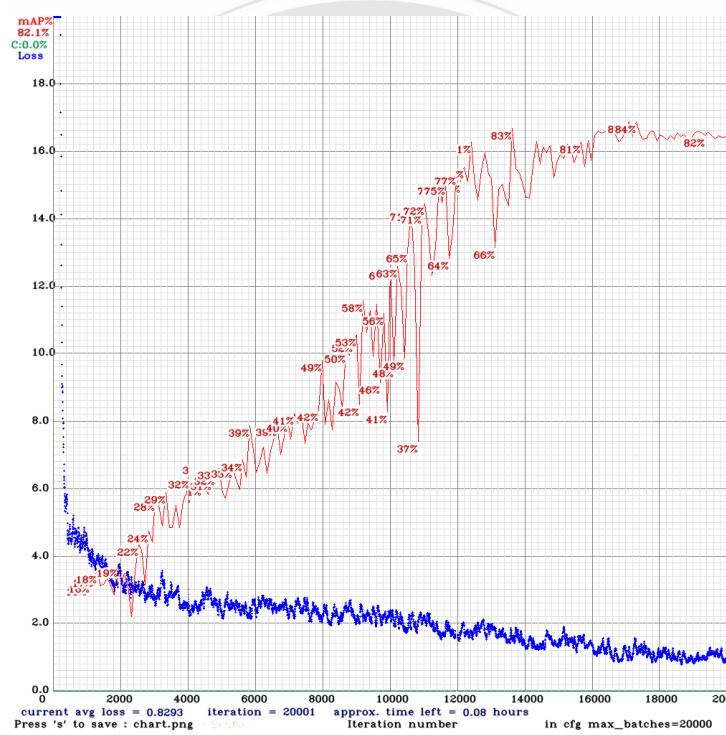


圖52. 可視化 STFT 頻譜圖方法第三次實驗之 mAP 與損失曲線圖

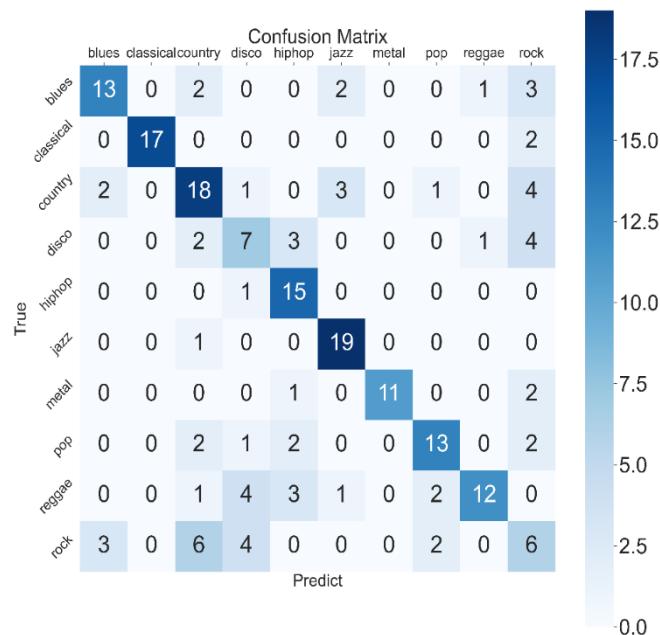


圖53. 可視化 STFT 頻譜圖方法第三次實驗之混淆矩陣圖

圖 52 為第三次實驗的 mAP 與 Loss 曲線，其中 mAP 為 82.1%，平均損失率為 0.8293。圖 53 為第三次實驗之混淆矩陣，其中 Accuracy 為 65.5%，Precision 為 69%，Recall 為 81%，F1 分數為 75%，測試集 mAP 為 84.3%。從圖 52 可以看出，在 Epochs 為 10,000 至 11,000 之間出現巨大起伏，一度跌至 37%，這也許是 Dropblock 在隨機丟棄時剛好丟棄到重要資訊導致模型需要在後續 Epochs 中學習回這些資訊，因此出現巨大起伏，而此起伏也導致損失率為 0.8293，且沒有出現前面實驗中的平緩。而從圖 53 可以看出，在測試集測試分類時相較於實驗二來的不準確，其中 Disco 與 Rock 音樂被分類到並無較相似的曲風中，這也與巨大起伏的發生有著密切的關係。

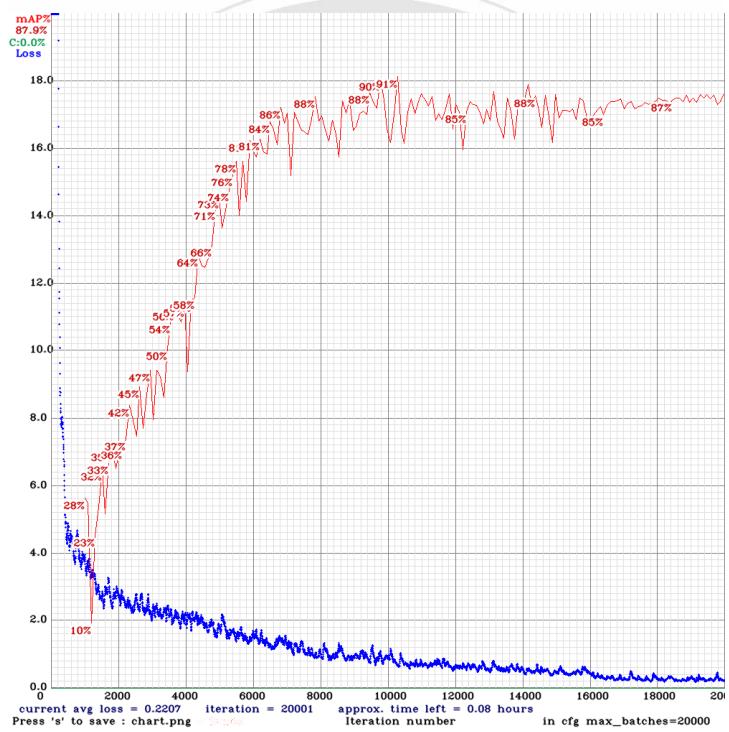


圖54. 可視化 STFT 頻譜圖方法第四次實驗之 mAP 與損失曲線圖

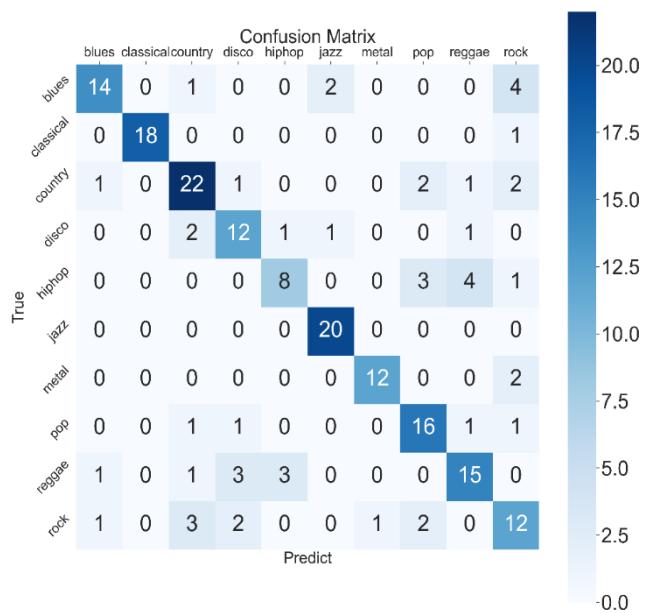


圖55. 可視化 STFT 頻譜圖方法第四次實驗之混淆矩陣圖

圖 54 為第四次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 87.9%，平均損失率為 0.2207。圖 55 為第四次實驗之混淆矩陣，其中 Accuracy 為 74.5%，Precision 為 76%，Recall 為 87%，F1 分數為 81%，測試集 mAP 為 90.55%。第四次實驗是 STFT 方法獲得最高準確率的實驗，從圖 54 可以看出，在訓練時的 mAP 曲線沒有出現前次實驗的問題，因此呈現平穩上升，而損失曲線則在 Epochs 為 16,000 時趨於平穩，使得這次實驗沒有出現太多意外。從圖 55 可以看出模型在分類 Hip-hop 音樂時較不準確之外其餘音樂曲風在分類上都有一定的準確性。

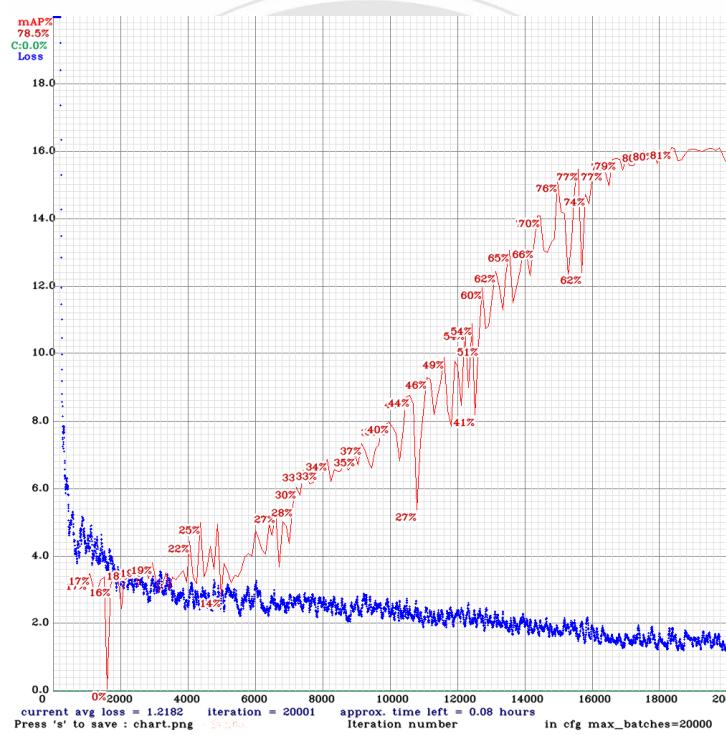


圖56. 可視化 STFT 頻譜圖方法第五次實驗之 mAP 與損失曲線圖

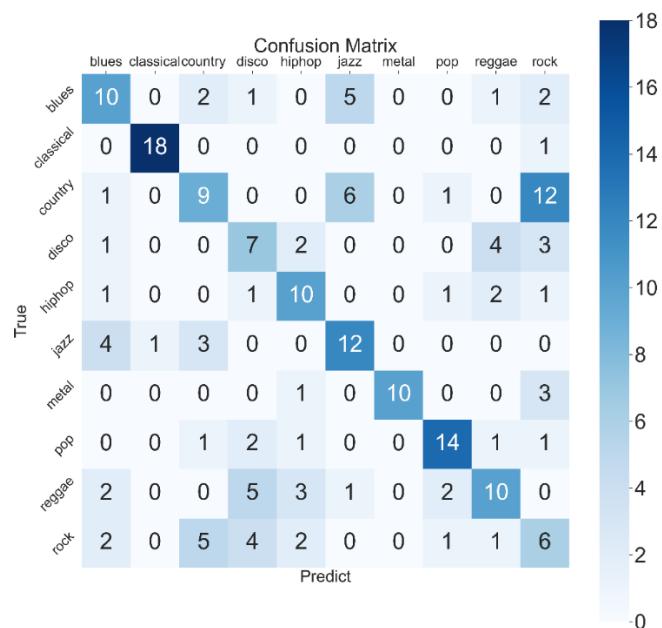


圖57. 可視化 STFT 頻譜圖方法第五次實驗之混淆矩陣圖

圖 56 為第五次實驗的 mAP 與 Loss 曲線，其中 mAP 為 78.5%，平均損失率為 1.2182。圖 57 為第五次實驗之混淆矩陣，其中 Accuracy 為 53%，Precision 為 55%，Recall 為 78%，F1 分數為 65%，訓練集 mAP 為 80.5%。從圖 56 可以看出，第五次實驗在 Epochs 為 10,000 之前都處於低處導致後續準確率上升時已經接近訓練結束導致準確率較低。而從圖 57 可以看出，由於訓練時準確率較低導致在測試集時的分類精確度也較不準確，其中 Country 甚至大部分音樂都被分類到 Rock 音樂。

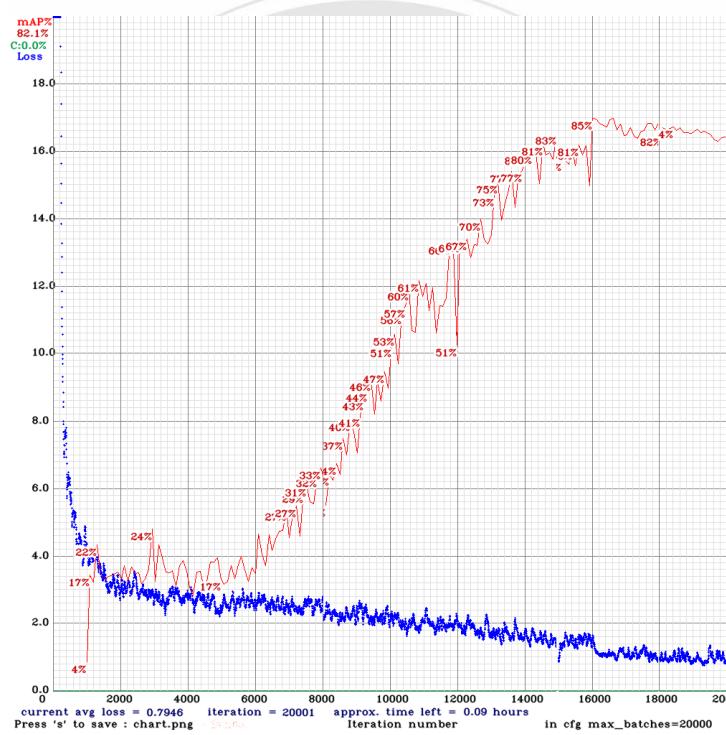


圖58. 可視化 STFT 頻譜圖方法第六次實驗之 mAP 與損失曲線圖

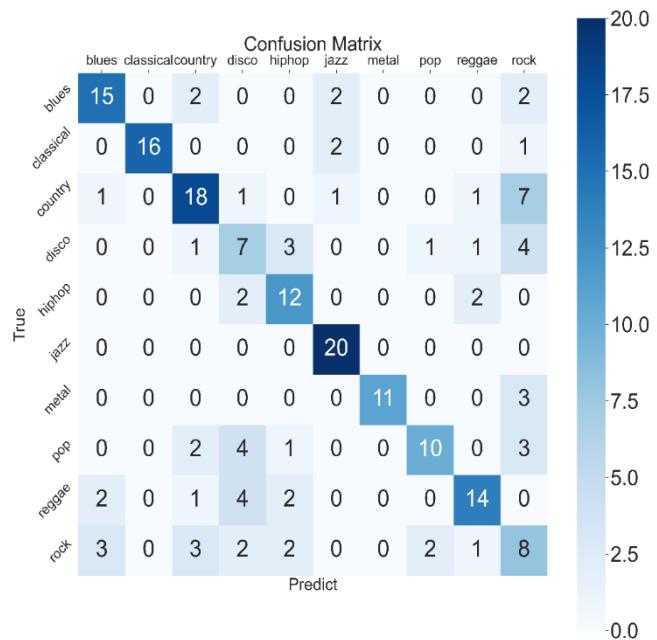


圖59. 可視化 STFT 頻譜圖方法第六次實驗之混淆矩陣圖

圖 58 為第六次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 82.1%，平均損失率為 0.7946。圖 59 為第六次實驗之混淆矩陣，其中 Accuracy 為 65.5%，Precision 為 67%，Recall 為 75%，F1 分數為 71%，測試集 mAP 為 83.59%。從圖 58 可以看出，與實驗五相似處在於 Epochs 為 8,000 之前的準確率都不高，但在 Epochs 為 8,000 之後就持續升高，因此較實驗五準確率來的高。而從圖 59 可以看出，實驗六的分類準確率與前幾次實驗差不多，僅在 Disco 與 Rock 音樂出現較零散的分類。

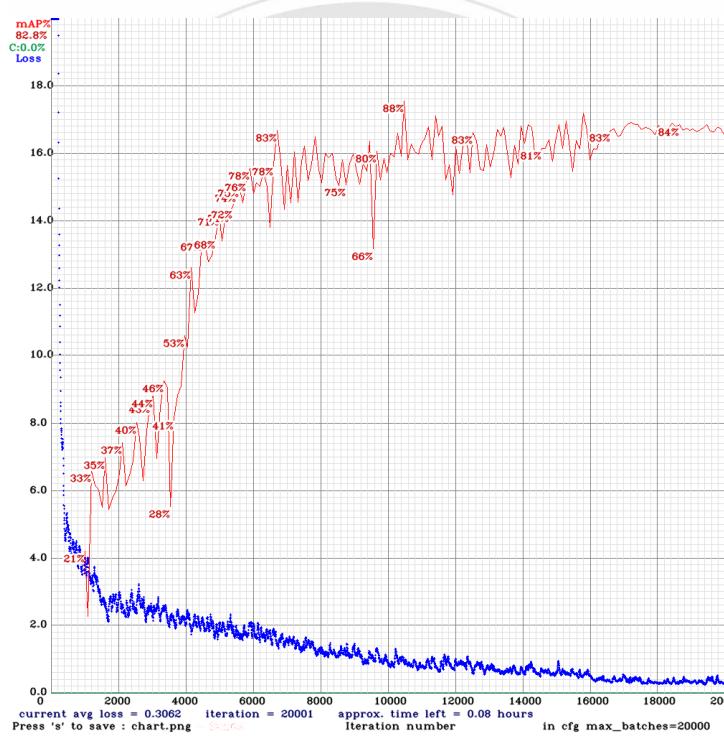


圖60. 可視化 STFT 頻譜圖方法第七次實驗之 mAP 與損失曲線圖

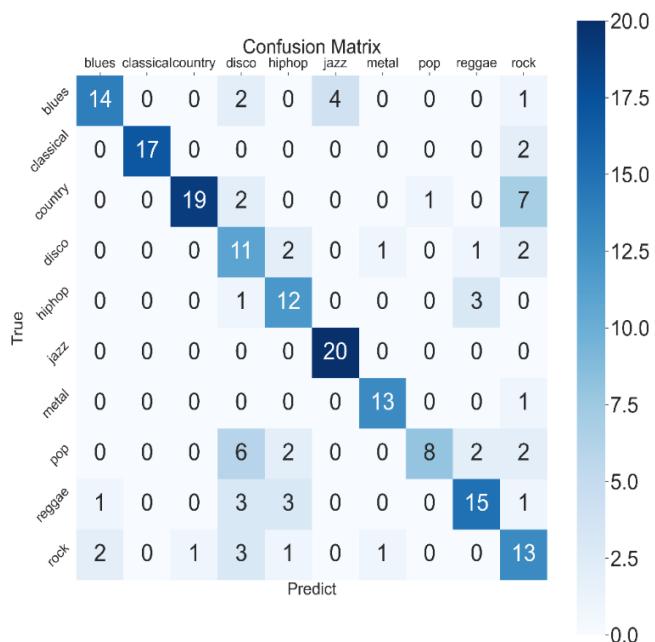


圖61. 可視化 STFT 頻譜圖方法第七次實驗之混淆矩陣圖

圖 60 為第七次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 82.8%，平均損失率為 0.3062。圖 61 為第七次實驗之混淆矩陣，其中 Accuracy 為 71%，Precision 為 73%，Recall 為 82%，F1 分數為 77%，測試集 mAP 為 87.73%。從圖 60 可以看出，第七次實驗的 mAP 呈曲線上升，且損失曲線在 Epochs 為 16,000 時趨於平緩，使得第七次實驗的結果與之前的實驗結果相差無幾。而從圖 61 可以看出，第七次實驗僅在分類 Pop 音樂時出現不穩定之外，其餘的音樂曲風都與前幾次實驗差不多。

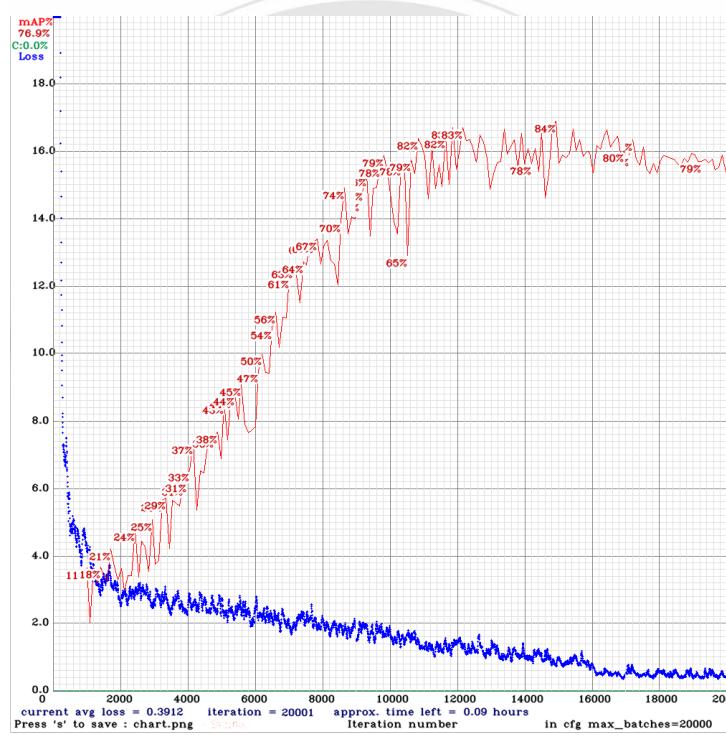


圖62. 可視化 STFT 頻譜圖方法第八次實驗之 mAP 與損失曲線圖

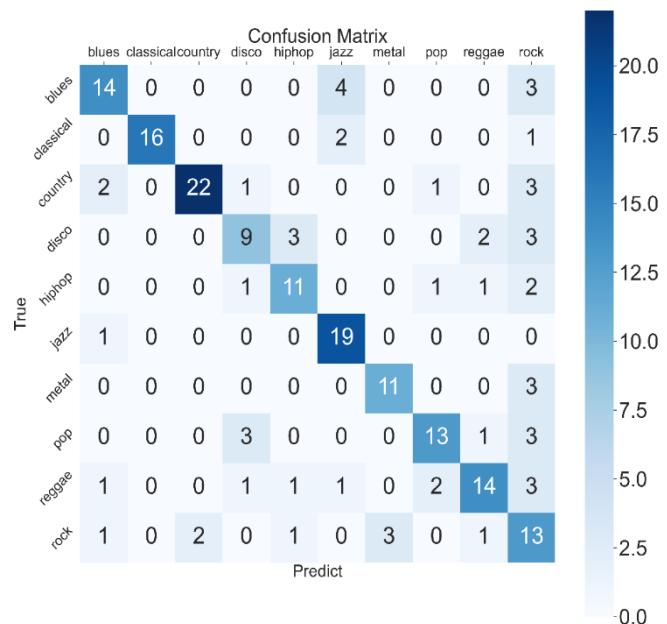


圖63. 可視化 STFT 頻譜圖方法第八次實驗之混淆矩陣圖

圖 62 為第八次實驗的 mAP 與 Loss 曲線，其中 mAP 為 76.9%，平均損失率為 0.3912。圖 63 為第八次實驗之混淆矩陣，其中 Accuracy 為 71%，Precision 為 70%，Recall 為 77%，F1 分數為 73%，測試集 mAP 為 81.68%。從圖 62 可以看出，第八次實驗在訓練時出現較多次起伏大的波動，使得第八次實驗的準確率不高，而損失率下降至 0.3912 並在 Epochs 為 16,000 之後趨於平緩，這是因為 Dropblock 在隨機丟棄區塊時也許丟棄到特徵，使得模型在訓練時需花費較多次 Epochs 才能學習到丟棄的特徵內容，而在第八次實驗中丟棄了多次的特徵使得模型訓練出現多次起伏，造成準確率並無前幾次實驗來的高。而從圖 63 可以看出，模型在分類音樂曲風時雖無出現較嚴重的分類錯誤，但整體的分類準確度都有些微下降，使得第八次實驗的結果並沒有前幾次實驗那麼好。

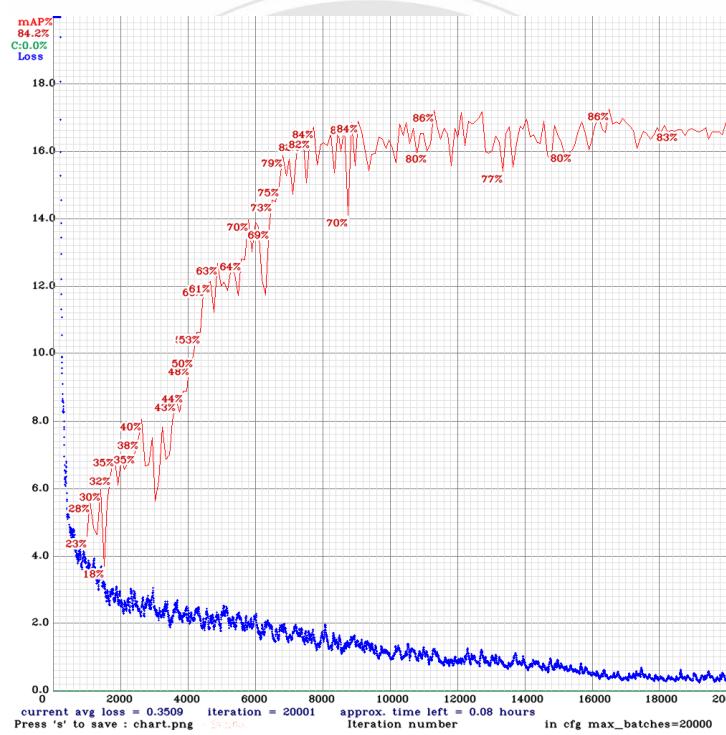


圖64. 可視化 STFT 頻譜圖方法第九次實驗之 mAP 與損失曲線圖

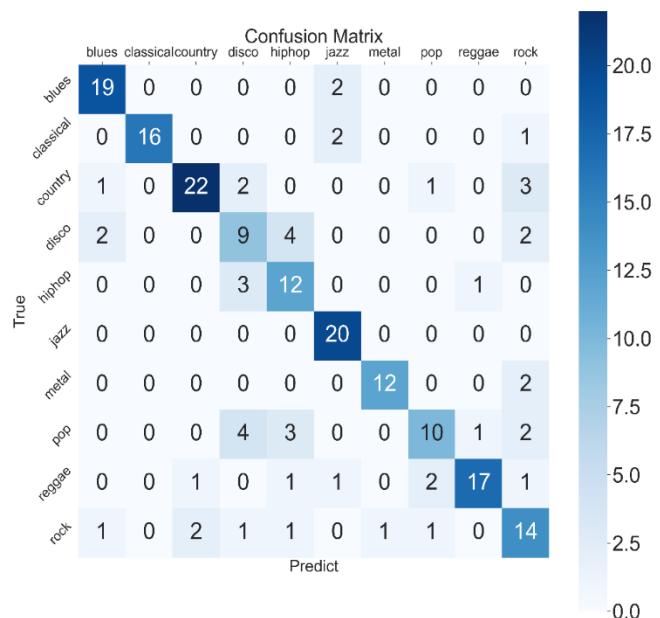


圖65. 可視化 STFT 頻譜圖方法第九次實驗之混淆矩陣圖

圖 64 為第九次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 84.2%，平均損失率為 0.3509。圖 65 為第九次實驗之混淆矩陣，其中 Accuracy 為 75.5%，Precision 為 74%，Recall 為 80%，F1 分數為 77%，測試集 mAP 為 86.21%。從圖 64 可以看出模型在訓練時有出現起伏但次數不多，使得模型整體的準確率是上升的，而從損失曲線可以看出，模型在 Epochs 為 16,000 時趨於平穩，這也與前幾次的實驗相似。而從圖 65 可以看出，模型僅在分類 Disco 與 Pop 音樂時有較多的錯誤，而其餘音樂曲風的分類則與前幾次實驗差不多。

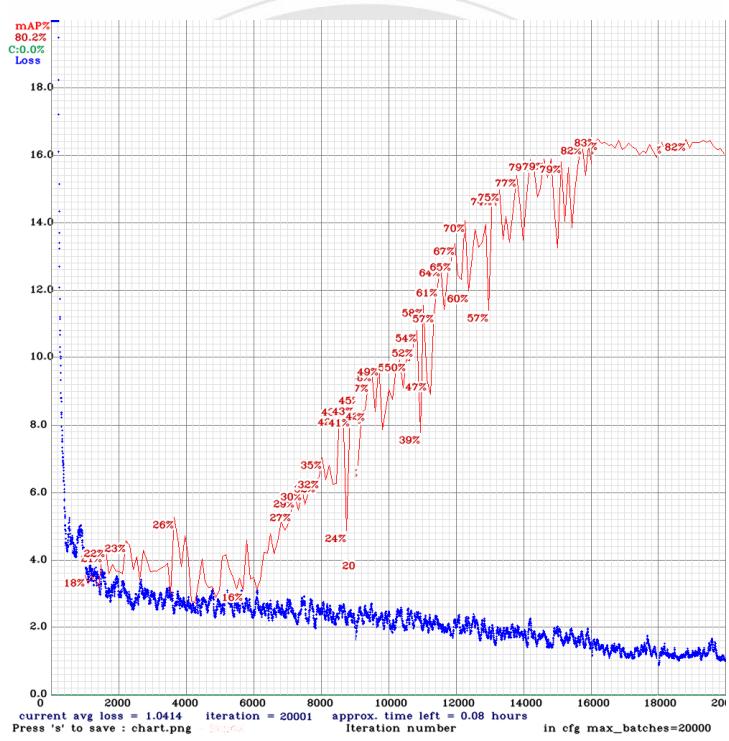


圖66. 可視化 STFT 頻譜圖方法第十次實驗之 mAP 與損失曲線圖

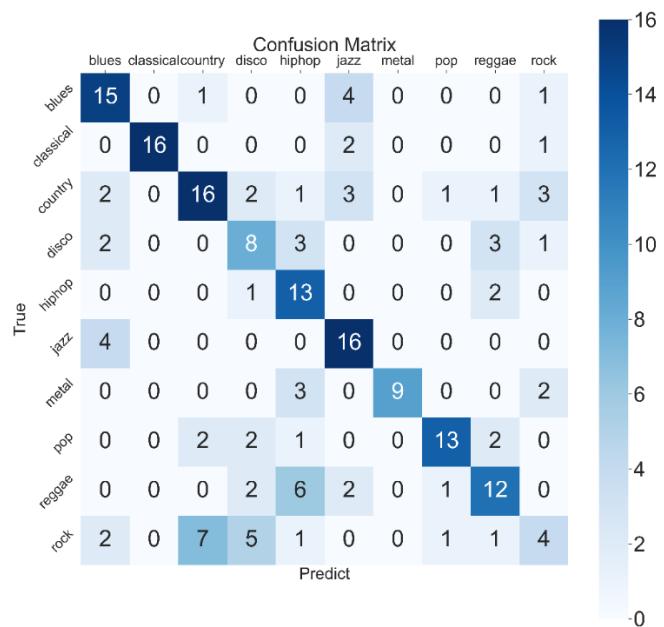


圖67. 可視化 STFT 頻譜圖方法第十次實驗之混淆矩陣圖

圖 66 為第十次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 80.2%，平均損失率為 1.0414。圖 67 為第十次實驗之混淆矩陣，其中 Accuracy 為 61%，Precision 為 62%，Recall 為 79%，F1 分數為 70%，測試集 mAP 為 82.25%。從圖 66 可以看出，第十次實驗的結果與第五次實驗結果類似，在 Epochs 為 8,000 之前都處於低準確率，也因此使得 Loss 落在 1.0414，且從圖 67 可以看出在 Rock 音樂分類中出現了與實驗五相似的錯誤結果，在圖 67 中模型將部分 Rock 音樂分類至 Country 音樂中，由此可以發現可視化 STFT 方法在分類有淵源或相似的音樂曲風時會出現較高的錯誤率。

在可視化 STFT 頻譜方法中得到的 Accuracy 為 68.8%，Precision 為 69.3%，Recall 為 80.2%，F1 分數為 74.3%，mAP 為 84.97%，如表 11 所示。表 12 為 10 次實驗各自的結果。從十次實驗可以看出模型在分類 Jazz 音樂最為準確，而 Hip-hop 音樂在分類時將大部分音樂分類到 Pop 與 Reggae 音樂，這是因為 Hip-hop 音樂起源於 Reggae，因此在 Hip-hop 音樂中常常會有 Reggae 音樂的元素與特色在裡面。相似的情況也發生在 Country 音樂與 Rock 音樂中，由於早期 Rock 音樂是由 Country 音樂衍生而來，因此可以在 Rock 音樂中找到 Country 音樂的元素，這也使得兩者在分類時較不容易。

表11. 可視化 STFT 頻譜圖方法結果統計

評分方式	結果
Accuracy	68.8%
Precision	69.3%
Recall	80.2%
F1-score	74.3%
mAP	84.97%

表12. 可視化 STFT 頻譜圖方法 10 次實驗結果

實驗次數	訓練 mAP	測試 mAP	Precision	Recall	F1-score
1	79.8%	87.38%	72%	81%	76%
2	83.0%	85.56%	75%	82%	78%
3	82.1%	84.3%	69%	81%	75%
4	87.9%	90.55%	76%	87%	81%
5	78.5%	80.50%	55%	78%	65%
6	82.1%	83.59%	67%	75%	71%
7	82.8%	87.73%	73%	82%	77%
8	76.9%	81.68%	70%	77%	73%
9	84.2%	86.21%	74%	80%	77%
10	80.2%	82.25%	62%	79%	70%

五、可視化梅爾頻譜實驗結果

在可視化梅爾頻譜方法中我們使用經過轉換過的圖片進行訓練。首先將 30 秒音訊透過 MATLAB 轉換成可視化頻譜圖，其中窗函數使用漢明窗並設定尺寸為 1,024 帀。重疊區域設定為 512，傅立葉變換尺寸設定為 4,096 帀，如表 13 所示。

表13. 可視化梅爾頻譜方法轉換之參數設定

項目	資訊
歌曲長度	30 秒
窗函數	漢明窗
窗函數尺寸	1,024 帀
重疊區域數	512 帀
傅立葉變換尺寸	4,096 币
梅爾濾波器數量	64 個

將轉換後的可視化梅爾頻譜送入 YOLOv4 中進行訓練，並設定 Epochs 次數為 20,000 次，Batch size 設定為 64，輸入長寬為 416*416，並將資料集拆分為訓練集 70%，驗證集 10%，測試集 20%，且進行 10 次實驗取得平均時以達到較準確的實驗結果，因此評分方式的結果都是 10 次實驗的平均結果。

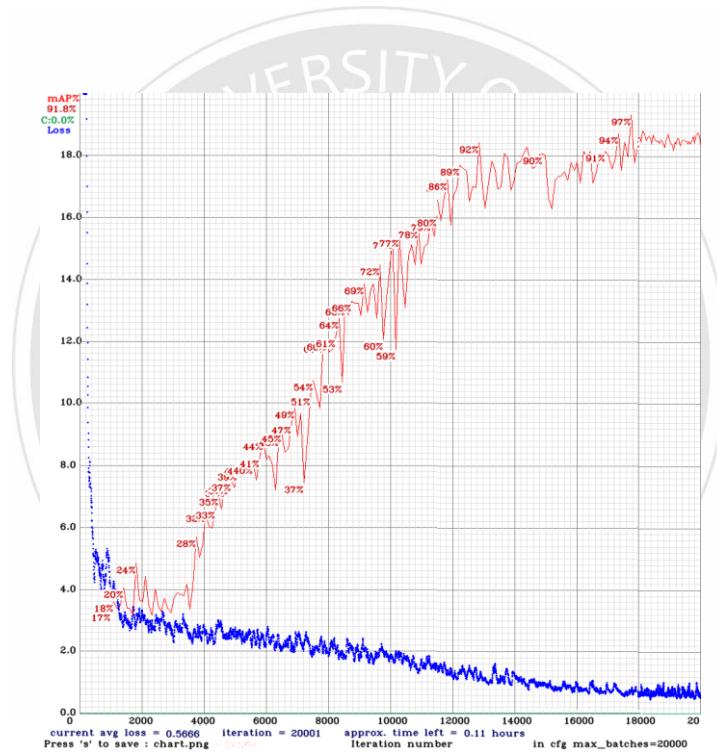


圖68. 可視化梅爾頻譜圖方法第一次實驗之mAP與損失曲線圖

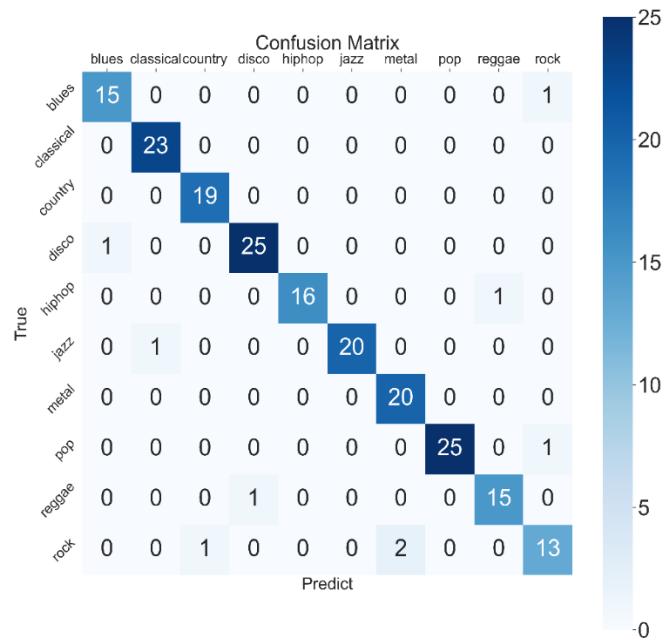


圖69. 可視化梅爾頻譜圖方法第一次實驗之混淆矩陣圖

圖 68 為第一次實驗的 mAP 與 Loss 曲線，其中 mAP 為 91.8%，平均損失率為 0.5666。圖 69 為第一次實驗的混淆矩陣，其中 Accuracy 為 95.5%，Precision 為 96%，Recall 為 96%，F1 分數為 96%，測試集 mAP 為 94.57%。從圖 50 可以看出，模型在訓練時 mAP 從 17% 上升至最高 97%，平均的訓練 mAP 為 91.8% 且呈現曲線上升，其中幾次出現波動表示 Dropblock 發揮作用，使模型在訓練時隨機丟棄一區塊的特徵防止出現過擬合，而這種上下起伏的波動也表示模型是穩定學習而非一次性的學習完，可使泛化性提升。而從損失曲線可以看出，平均損失率是落在 0.5666，從整體曲線趨勢來看可以發現，損失曲線呈現有斜率下降並從 Epochs 為 14,000 次時趨於平穩但卻落在 0.5666，這顯示模型理應還有訓練空間，為了確定此狀況並非偶然發生，因此我們進行了 10 次實驗以驗證此結果。從圖 69 的混淆矩陣可以看出，可視化梅爾頻譜在測試集上的結果中是良好的，各個曲風分類錯誤相比其他方法來的更少，僅出現零星次數。在測試集中總共有 200 首歌曲應用於測試集，這些歌曲是不參與模型訓練的，因此可以作為模型評估用途。在圖 69 的混淆矩陣中，10 種曲風所佔的歌曲比例是不相同的，X 軸是模型預測該首歌的曲風，Y 軸是該首歌實際的曲風，交叉比對後得出中間顏色較深的即為預測與實際結果相同的歌曲總數，從混淆矩陣可以看出模型在測試集中分類的正確性。

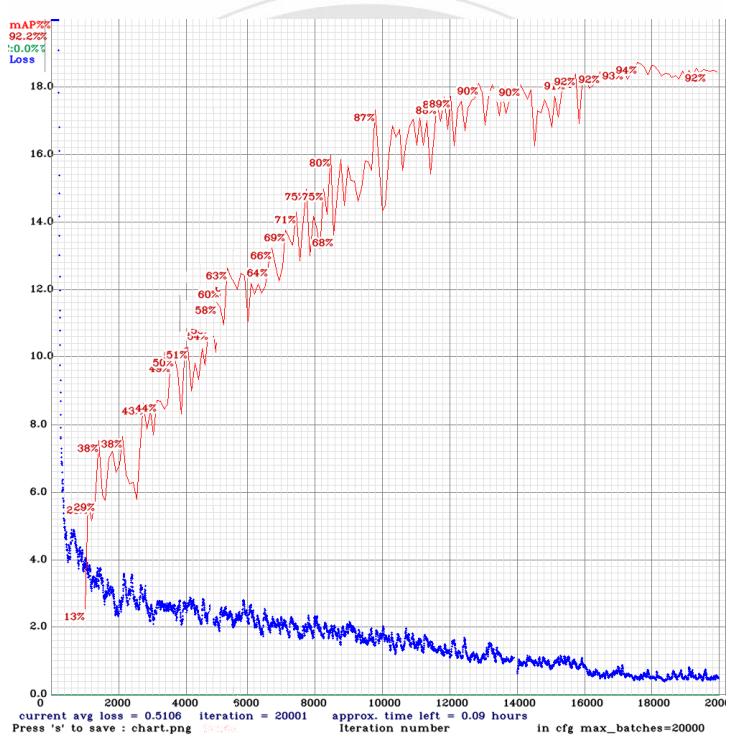


圖70. 可視化梅爾頻譜圖方法第二次實驗之mAP與損失曲線圖

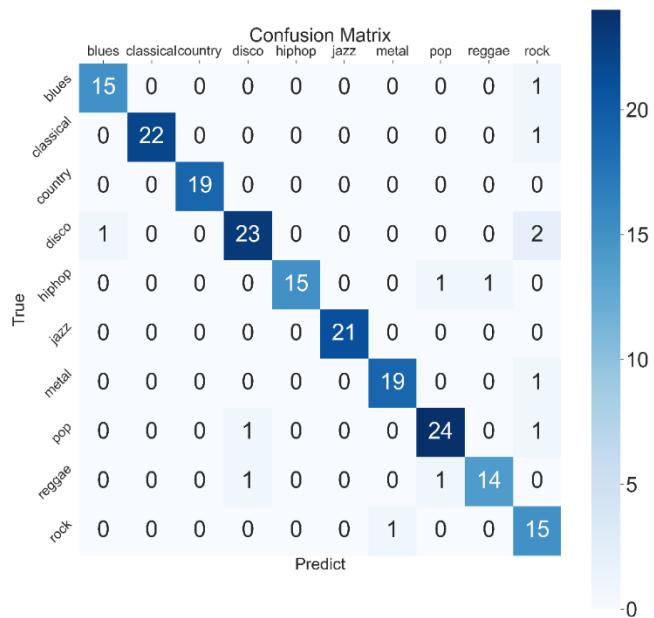


圖71. 可視化梅爾頻譜圖方法第二次實驗之混淆矩陣圖

圖 70 為第二次實驗的 mAP 與 loss 曲線，其中 mAP 為 92.2%，平均損失率為 0.5106。圖 71 為第二次實驗的混淆矩陣，其中 Accuracy 為 93.5%，Precision 為 95%，Recall 為 97%，F1 分數為 96%，測試集 mAP 為 96.48%。從圖 52 可以看出，與第一次實驗相比，第二次實驗中 mAP 上升的速度更快並持續上升，跌落部分為 Dropblock 造成的，因此與第一次實驗相比，準確率可以上升至 92.2%，並在 Epochs 為 16,000 後損失曲線才趨於平穩，相較於第一次實驗的結果更優秀。

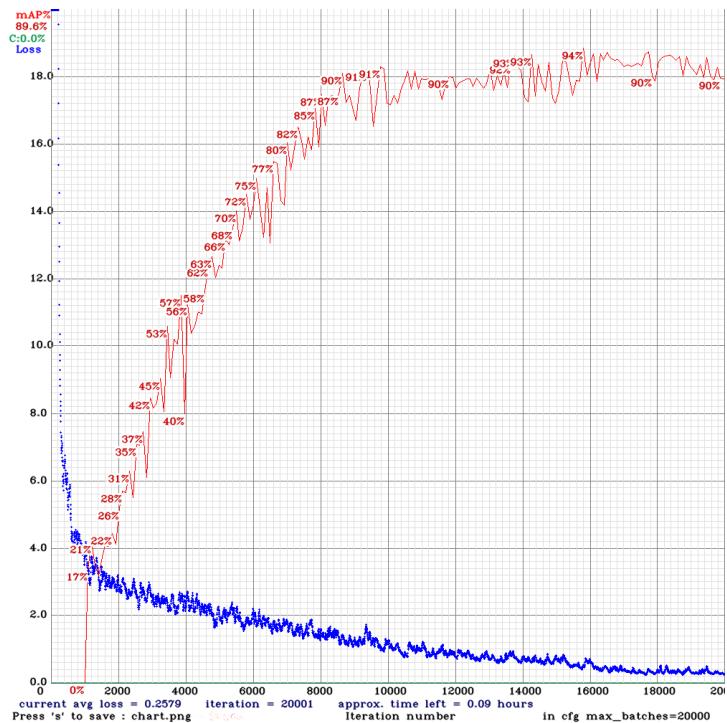


圖72. 可視化梅爾頻譜圖方法第三次實驗之 mAP 與損失曲線圖

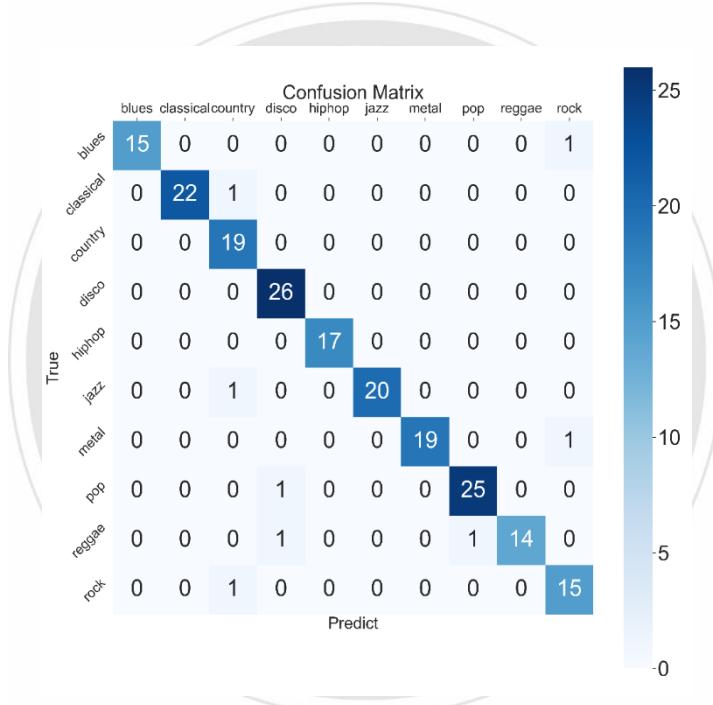


圖 73. 可視化梅爾頻譜圖方法第三次實驗之混淆矩陣圖

圖 72 為第三次實驗的 mAP 與 Loss 曲線，其中 mAP 為 89.6%，平均損失率為 0.2579。圖 73 為第三次實驗之混淆矩陣，其中 Accuracy 為 96%，Precision 為 95%，Recall 為 96%，F1 分數為 96%，測試集 mAP 為 97.84%。從圖 72 可以看出，與第二次實驗相比，第三次實驗 mAP 曲線在 Epochs 為 8,000 前急劇上升且後期波動較大，因此準確率相較於第二次實驗來的低，這意味著如果太快學習完所有東西且 Epochs 次數沒有增加，那麼訓練的準確率就會較低。從圖 73 可以看出，即使在訓練時的 mAP 相較於第二次實驗來的低，但在測試時則是優於第二次實驗，從混淆矩陣可以看出，第二次實驗分類錯誤率高的 Disco 音樂在第三次實驗中得到較好的結果，而其他曲風都保持差不多的準確率。

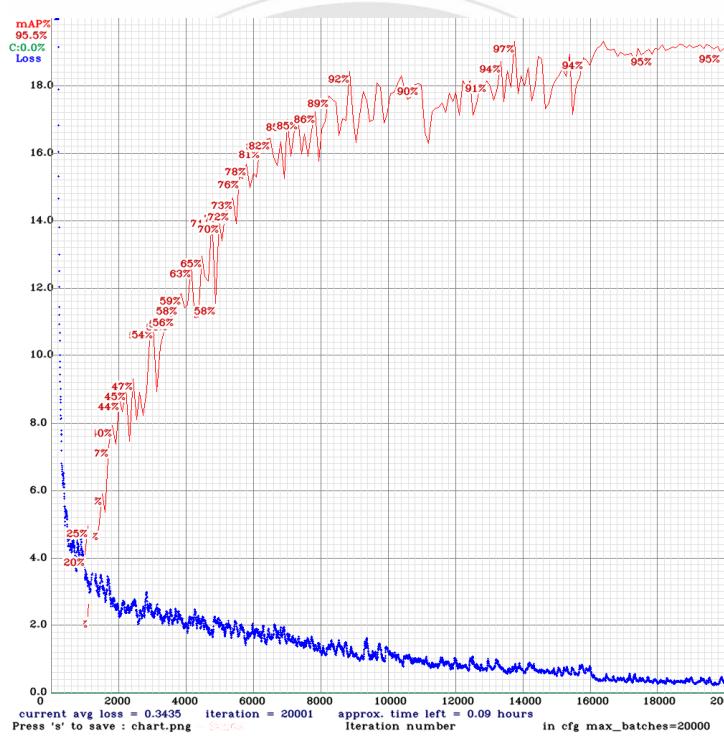


圖74. 可視化梅爾頻譜圖方法第四次實驗之mAP與損失曲線圖

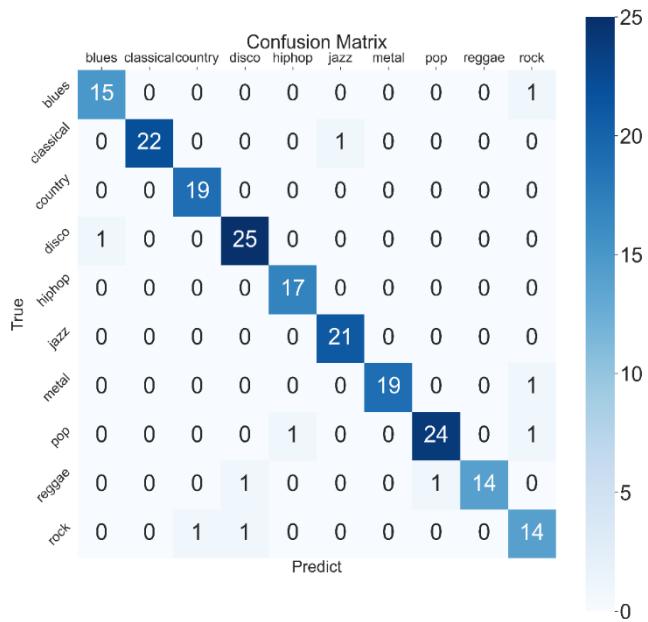


圖75. 可視化梅爾頻譜圖方法第四次實驗之混淆矩陣圖

圖 74 為第四次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 95.5%，平均損失率為 0.3435。圖 75 為第四次實驗之混淆矩陣，其中 Accuracy 為 95%，Precision 為 94%，Recall 為 99%，F1 分數為 97%，測試集 mAP 為 98.82%。第四次實驗是可視化梅爾頻譜方法中得到最高準確率的實驗，從圖 74 可以看出，mAP 曲線的幅度相較於前幾次實驗來的大，並在 Epochs 為 16,000 之後還繼續上升，使得第四次實驗的訓練 mAP 高達 95.5%。而從圖 75 可以看出，第四次實驗在測試集分類時相較於前幾次實驗都更加準確一些但不多，從結果來看似乎已經到達可視化梅爾頻譜圖的極限。

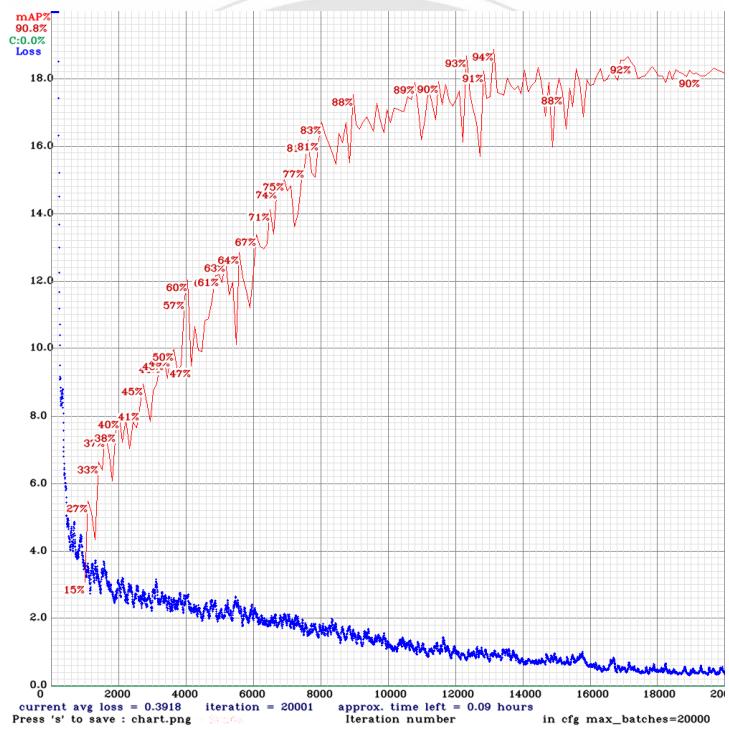


圖76. 可視化梅爾頻譜圖方法第五次實驗之mAP與損失曲線圖

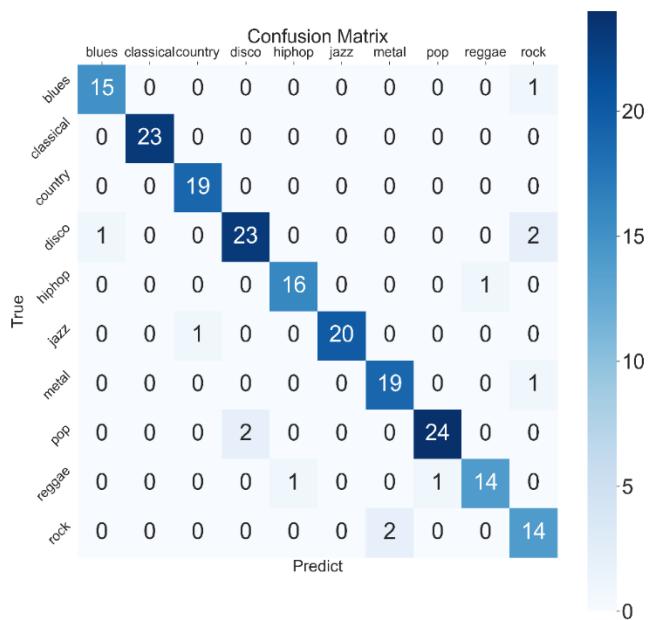


圖77. 可視化梅爾頻譜圖方法第五次實驗之混淆矩陣圖

圖 76 為第五次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 90.8%，平均損失率為 0.3918。圖 77 為第五次實驗之混淆矩陣，其中 Accuracy 為 93.5%，Precision 為 94%，Recall 為 96%，F1 分數為 95%，測試集 mAP 為 98.81%。從圖 76 可以看出，第五次實驗的 mAP 曲線圖相較前幾次實驗的波動幅度與次數來的大且多，尤其在 Epochs 為 4,000 至 6,000 之間，使得第五次實驗的 mAP 僅 90.8%。而從圖 77 可以看出，由於訓練時的多次波動造成模型在分類上有較多錯誤出現，其中在 Disco 與 Pop 音樂分類中出現較多錯誤。

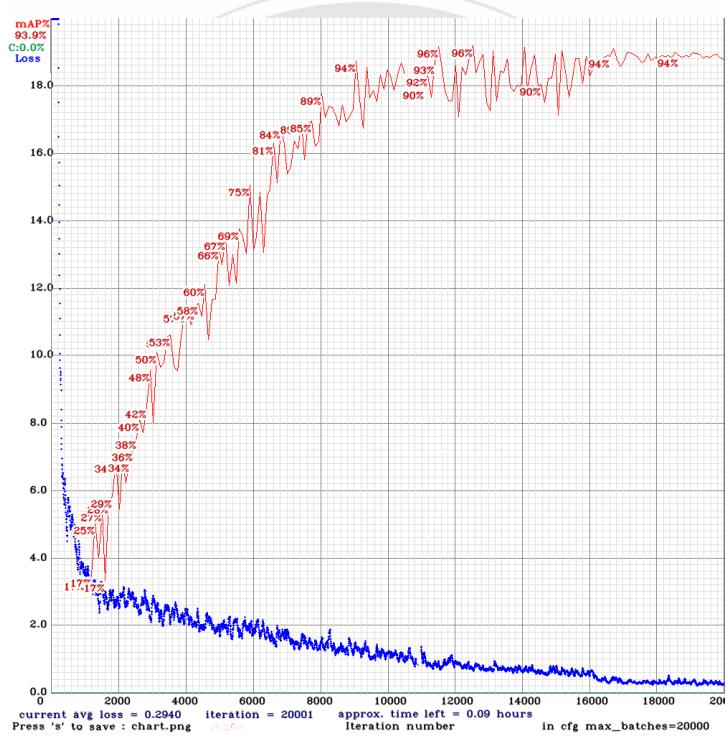


圖78. 可視化梅爾頻譜圖方法第六次實驗之mAP與損失曲線圖

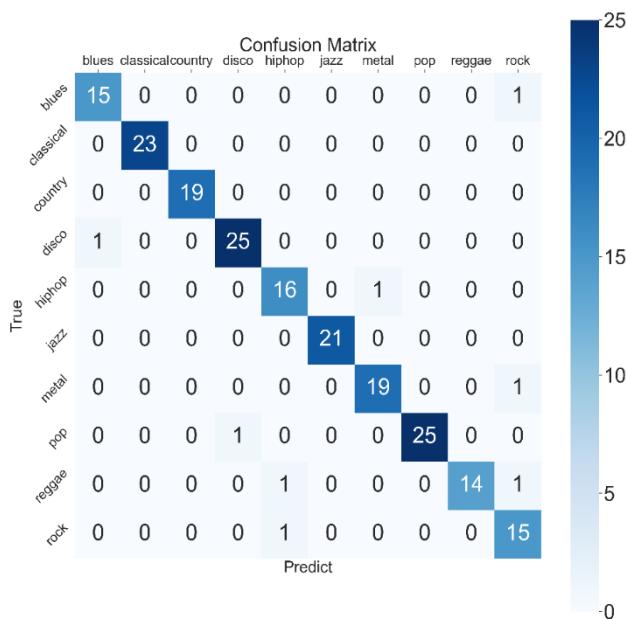


圖79. 可視化梅爾頻譜圖方法第六次實驗之混淆矩陣圖

圖 78 為第六次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 93.9%，平均損失率為 0.2940。圖 79 為第六次實驗之混淆矩陣，其中 Accuracy 為 96%，Precision 為 93%，Recall 為 96%，F1 分數為 95%，測試集 mAP 為 98.53%。從圖 78 可以看出，第五次實驗在 Epochs 為 12,000 至 15,000 之間出現較大波動，但由於當時 mAP 已經到達高點，因此並無太大影響準確率。而從圖 79 可以看出，第六次實驗在分類時結果與前幾次實驗相差無幾，僅有零星分類錯誤。

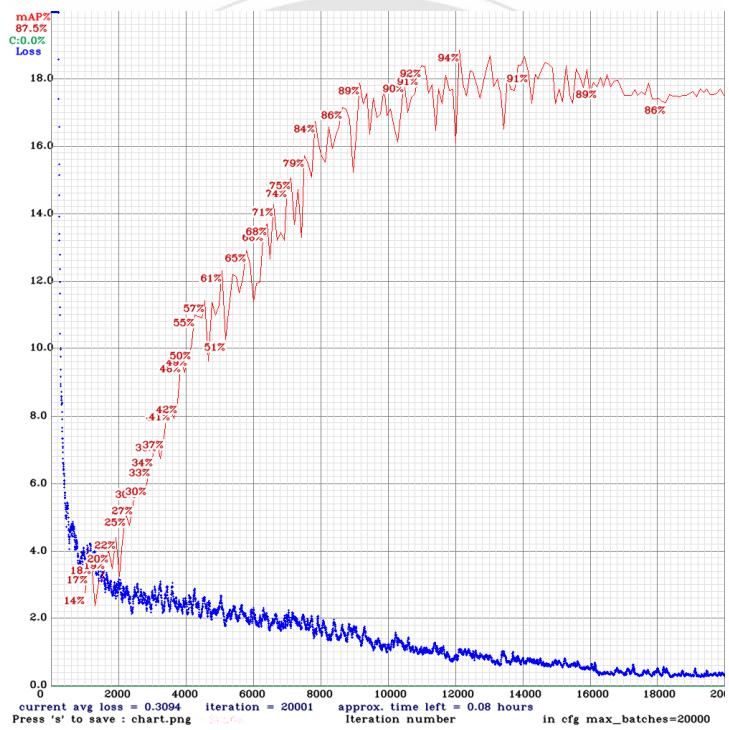


圖80. 可視化梅爾頻譜圖方法第七次實驗之mAP與損失曲線圖

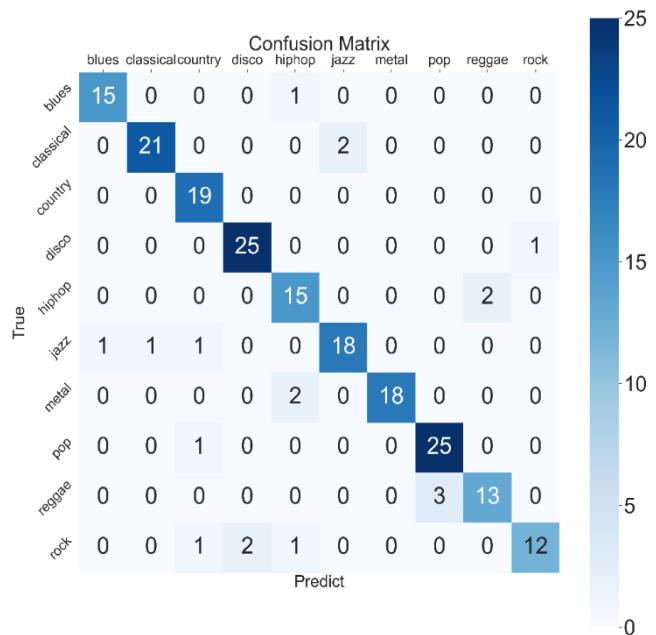


圖81. 可視化梅爾頻譜圖方法第七次實驗之混淆矩陣圖

圖 80 為第七次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 87.5%，平均損失率為 0.3094。圖 81 為第七次實驗之混淆矩陣，其中 Accuracy 為 90.5%，Precision 為 87%，Recall 為 96%，F1 分數為 91%，測試集 mAP 為 99.26%。從圖 80 可以看出，在 Epochs 為 12,000 時出現如實驗六相同的巨幅波動但不如實驗六有回升，因此在訓練時的 mAP 僅獲得 87.5% 的準確率。而從圖 81 可以看出，由於訓練準確率低下的關係造成模型在測試集中分類錯誤率增加，其中 Jazz、Reggae、Rock 音樂都出現大量錯誤，而從這些音樂分類錯誤的曲風來看並非與音樂起源有關，因此為模型在訓練時出現的波動導致分類準確率下降。

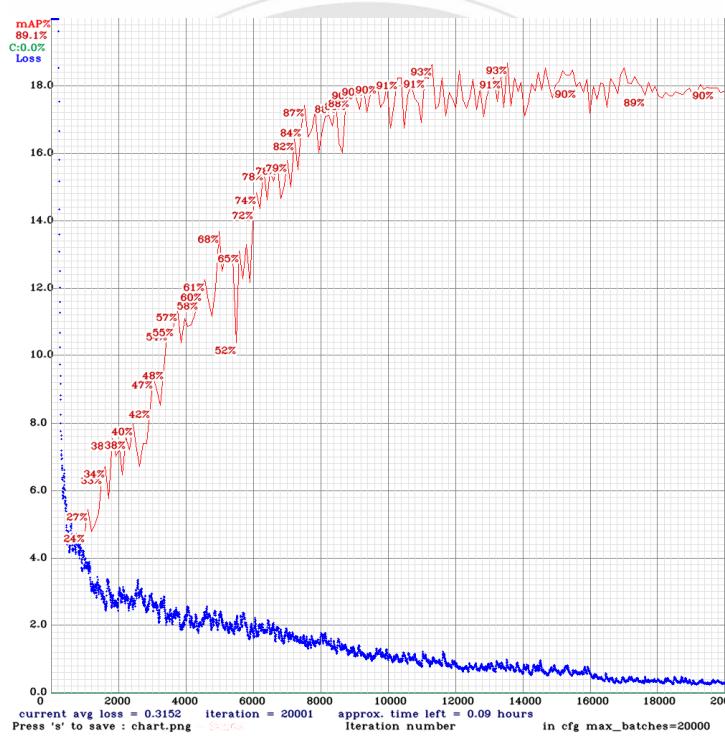


圖82. 可視化梅爾頻譜圖方法第八次實驗之mAP與損失曲線圖

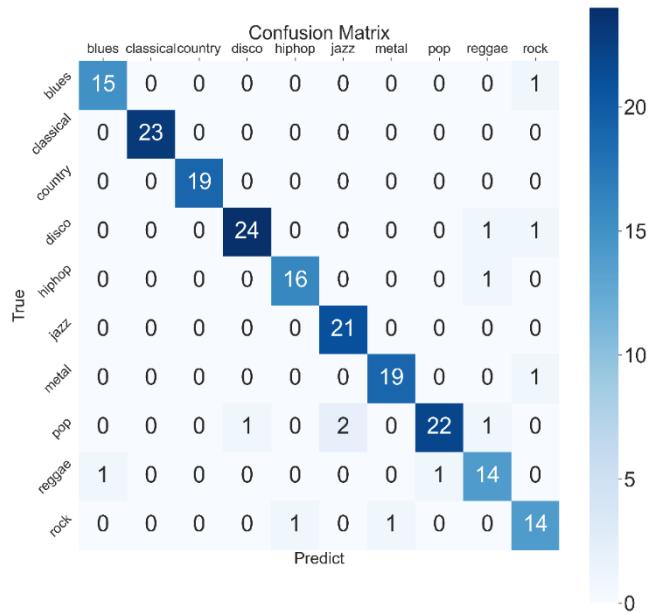


圖83. 可視化梅爾頻譜圖方法第八次實驗之混淆矩陣圖

圖 82 為第八次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 89.1%，平均損失率為 0.3152。圖 83 為第八次實驗之混淆矩陣，其中 Accuracy 為 93.5%，Precision 為 91%，Recall 為 94%，F1 分數為 93%，測試集 mAP 為 97.91%。從圖 82 可以看出，除了在 Epochs 為 4,000 至 6,000 之間出現巨大波動外，其餘時候都呈現較穩定狀態，由於準確率上升幅度較平緩因此在訓練時的準確率僅 89.1%。而從圖 83 可以看出，第八次實驗在 Pop 音樂出現較多錯誤之外其餘分類沒有因為訓練時的穩定狀態出現的較低準確率而受到影響，而是與前幾次實驗出現相似的分類結果。

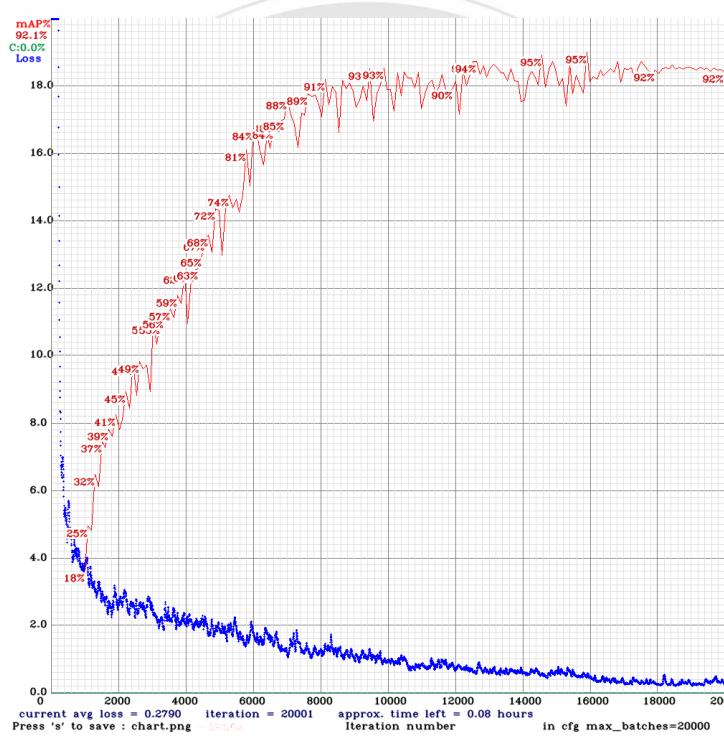


圖84. 可視化梅爾頻譜圖方法第九次實驗之mAP與損失曲線圖

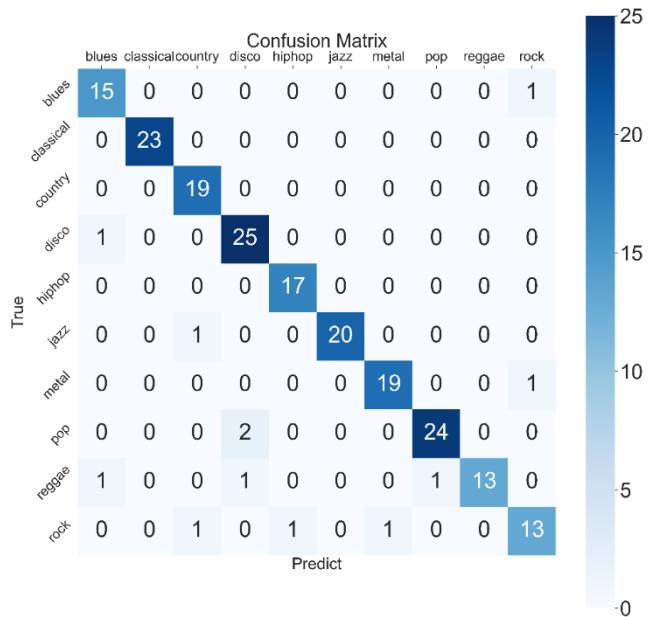


圖85. 可視化梅爾頻譜圖方法第九次實驗之混淆矩陣圖

圖 84 為第九次實驗的 mAP 與 Loss 曲線圖，其中 mAP 為 92.1%，平均損失率為 0.2790。圖 85 為第九次實驗之混淆矩陣，其中 Accuracy 為 94%，Precision 為 96%，Recall 為 98%，F1 分數為 97%，測試集 mAP 為 98.87%。從圖 84 可以看出，第九次實驗的曲線圖與第四次實驗相似，但相較於第四次實驗的結果，第九次實驗的曲線上升較穩定，因此準確率較第四次實驗來的低。而從圖 85 可以看出，第九次實驗在曲風分類上也與第四次實驗較接近，但在 Reggae 與 Rock 音樂分類上有較多分類錯誤發生。

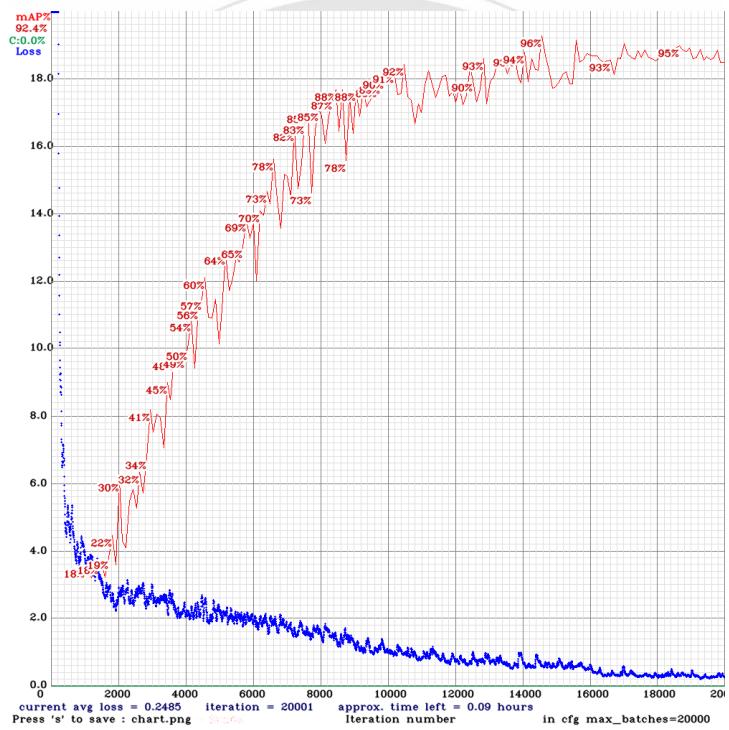


圖86. 可視化梅爾頻譜圖方法第十次實驗之mAP與損失曲線圖

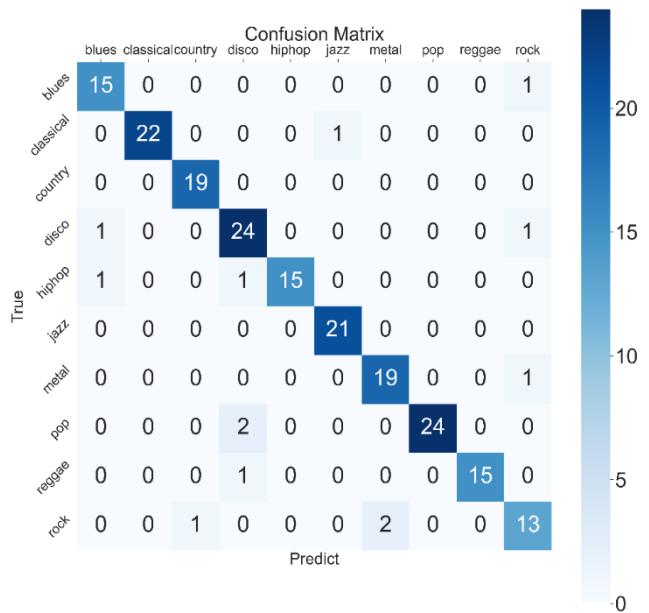


圖87. 可視化梅爾頻譜圖方法第十次實驗之混淆矩陣圖

圖 86 為第十次實驗的 mAP 與 Loss 曲線，其中 mAP 為 92.4%，平均損失率為 0.2485。圖 87 為第十次實驗之混淆矩陣，其中 Accuracy 為 93.5%，Precision 為 95%，Recall 為 95%，F1 分數為 95%，測試集 mAP 為 98.19%。從圖 86 可以看出，第十次實驗在訓練時的 mAP 在 Epochs 為 6,000 至 10,000 之間出現較大幅度波動，但並無出現巨大跌幅，因此在波動過後的 mAP 曲線持續呈現穩定上升。而從圖 87 可以看出，第十次實驗在分類時也與前幾次實驗結果相似，僅在 Rock 音樂中有出現較多錯誤分類結果。

在可視化梅爾頻譜方法中得到的 Accuracy 為 94.5%，Precision 為 93.6%，Recall 為 96.3%，F1 分數為 95.1%，mAP 為 97.93%，如表 14 所示。表 15 為 10 次實驗各自的結果。從 10 次實驗中分類錯誤的曲風中可以看到，分類錯誤次數較多的曲風為 Pop、Reggae、Rock 等曲風，其中 Pop 音樂在分類時錯誤的分給了 Hip-hop 與 Rock 音樂，這是因為 Pop 音樂在不同時代所代表的曲風是不相同的，但通通稱為 Pop 音樂，也就是流行音樂。Reggae 音樂在分類時錯誤的分給了 Disco 與 Pop 音樂，這是因為 Reggae 音樂有著巨大的影響力，其特色與元素被後來的曲風所接納並融合，因此可以在許多音樂曲風中看到 Reggae 音樂的靈魂。Rock 音樂在分類時錯誤的分給了 Country 與 Disco 音樂，早期的 Rock 音樂是由 Country 音樂所衍生出的曲風因此極為相似，不同之處在於使用的樂器與歌詞內容，也因此容易辨識錯誤。

表14. 可視化梅爾頻譜圖方法結果統計

評分方式	結果
Accuracy	94.5%
Precision	93.6%
Recall	96.3%
F1-score	95.1%
mAP	97.93%

表15. 可視化梅爾頻譜圖方法 10 次實驗結果

實驗次數	訓練 mAP	測試 mAP	Precision	Recall	F1-score
1	91.8%	94.57%	96%	96%	96%
2	92.2%	96.48%	95%	97%	96%
3	89.6%	97.84%	95%	96%	96%
4	95.5%	98.82%	94%	99%	97%
5	90.8%	98.81%	94%	96%	95%
6	93.9%	98.53%	93%	96%	95%
7	87.5%	99.26%	87%	96%	91%
8	89.1%	97.91%	91%	94%	93%
9	92.1%	98.87%	96%	98%	97%
10	92.4%	98.19%	95%	95%	95%

六、所有方法之結果與優缺點

我們將 CNN 方法、CRNN 方法、LSTM 方法、可視化 STFT 方法、可視化梅爾頻譜方法的結果統計於表 16 中，由於可視化 STFT 方法與可視化梅爾頻譜方法是使用圖像辨識與分類的方法，因此多出 mAP 欄位。從表 16 可以看出，可視化梅爾頻譜方法所獲得的準確率是最高的，而 LSTM 則是最低。CNN 方法的優勢在於模型建置簡單且快速，並且可以直接增加層數而不用更改整體架構，在本研究中我們僅使用 5 層卷積層就可以獲得 73.1 的準確率，且 CNN 方法也方便與其他神經網路方法進行擴充，如[62]的 SampleCNN、[52]的 PRCNN 等方法，然而其優點也是缺點，因為僅使用 CNN 方法的準確率在本研究中已經到達瓶頸無法突破，因此許多人使用 CNN 方法在不更改模型架構下會在分類處使用支持向量機(Support Vector Machine, SVM)、決策樹(Decision Tree)、K-近鄰演算法(K Nearest Neighbor, KNN)等方法在分類時進行加強，如[4]所使用的各式分類器方法。CRNN 方法是基於 CNN 的擴充方法，我們參考了[57]所提出的 CRNN，在最後一層卷積層之後添加 RNN 以補足 CNN 方法的不足，而從結果來看 CRNN 是接近於 CNN，在之前的實驗[65]中我們的 CRNN 出現問題導致當時在訓練時 Epochs 為 12 次時被中斷，因此僅獲得 43%的準確率，而在本研究中我們找出上次實驗的問題並修正後則獲得了 68%的準確率但出現過擬合導致結果低於 CNN，因此在後續實驗中將根據此問題進行修改。LSTM 方法在實驗中得

到的結果是最低的，從訓練曲線圖可以看出模型出現兩次的上漲曲線導致準確率較低，由於僅使用 LSTM 作為模型建置方法而不添加其他神經網路則會使 LSTM 的並行處理存在劣勢、對於較長序列資料存在缺陷、訓練佔用資源多且訓練時間較長等三個缺陷被放大，由於音樂屬於長時間序列，因此我們認為應使 LSTM 與其他神經網路進行結合，才可發揮 LSTM 對於時間序列處理的優勢。可視化 STFT 方法與可視化梅爾頻譜方法都是使用 MATLAB 進行可視化的轉換，因此雖然可視化方法可以獲得較高的準確率且可視化梅爾頻譜獲得的準確率最高，但兩者都多出轉換的步驟，因此相較 CNN、CRNN、LSTM 等方法來的不便利，而可視化方法都是使用 YOLOv4 進行訓練與測試，因此在擴充模型架構與應用時需基於 YOLOv4 架構進行，需耗費較多時間與精力，但由可視化 STFT 與可視化梅爾頻譜方法的結果可以得出，使用可視化方法相較於傳統頻譜方法可以獲得更高的準確率。我們同時與其他人所提出的方法進行比較，如表 17 所示。由於無法得知其他方法的 Precision、Recall、F1 分數等評分指標，因此在表 17 中僅比較 Accuracy，其中[34]是使用 CNN 搭配可視化頻譜圖、[43]是使用 CNN 搭配 STFT 方法、[51]是使用自己所提出的 Modified Group Delay Features (MODGDF)方法，使用 MFCC 作為輸入並使用 SVM 作為分類器、[61]是使用 SVM 方法、[11]是使用 CNN 搭配 SVM 進行實驗、[12]是使用 STFT 搭配 SVM 分類器方法、

[19]與[20]是使用非負張量分解（non-negative tensor factorization, NTF）方法進行實驗。

表16. 各式方法實驗結果統計表

實驗方法	Accuracy	Precision	Recall	F1-score	mAP
CNN	73.1%	65.1%	61.7%	61.3%	73.08%
CRNN	68%	69.6%	67.9%	67.3%	62.9%
LSTM	42.5%	44.78%	40%	37.8%	49.78%
可視化 STFT	68.8%	69.3%	80.2%	74.3%	84.97%
可視化梅爾頻譜	94.5%	93.6%	96.3%	95.1%	97.93%

表17. 本研究結果與其他方法之比較

實驗方法	Accuracy
本研究之 CNN	73.1%
本研究之 CRNN	68%
本研究之 LSTM	42.5%
本研究之可視化 STFT	68.8%
本研究之可視化梅爾頻譜	94.5%
參考文獻[11]	72.6%
參考文獻[12]	71.3%
參考文獻[19]	75%
參考文獻[20]	78.9%
參考文獻[34]	84.7%
參考文獻[43]	67%
參考文獻[51]	75.5%
參考文獻[61]	81.5%

伍、結論

由於科技的進步使得人們的生活水平日漸提高，隨之而來的是生活壓力也漸漸升高，在這樣的高壓力下若無法適當的釋放壓力會帶來嚴重的後果，進而增加社會的自殺率與犯罪率，因此如何快速的釋放壓力是在這個高壓社會中生活的我們需要面對的一個重要議題。由於音樂是最方便取得且效率最高的紓壓方式，且電腦與網路的飛速發展使音樂製作的門檻降低，許多的音樂家與製作人將他們製作的音樂上傳至網際網路與串流媒體上，而龐大的線上音樂資料庫使得人們在搜尋音樂時難以進行搜尋與分類，如同圖書館會依照書本種類進行分類，音樂資料庫也需要如同圖書館一樣的分類方法，因此在本研究中我們提出依照音樂曲風的分類基準，使用 CNN、CRNN、LSTM、可視化 STFT 頻譜圖、可視化梅爾頻譜圖等五種方法進行比較以找出準確率最高的方法，並使用 Accuracy、Precision、Recall、F1 分數、mAP 等五種評分指標進行評分，結果表明，使用可視化梅爾頻譜圖在 5 種方法中都獲得最好的結果，其 Accuracy 為 94.5%、Precision 為 93.6%、Recall 為 96.3%、F1 分數為 95.1%、mAP 為 97.93%，並且我們同時與其他研究者的方法進行比較，但由於其他研究者僅使用 Accuracy 做為評分指標，因此我們只比較了 Accuracy 的結果，在與其他研究者的方法比較中也是可視化梅爾頻譜圖有著較高的準確率。

陸、未來研究方向

在本研究中，我們使用的 CRNN 與 LSTM 方法在驗證與測試時都發生過擬合的情況，雖然兩者都有添加 Dropout 以防止過擬合但還是有出現該狀況，因此在未來研究中我們會持續分析與優化兩者存在的問題。而對於可視化 STFT 與可視化梅爾頻譜兩個方法，得益於 YOLOv4 的強大效能使得我們提出的可視化方法有著優異的結果並證明了可視化頻譜圖的可行性，因此在未來的研究我們將致力於研究出適合用於可視化頻譜圖的神經網路架構，並與 YOLOv4 方法進行比較與分析，以期望找出最適合用在可視化頻譜圖的神經網路架構。而對於 GTZAN 資料集，由於 GTZAN 資料集屬於平衡資料集，且僅有 1,000 首歌曲，因此在未來研究的另一個方向是使用其他研究常用的資料集進行研究與分析，包含 MagnaTagATune(MTT)、Free Music Archive (FMA)、Million Song Dataset (MSD)、Extended Ballroom Dataset 與 Beatport EDM Key Dataset 等資料集，這些資料集的相關資料如表 18 所示。

表18. 各音樂資料集相關資訊

類別＼名稱	GTZAN	MagnaTag ATune (MTT)	FreeMusicArchive (FMA)	MillionSongDataset (MSD)	BeatportED MKeyDataset	ExtendedBallroomdataset
曲風數	10	未提供，但提供標籤數共 188 個	16	未提供，但提供標籤數共 7,643 個	12	13
歌曲數	1000	25863	106,574	1,000,000	1,486	4180
總容量	1GB	3GB	917GB	280GB	2.1GB	3.2GB
歌曲長度	30s	29s	完整歌曲	完整歌曲	120s	30s
音頻數據	.au, 22,050Hz, 16-bits, Mono	.mp3, 16,000Hz, 32kbps, Mono	未知	不提供歌曲	.mp3, 96kbps	.mp3, 192kbps

參考文獻

- [1] 牛璽翔,以圖像化音樂情緒分類系統應用於音樂風格分析及曲目選取,碩士論文,國立陽明交通大學,2015。
- [2] 吳振銘,應用改良式 K-means 分群法於個人化音樂推薦服務系統之實現,碩士論文,國立高雄應用科技大學,2012。
- [3] 吳紹敬,結合時間與非時間特徵進行音樂曲風分類之研究,碩士論文,國立陽明交通大學,2010。
- [4] 易洋,使用 CNN 為基礎之特徵擷取器及複合分類器結構之音樂曲風分類,碩士論文,國立臺灣科技大學,2019。
- [5] 姚路遙遙(2021). "Swish 激活函數詳解." CSDN. <https://reurl.cc/MN1NdM> .(May 22, 2022).
- [6] 張育瑞,基於深度學習之 AAC 壓縮域翻唱歌快速檢索,碩士論文,國立中央大學,2015。
- [7] 演算法筆記. "Audio." <https://reurl.cc/XVDVea> .(May 22, 2022).
- [8] 蔡振家(2015),「音樂與情緒反應[電子版]」,科學發展期刊,第 506 期,第 18-23 頁。
- [9] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [10] A. C. Guyton and J. H. Hall, *Textbook of medical physiology*. Philadelphia: Elsevier Saunders, 1996.
- [11] A. Elbir, H. B. Çam, M. E. Iyican, B. Öztürk, and N. Aydin, "Music genre classification and recommendation by using machine learning techniques," *In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1-5, 2018.
- [12] A. Elbir, H. O. İlhan, G. Serbes, and N. Aydın, "Short Time Fourier Transform based music genre classification," *Biomedical Engineerings' Meeting*, pp. 1-4, 2018.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [14] A. Kathuria. (2018)"Intro to Optimization in Deep Learning: Vanishing Gradients and Choosing the Right Activation Function." PaperspaceBlog. <https://reurl.cc/XVDVWM> .(May 22, 2022).

- [15] Audionetwork(2020). "THE HISTORY OF ROCK MUSIC." <https://reurl.cc/yM1Me6> .(May 22, 2022).
- [16] C. Gillett(2018). "The origins of metal and how it found its place in the music industry." Young Post. <https://reurl.cc/GEZEb3> .(May 22, 2022).
- [17] C. Livingston, "Naming country music: An historian looks at meanings behind the labels," *Philosophy of Music Education Review*, vol. 9, no. 2, pp. 19-22, 2001.
- [18] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390-391, 2020.
- [19] E. Benetos and C. Kotropoulos, "A tensor-based approach for automatic music genre classification," *In IEEE 2008 16th European Signal Processing Conference*, pp. 1-4, 2008.
- [20] E. Benetos and C. Kotropoulos, "Non-Negative Tensor Factorization Applied to Music Genre Classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 1955-1967, 2010.
- [21] EnglishClub. "History of Country Music." <https://reurl.cc/MN1NbX> .(May 22, 2022).
- [22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
- [23] G. Sousa(2017). "Where Did Jazz Music Come From?" WorldAtlas. <https://reurl.cc/oQzQel> .(May 22, 2022).
- [24] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293-302, 2002.
- [25] H. A. and B. Jr., *Black Studies, Rap, and The Academy*. University of Chicago Press, 1995.
- [26] Heartexlabs(2015). "Labelimg." Github. <https://reurl.cc/xQxQ9N> .(July 19, 2022).
- [27] H. G. Kim, G. Y. Kim, and J. Y. Kim, "Music Recommendation System Using Human Activity Recognition From Accelerometer Data," *IEEE*

Transactions on Consumer Electronics, vol. 65, no. 3, pp. 349-358, 2019.

- [28] J. Roger(2005). "MFCC." <https://reurl.cc/gMkM2V> .(May 22, 2022).
- [29] JazzObserver. "The Origins of Jazz." <https://reurl.cc/Qbnbb2> .(May 22, 2022).
- [30] J. Nicholas (2022). "A brief history of classical music." GRAMOPHONE. <https://reurl.cc/1mjmeY> .(May 22, 2022).
- [31] J. S. Bridle and M. D. Brown, "An experimental automatic word recognition system," *JSRU report*, vol. 1003(5), no. 33, 1974.
- [32] J. W. Chen, C. H. Liu, and Y. F. Liao, "Deep Neural Networks for Audio Event Detection," *2016 Conference on Computational Linguistics and Speech Processing (ROCLING 2016)*, pp. 311-324, 2016.
- [33] K. Chang (2018). "短時傅立葉變換(Short Time Fourier Transform)原理及 Python 實現." 博客園. <https://reurl.cc/qNzNk3> .(May 22, 2022).
- [34] K. Hassen, H. Janßen, D. Assenmacher, M. Preuss, and I. Vatolkin, "Classifying Music Genres Using Image Classification Neural Networks," *Archives of Data Science, Series A (Online First)*, vol. 5, no. 1, 2018.
- [35] K. Kumar, C. Kim, and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," *IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pp. 4784-4787, 2011.
- [36] K. Myers (2016). "The History Of Pop Music In 5 Defining Decades." Culture Trip. <https://reurl.cc/m363IW> .(May 22, 2022).
- [37] L. Chaudhuri (2021). "What is blues music?" CLASSICALmusic. <https://reurl.cc/KQKQ09> .(May 22, 2022).
- [38] ListaDiscoteca. "History of Pop Music: Complete Guide about this Genre." <https://reurl.cc/qNzNVg> .(May 22, 2022).
- [39] L. E. Barrett, *The Rastafarians: Twentieth Anniversary Edition*. Beacon Press, 1997.
- [40] MasterClass (2022). "All About Disco: Inside the History and Influence of Disco Music." MasterClass. <https://reurl.cc/YX7XED> .(May 22, 2022).
- [41] MasterClass (2022). "Hip-Hop Music Guide: History of Hip-Hop and Notable Artists." MasterClass. <https://reurl.cc/3YrYX0> .(May 22, 2022).
- [42] MasterClass (2020). "Reggae Music Guide: A Brief History of the Reggae Genre." MasterClass. <https://reurl.cc/W1a1ND> .(May 22, 2022).

- [43] M. R. Nirmal and S. Mohan, "Music Genre Classification using Spectrograms," *International Conference on Power, Instrumentation, Control and Computing*, pp. 1-5, 2020.
- [44] N. George, *Hip Hop America*. Penguin Books, 2005.
- [45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [46] P. Antoniadis (2022). "Activation Functions: Sigmoid vs Tanh." Baeldung. <https://reurl.cc/dWzWyM> .(May 22, 2022).
- [47] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern recognition and artificial intelligence*, vol. 116, pp. 374-388, 1976.
- [48] P. S. Lokhande and B. S. Tiple, "A framework for emotion identification in music: Deep learning approach," *International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 262-266, 2017.
- [49] P. Wu (2020). "Yolov4 筆記：Optimal Speed and Accuracy of Object Detection." Medium. <https://reurl.cc/MN1N64> .(May 22, 2022).
- [50] R. Miracle (2020). "YOLOv4 中的 Mish 激活函數." 人人焦點. <https://reurl.cc/7p9p4N> .(May 22, 2022).
- [51] R. Rajan and H. A. Murthy, "Music genre classification by fusion of modified group delay and melodic features," *In 2017 Twenty-third National Conference on Communications (NCC)*, pp. 1-6, 2017.
- [52] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo, "Parallel Recurrent Convolutional Neural Networks-Based Music Genre Classification Method for Mobile Devices," *IEEE Access*, vol. 8, pp. 19629-19637, 2020.
- [53] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357-366, 1980.
- [54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *1997 Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [55] S. Egge (2017). "The Origins of Country Music." Norton Center for the Arts. <https://reurl.cc/XVDVAe> .(May 22, 2022).

- [56] S. Holden. "heavy metal." Britannica. <https://reurl.cc/RXnXYg> .(May 22, 2022).
- [57] S. Panwar, A. Das, M. Roopaei, and P. Rad, "A deep learning approach for mapping music genres," *12th System of Systems Engineering Conference (SoSE)*, pp. 1-5, 2017.
- [58] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927-939, 2016.
- [59] S. Vishnupriya and K. Meenakshi, "Automatic Music Genre Classification using Convolution Neural Network," *International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-4, 2018.
- [60] T. Grierson (2018). "The Origins and History of Rock Music." LiveAbout. <https://reurl.cc/AOnO3K> .(May 22, 2022).
- [61] T. Kobayashi, A. Kubota, and Y. Suzuki, "Audio feature extraction based on sub-band signal correlations for music genre classification," *In 2018 IEEE International Symposium on Multimedia (ISM)*, pp. 180-181, 2018.
- [62] T. Kim, J. Lee, and J. Nam, "Comparison and analysis of SampleCNN architectures for audio classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 285-297, 2019.
- [63] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, and J. Gan, "Unsupervised feature selection by self-paced learning regularization," *Pattern Recognition Letters*, vol. 132, pp. 4-11, 2020.
- [64] Y. H. Cheng, P. C. Chang, and C. N. Kuo, "Convolutional Neural Networks Approach for Music Genre Classification," *International Symposium on Computer, Consumer and Control (IS3C)*, pp. 399-403, 2020.
- [65] Y. H. Cheng, P. C. Chang, D. M. Nguyen, and C. N. Kuo, "Automatic Music Genre Classification Based on CRNN," *Engineering Letters*, vol. 29, no. 1, pp. 312-316, 2020.
- [66] Y. H. Cheng and P. C. Chang, "Long Short Term Memory Model Applied to Music Genre Classification," *ICSEVEN*, 2021.
- [67] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A Survey of Audio-Based Music Classification and Annotation," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303-319, 2011.