Artificial Neural Network Final Project :

以淺層 CNN 類神經網路來完成 Audio Classification

搭配資料增量的方式提升泛化能力

by s7023369667

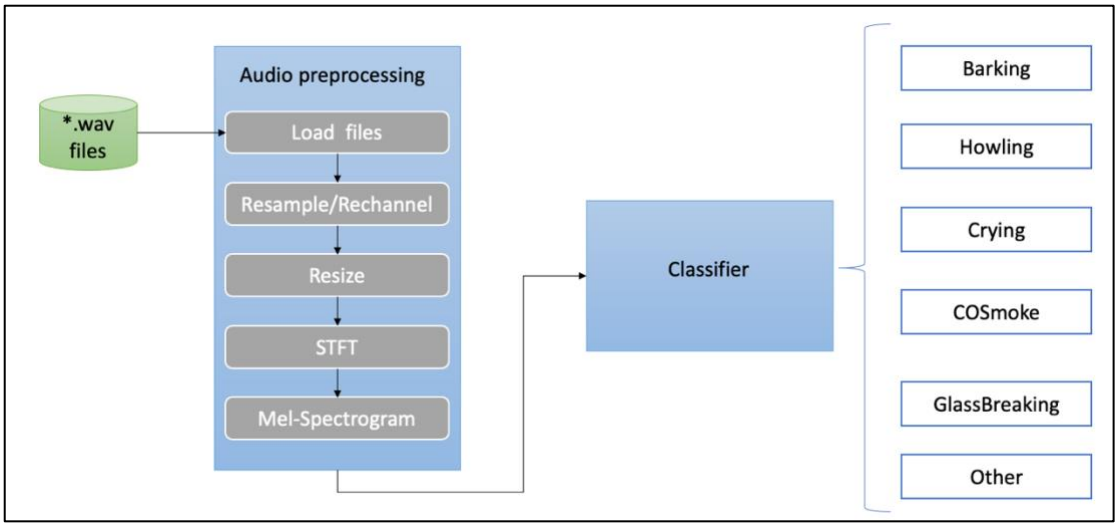
摘要：

本文使用淺層的 CNN 類神經網路模型參數的數量較少時計算的複雜度較低的情況下，不僅能節省成本也可以達到非常好的預測結果，在經過各式各樣針對模型在訓練中超參數的設定實驗企圖完成一個最佳解的問題，同時實驗了搭配資料增量的方式來增加音訊檔的分類成效，其中資料增量的目的是在針對訓練資料集當中所沒有的極端資料之特性來做資料增強且增量的方式，為了可以讓訓練過程也能有效學習到訓練資料集中沒有的特徵，同時也不會因此而讓訓練特徵受到混淆而導致訓練不起來的結果，也在最後數據證明發現我們的策略是有效的提升預測狀況，並且提升泛化能力。

研究目的及方法：

針對 2021 年度的 AI Cup 挑戰賽項目中的題目：“Tomofun 狗音辨識 AI 百萬挑戰賽”，並利用其官方所提供的訓練資料集以淺層的 CNN 類神經網路模型的方式來訓練分類器，並以此分類器來實現狗音分類的目的，本文所採用網路模型是經過 keras 並以 tenserflow 為底層框架的模型，以監督式學習的方式在已經經過官方包住分成六大類的 1200 個音檔進行訓練，在訓練過程中利用向前傳導以及向後傳導的方式來不斷地更新模型中的參數，訓練目標是找到最佳的模型參數，如此一來採用訓練結果之最佳模型參數來製作我們的分類器，最後將官方提供的測試資料集通過我們的分類器來進行分類。而其中在未進行資料增量前的訓練資料共有 1200 個音檔，而測試資料則有 10000 個音檔，其中測試資料包含很多極端、充滿背景雜音等的音檔，所以必須為模型添增泛化能力，否則在訓練結果即使非常好的情況下，預測結果可能也不盡理想，於是採用資料增量的方式來完成提升泛化能力的目標，將訓練資料增加為 3600 個音檔，其中包含原本的 1200 個音檔以及增量的 2400 個音檔，最後訓練結果大幅的提升，且預測結果也有相當程度的提升，代表策略所帶來的泛化效果是有增加的，可見我們的策略是有效的。

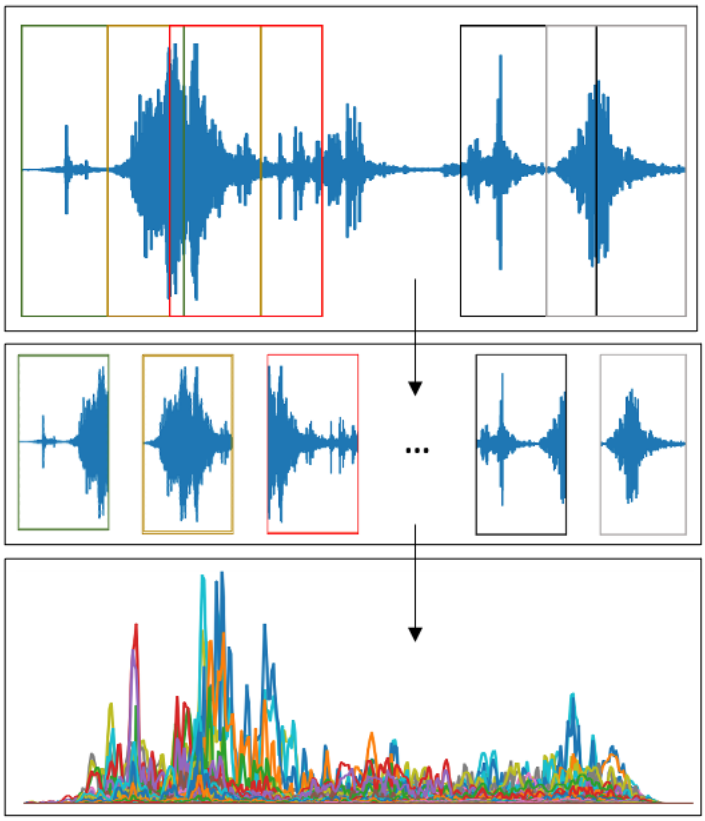
圖表 1：將全部的音訊檔(.wav)，經過一連串的音訊前處理後放入模型進行訓練，依據官方所標記的 label 來進行監督式學習，將結果分成六大類{Barking , Howling , Crying , COSmoke , GlassBreaking , Other}。



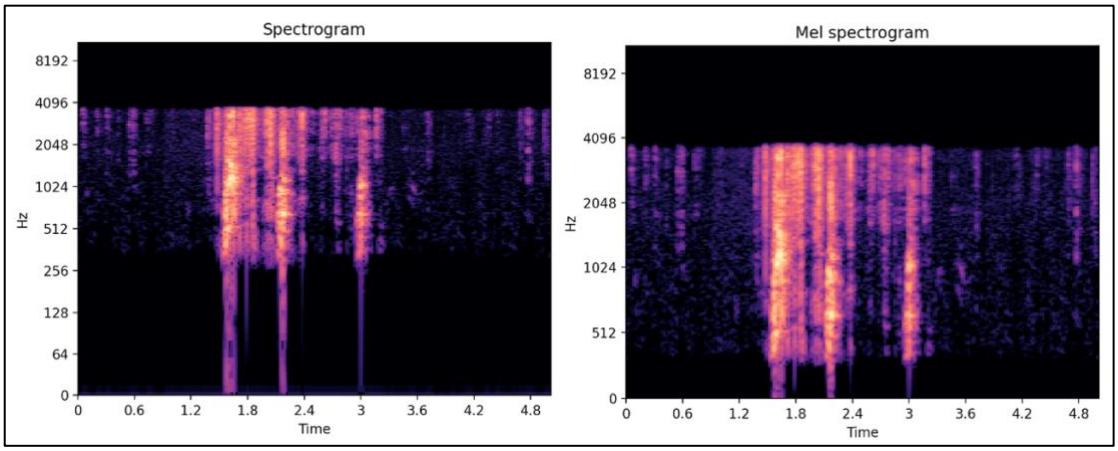
音訊前處理：

從圖表 1 可以完整看出我們的工作流程。將音檔透過 python 第三方套件 librosa 讀入，並透過 librosa 的內部函式完成很多處理，首先必須根據讀入的音檔進行 Rechannel(目標是將所有的音檔都進行單一聲道的轉換)根據官方需求，並且將音檔讀入後的長度進行 Resize 的統一後進行 STFT(Short-time Fourier Transform)，採用 STFT 的原因是在針對非週期性訊號的處理中，我們非常需要頻率和振幅在時間軸上的表現，當我們對整個訊號直接作快速傅立葉轉換時，會喪失跟時間軸的對應關係，所以我們必須用一個相等長度的 window 來把訊號切成一塊一塊且有部份 overlapped 的方式，才對切出來的每塊 windows 各別進行快速傅立葉轉換，以留下我們所需要的頻率在時間軸上的振幅特徵，FFT 的大致做法是將訊號進行採樣後從 time-domain 轉成 frequency-domain，並將各別取得的傅立葉轉換在平面上進行重疊來組成我們所需要的頻譜，但取得這個頻譜後，其實還不夠讓我們拿這個特徵直接進行後練我們的分類器，我們還必須對這個頻譜做梅爾值的對應，因為在人類進行分類時是大致上依據頻率和振幅在時間序列上的改變來進行分群分類，但人耳卻在對於頻率的敏感度上並不是那麼敏銳，是呈現一種隨著頻率越高而越遲鈍的非線性的關係，所以特別對於越高頻的頻率敏感度會呈指數下降，所以這樣直接取用的頻率對我們的分類上沒有直接幫助，甚至會因此而誤判，所以我們必須將前面所得到的頻譜進行梅爾值得轉換，將頻譜通過梅爾濾波器進行 one-to-one 對應後，透過這個轉換讓我們不再受限於上述情況，而所得到的梅爾頻譜就是我們要的特徵序列，由圖表 3 可以看出原本的頻譜和梅爾頻譜得差別，就在我們具備了所需要的特徵，接著就是將特徵序列丟進我們的模型進行訓練，採用監督式學習的方式搭配官方所給的對應 label，讓模型逐漸學習到分類的能力。

圖表 2：Short-Time Fourier Transform(STFT)的示意圖



圖表 3：左圖是經過 short-time Fourier transform 後重疊所得到的頻譜圖、
右圖是將左圖的頻譜再經過梅爾濾波器的 mapping 後所得到的梅爾頻譜圖

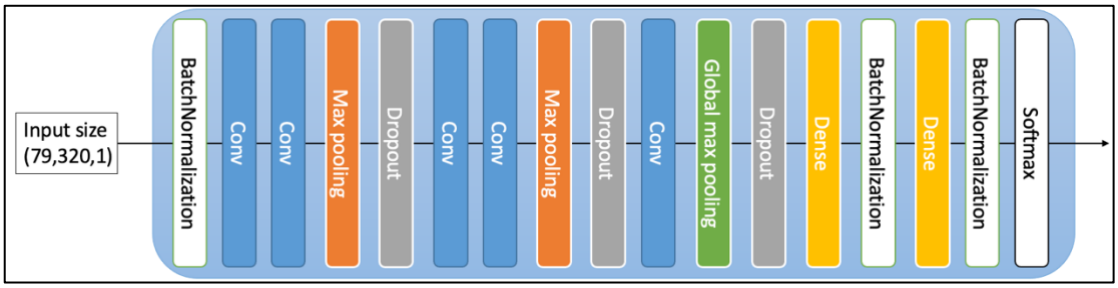


模型架構：

由圖表 4 的模型架構是採用淺層的 CNN 類神經網路架構，input layer 加上 BatchNormalization 並使用 momentum=0.9 連接兩層 Convolution layer filter 皆設定為 16，接 Maxpooling 和 Dropout(0.1)，連接兩層 Convolution layer filter 皆設定為 32，接 Maxpooling 和 Dropout(0.1)，連接上一層 Convolution layer filter 設定為 128，接 Globalmaxpooling 和 Dropout(0.1)，接 Dense 和 BatchNormalization

並使用 momentum=0.9，接 Dense 和 BatchNormalization 並使用 momentum=0.9，output 經過 softmax 激活函數，取得對於屬於每一類的機率值，總參數量只有 91450。

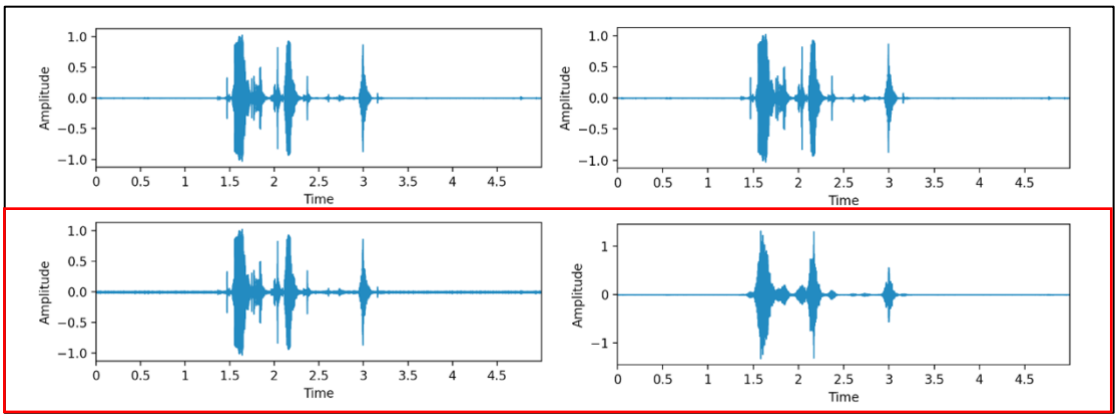
圖表 4：模型架構



資料增強、增量：

為了提升模型在預測時的泛化能力，針對訓練資料集中沒有測試資料集中的雜訊、極端高頻音訊等等來做資料增強，分別對於原本的音訊檔加上白噪音以及做音調增高這兩種增強方式，且將原先的訓練資料 1200 個音檔都進行這兩種資量增強處理後的 2400 個音檔，所以總共 3600 個音檔都經過前處理後批次放入模型進行訓練。由圖表五可以看出原始波形再經過增強後的差別，由圖表六可以看出我們丟進模型訓練所使用的一些超參數數值。

圖表 5：紅框外的是原始音訊檔的波形圖，其分別往下對應的是
加入白噪音的結果（左圖）以及經過音調調整的結果（右圖）



圖表 6：資料增量後的訓練集和驗證集的切分，以及模型參數的使用。

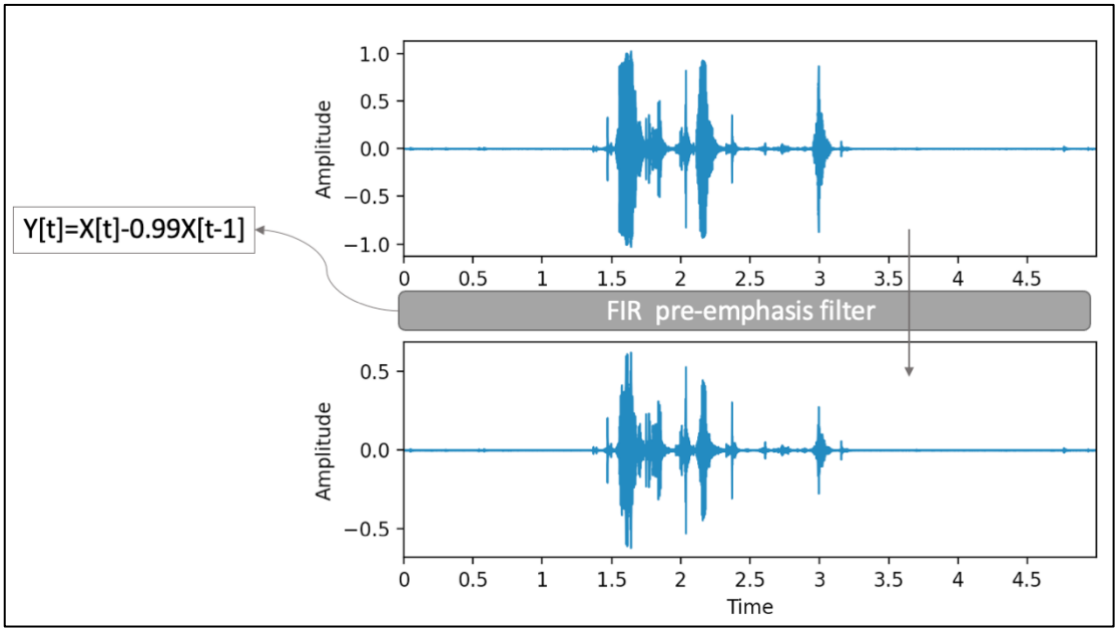
	Train	Predict
Total datafiles	3600	10000
Train files	3240	
Validation files	360	
Batch size	64	64
parameters	91450	
Dropout rate	0.1	0.1

考慮到測試資料集的多種類複雜性，不只是吵雜的背景雜訊或者極端高頻的音檔等等，我們的模型在學習的過程中都是沒有經歷過的，所以將給予測試集的每個音訊檔都經過一個自制的濾波器，目的是將背景雜訊和音調的部分來進行處理，而濾波器是使用 FIR 的 LTI 系統，公式所夾帶的係數α做為濾波係數來調和雜訊和音調。本文是採用α =0.99來做為濾波係數，由圖表 7 可以看出原始波形通過濾波器後的波形，變得比較平滑、也讓厚度變得比較輕盈，但實際打開音檔來比較沒有太大的差別。

公式：FIR 的 LTI 系統

$$Y[t]=X[t]- \alpha X[t-1]$$

圖表 7：上圖為原始音訊檔的波形圖，下圖為經過濾波器處理後的波形圖。



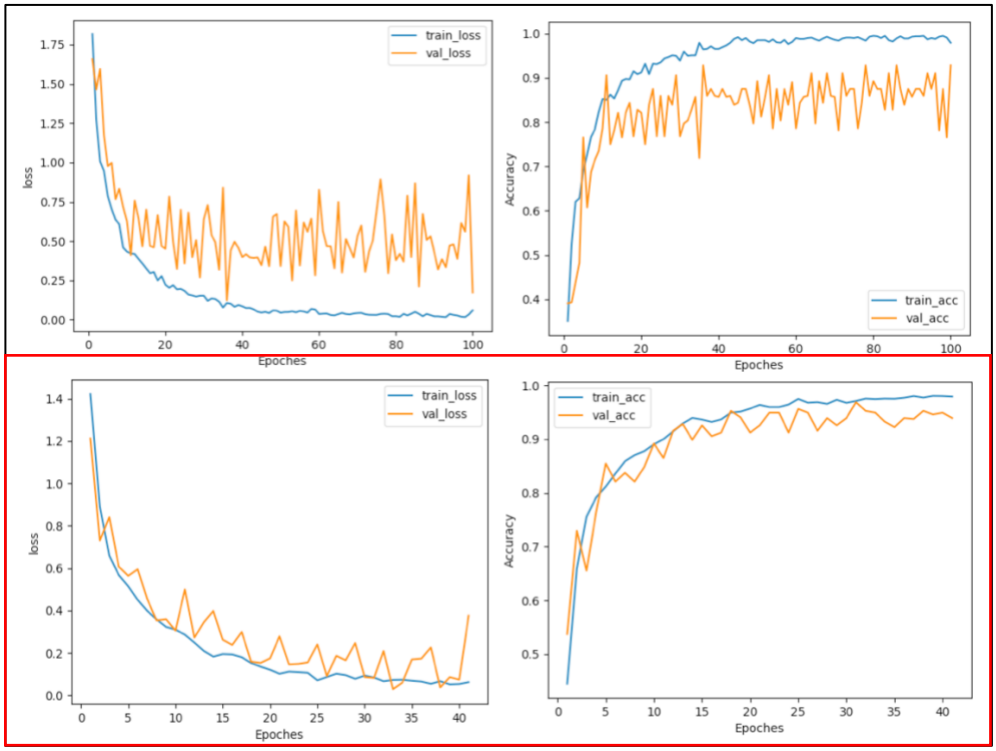
實驗結果：

圖表 8：分隔線上下方分別代表在未使用和使用資料增量時最佳模型對於訓練集的分類報告。

Before	Braking	Howling	Crying	COSmoke	GlassBreaking	Other
Precision	0.92	0.83	0.92	0.87	0.94	0.89
Recall	0.86	1.00	0.67	0.93	0.94	0.94
F1-score	0.89	0.91	0.77	0.90	0.94	0.92
support	28	24	18	14	18	18
After	**Braking**	**Howling**	**Crying**	**COSmoke**	**GlassBreaking**	**Other**
Precision	0.81	0.96	0.95	0.94	0.96	0.98
Recall	0.98	0.92	0.93	1.00	0.98	0.84
F1-score	0.88	0.94	0.94	0.97	0.97	0.91
support	51	73	68	47	52	69

經過一連串的訓練，在沒有使用資料增量的方式前我們的訓練狀況比較不佳，猜測是模型太強又搭配 Dropout 的方式所以起伏才會這麼大，而且訓練剛開始很快的 loss 曲線就立刻面臨瓶頸，起初針對很多超參數做調整，甚至是根換過成其他的模型，但還沒找到適合的模型導致效果都不太好，於是在嘗試了將近兩個禮拜後決定先對預測泛化程度先試著做提升，才想到可以增加資料的強度以及使用其他開源資料集的方向著手，後者沒做的原因是我覺得模型在初步訓練的情況其實就有抓到重點特徵，只是在面對太吵雜的背景會受到混淆，所以如果我取用其他的資料集不難保會更受混淆，於是考慮對官方給的資料集進行增強，並增量加進模型進行訓練，而結果也非常如預期的往好的方向發展，由圖表 8 的報告可以看出 f1-score 對於資料增量前後的 precision 和 recall 的調和平均數有大幅的差異，也由圖表 9 可以看出資料增量後的結果對於訓練也有實質上的影響。

圖表 9：紅色框內的是資料增量後的訓練結果，並利用 keras 的 callbacks 函式中的 EarlyStopping 和 ModelCheckPoint，讓模型在 overfitting 之前就停止訓練並且存下最好的參數模型。



結論：

在經過一整個月的研究，對於第一次做音訊分類問題的我來說，學到非常多的相關知識，也還好能搭配這學期修的 DSP 讓我在對訊號的處理上不會太坎坷，最後自己架的模型，以及自己做的資料增量的策略能達到這麼高的效果真的非常開心，在對於類神經網路方面的研究有更清楚的方向，雖然比賽最後沒有晉級，最終以上傳評分 96.2 告終，但未來相信還有機會再繼續對音訊處理方面的研究或比賽再次參加且根據這次所學能有更好的發揮。