目錄

摘要

近日網路發達，影片的流通更為便利，但為影片加上適當的音效往往需要耗費許多人力與時間資源。本研究旨在研究利用人工智慧深度網路技術訓練模型，並期望訓練後的模型能為當下影片的內容配上適當的音效，使得幫影片配音的過程可以加速，並減少人力需求。本研究主要利用 Keras 這個 Python 套件創建模型，並參考 ResNet 這個著名的深度學習模型，透過大量已標記圖片當作訓練資料，並進一步修改模型的架構，使準確率得到提升。研究結果符合預期，能在適合的類別中為影片配上合理的音效。

1 研究動機

平常在看許多 youtuber 的影片 ，發現他們都在影片中加入很多音效，製造氣氛，我們也發現音效真的是影片中一很個重要的元素。但是我們相信當影片創作者在構思要在何處放音效、又要實際將音效放到影片之中，想必相當麻煩複雜。因此我們希望做出一個能根據畫面內容而配出相對應音效的系統，只要匯入影片檔，就能幫影片適時的加上各種音效。

2 文獻探討

關於影片配音效及影像辨識的文獻，我們找到以下幾篇發表論文：

視覺影像 (Visually indicated sounds) [1]

此篇研究目的希望從敲擊物品的無聲影片中，模擬、生成對應的聲音。

研究團隊提出的方法先以捲積神經網路（Convolutional Neural Network,CNN）對影片的影像進行特徵提取（feature extraction），再將代表圖片特徵的隱藏向量（latent vector）通過循環神經網路（RNN, Recurrent Neural Network），在每個「音訊採樣點」上回傳代表對應音訊特徵的隱藏向量，最後每一小段時間以音訊資料庫中隱藏向量最相近的片段作為其音效。綜上，其解決方法如圖 1所示。

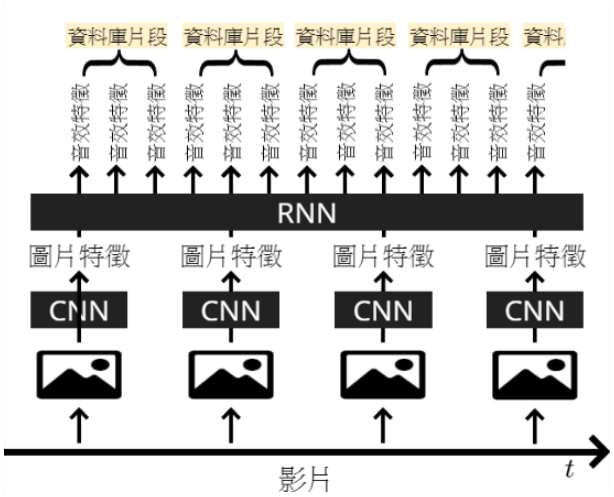

圖 1: Visually Indicated Sounds 流程示意圖

殘差神經網路（Residual Network, ResNet）[7]

ResNet 是利用 identity connection 將一層的輸出跨越多層後連結到另一層（直接作為該層的輸入），如圖 2為跨越兩層的實例。藉由這個小改變，直接的連結讓梯度可以跨越層的傳播，解決梯度消失的問題，使訓練深層的神經網路更加容易。

在影像辨識系統上，有許多捲積神經網路的延伸結構，我們以曾經在多個影像辨識比賽獲得冠軍的 ResNet，作為本次實驗將使用的模型。

儘管我們沒有足夠的時間和資源蒐集大量的資料、訓練較龐大的 RNN 模型，依然在許多方面都能參考文獻中研究的結果：

- 將影片每個畫面通過影像辨識系統，進行特徵提取。
- 以資料庫中的音效為其配音效。

3 研究目的

本研究的目標為用機器學習的方法，訓練一個可自動幫影片配出符合當下影片內容的音效之自動配音人工智慧系統。

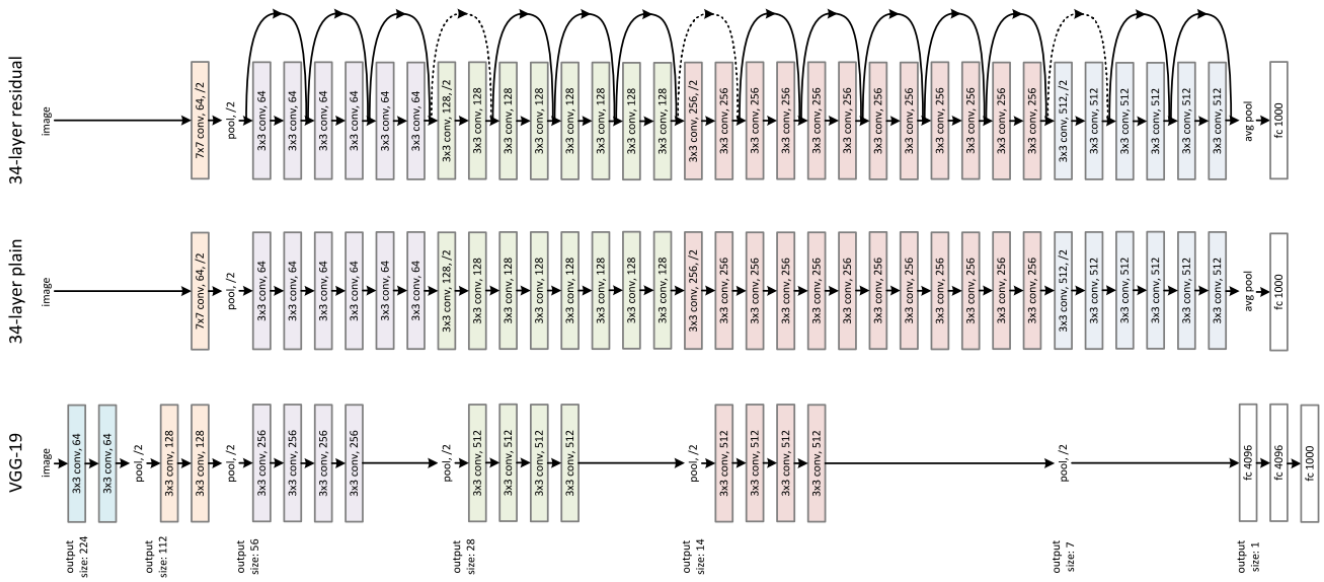

圖 2: ResNet 示意圖

4 研究方法

4.1 研究工具

4.1.1 器材

- 電腦 ×3
- 實驗室遠端伺服器

4.1.2 資料

- 圖片：Google Search、unsplash.com[11]
- 影片：YouTube[10]
- 音效：freesound.org[9]

4.1.3 軟體

- Google Colab[2]：線上的 Python 編輯、執行環境，不需要自己下載套件，專為機器學習的研究而設計。

- Keras[3]：現今最廣泛應用在機器學習上的 API，提供簡單的語法來建構自己的深層神經網路。
- `google_images_download`[4]：Python 套件，可大量下載 Google Search 的圖片。
- `PIL`[5]：Python 套件，可快速進行影像處理。
- `matplotlib`[6]：Python 套件，方便將數據、圖片視覺化。
- `OpenCV`[8]：Python 套件，可擷取影片中的畫面。

4.2 研究流程

對於自動配音效的問題，最初的構想將其簡化成幾項子任務分而治之：

1. 針對單一圖片分類到適合該圖片的音效類別
   (a) 蒐集訓練資料
      - 下載大量圖片
      - 進行圖片前處理
   (b) 建構神經網路
   (c) 訓練、調整類神經網路
   (d) 討論模型表現
2. 利用模型為影片配音效
   (a) 擷取影片的圖片
   (b) 以模型為每個畫面 (frame) 標記類別
   (c) 處理標記，並配上音效。
   (d) 討論配音成效
3. 針對結果調整模型

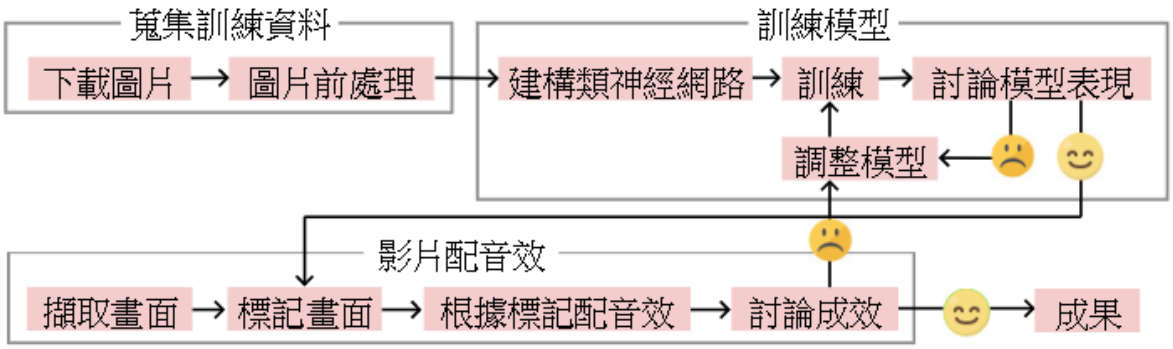

圖 3: 研究流程圖

5 研究過程

我們透過視訊與荷蘭的科學動畫繪圖師討論，理解了這方面產業的需求，於是我們擬定了大致上的研究流程與目標。

按照討論出來的的流程，我們在網路上搜尋到適合的套件，成功處理影像、建立自己的深層神經網路，然而訓練完成的模型為影片預測、標記、配音後的結果並不符合預期。我們推測是模型結構不夠完善，以及訓練資料不足所致。為了解決以上的問題，我們調整模型，改以遷移學習（Transfer Learning）的方式改進深層神經網路的表現，最終大幅增加訓練結果的正確率。

依然，我們發現原本的構想未考慮到無聲（不配音效），或太多音效混雜的情形，因此在訓練資料上做些調整，使模型能將「無音效」也視為一個類別，結果有稍微改善，但辨識率能仍再提升。

詳細實驗過程說明如下：

5.1 下載圖片資料庫

最初上網尋找配有音效的影片，並擷取每一種音效出現時的畫面作為訓練資料。但隨著模型漸大，需要的資料量快速上升時，我們上網找到 `google_images_download`[4] 這個 Python 套件，可以用指令大量下載利用 Google 搜尋引擎搜尋到的圖片。利用自動化的下載方式節省了一張一張截取的時間，但可能造成蒐集到的圖片有較多的雜訊。

作為實驗，我們先選用 20 種音效相關圖片類別進行辨識（總數為 2804 張），每個類別與圖片張數如下表所示。

類別	數量	類別	數量	類別	數量	類別	數量
train passing	183	water drop	151	river	97	stars shining	161
lightning	145	writing	153	screaming	80	high heels	192
wave	196	typing	104	ambulance	74	dog barking	198
cars	187	door opening	133	glass breaking	120	cat meowing	184
cheer	183	fire	94	helicopter	96	blender	75

5.2 進行圖片前處理

一般圖片辨識神經網路的輸入會以圖片每個像素的 RGB 數值作為輸入，即以圖片長度、圖片寬度、RGB 這三個維度構成的 長 × 寬 × 3 三維陣列為輸入。

而為了方便處理，通常會將圖片變成相同大小（長寬相同），再進行訓練。對應的，我們也找到專門處理圖片的 Python 套件 `PIL`[5]，可以支援將圖片以**重新取樣**的方式縮放，讀取圖片則以 `matplotlib`[6] 套件將圖片檔轉換成上述的三維陣列。而本次實驗我們統一將圖片轉換成 224 × 224 的大小。

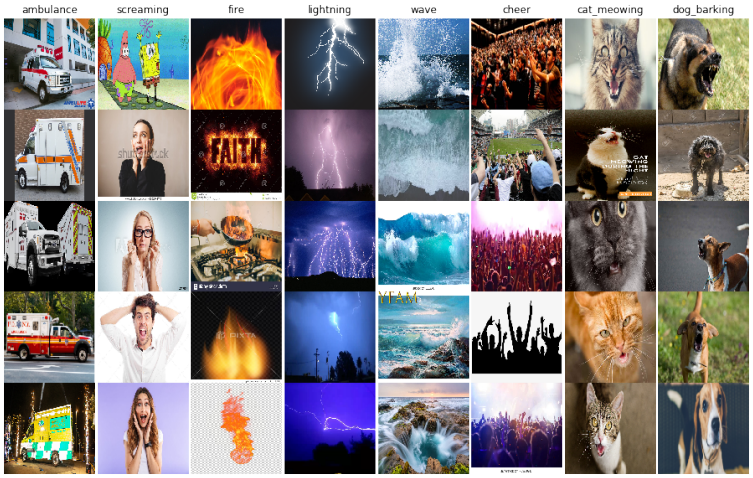

圖 4: 資料庫的樣本

5.3 建構深層神經網路

起初我們尋找最適合應用在圖片辨識的神經網路，除了最基本的捲積神經網路（CNN）外，有許多研究團隊開發出更優化的結構諸如 ResNet、Inception、NASNet 等等，最終我們以 Keras 提供的 ResNetV2 範例為架構，搭建 5 個區塊（blocks）的 ResNetV2 進行訓練。

5.4 訓練、調整深層神經網路

針對圖片辨識的模型訓練時，常常使用資料擴增（data augmentation）的方式增加資料數量及多樣性，藉由旋轉、平移、縮放... 等不同的方式作用在原圖上，使相同的資料可以被複製多次使用，更可以期望模型學習到稍微變化的圖片依然屬於同個類別。本次實驗詳細使用的資料擴增參數如下表所列：

旋轉角度	15°	橫向平移	15%	垂直平移	15%	裁切	15%
縮放	15%	色調平移	10%	水平翻轉	True		

訓練時我們將 20% 的資料當成驗證資料（validation data）和測試資料（testing data），用以監督、客觀驗證模型的表現。在使用 Adam 優化下，我們跑了 150 次（epochs），訓練紀錄如圖 5所示，最終 20 類分類（測試資料）正確率約為 76%。

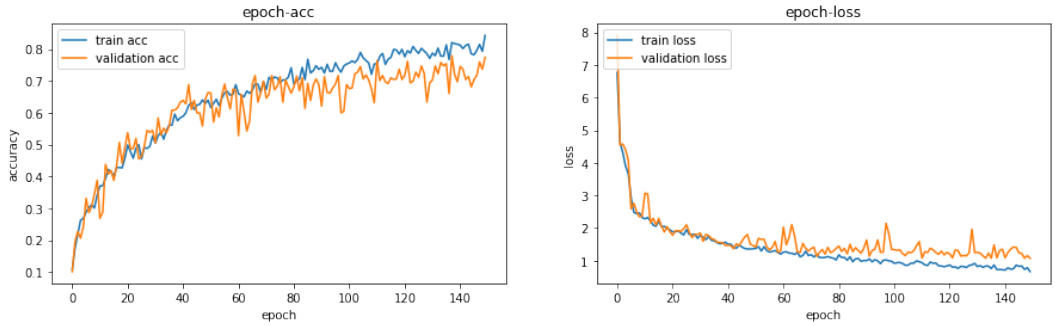

圖 5: 訓練結果

5.5 討論模型表現

除了正確率外，為了深入探討模型在各個類別的表現，我們利用混淆矩陣（confusion matrix）來檢視每個類別的分類情形。混淆矩陣是將實際類別與辨識類別的關係，以大小為類別數 × 類別數 的矩陣呈現，製成圖表，方便視覺化模型在每個類別下的表現。

假設第 i 張圖片的類別為 C_i，辨識結果為 P_i，則混淆矩陣 M 的元素可定義為：

$$M_{i,j} = \frac{|\{t|C_t = i \wedge P_t = j \wedge 0 \leq t < |C|\}|}{|\{t|C_t = i \wedge 0 \leq t < |C|\}|}$$

而我們依據模型在測試資料上的預測繪製了圖 6的表格。

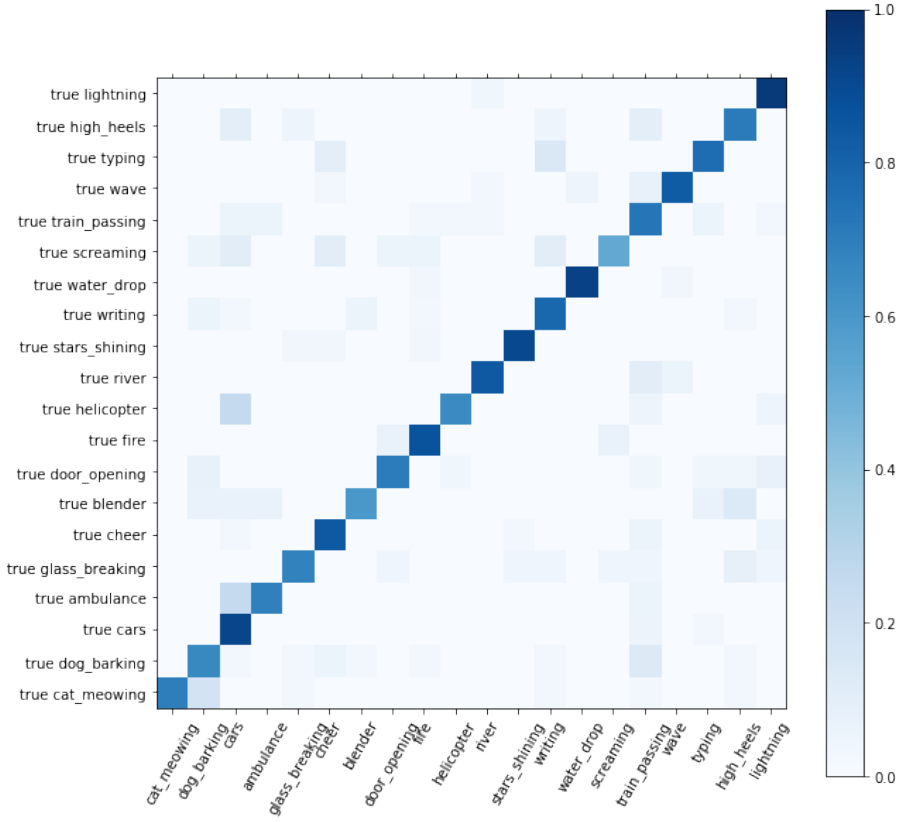

圖 6: 混淆矩陣

從混淆矩陣與正確率可以看出，分類情況應該是符合預期的，各個類別雖有錯誤，但都不會有太大的偏差。值得一提的是，混淆矩陣非正確的統計中，在 (`true helicopter`, `cars`)，也就是直升機被誤認成車子，的情形是較明顯、較常出現的。同時我們也發現車子和直升機的形狀確實有些相近，因此模型的預測雖有錯誤，但卻是可以被理解的，這樣的解釋亦可套用在其他較常出錯的例子上。

5.6 擷取影片中的圖片

如前面提及，在經過神經網路標記前需要是固定大小的三維陣列圖片，因此我們找到 `OpenCV`[8] 這個套件，可以將影片中的每個畫面擷取成 (長 × 寬 × 3) 的格式。再用前提的 `PIL` 套件，將圖片縮放成固定 (224 × 224) 的大小，即完成影片的前處理作業。

5.7 以模型為每個畫面標記類別

取得每個畫面（frame）的三維陣列後以訓練好的模型預測之，可以得到每幀對應的類別機率分布。考慮到辨識的結果可能有很多雜訊（辨識情形並不穩定），若在標記時僅考慮單張畫面的類別，生成的音效會出現斷斷續續的問題。因此在處理標籤的部分我們採用**眾數濾波**來處理雜訊，即每個畫面的類別標記定為其前後 k 秒內畫面的類別中的眾數（前後 k 秒內出現最多次的類別）。假設第 i 個畫面的類別為 C_i，fps 代表幀數（每秒畫面數量），則第 i 個畫面的類別標籤 T_i 為 $\{C_{i-k\times fps}, C_{i-k\times fps+1}, \cdots, C_{i+k\times fps-1}, C_{i+k\times fps}\}$ 中的眾數。

取得一連串的類別標記後，將相連、相同的標記配上該種音效，若該類別持續出現的時間大於資料庫中音效之長度，為求方便先以重複播放的方式解決。

以此較簡化的演算法，避免文獻探討時提到蒐集大量音效、將音效轉化成音效特徵向量的需求，意味著將音效特徵從原本的多維向量簡化以類別（一維向量）取代，損失了相同音效應有的多樣性，但也降低了研究的複雜度。

根據上述方法，首要任務為尋找每個類別相對應的音效，因此我們上網找到免費的音效庫 free sound[9]、蒐集完 20 種類別對應的音效，並在 youtube 上找到與我們所選取類別較相近的影片 [10] 作為測試影片，以上述的演算法為此影片配音效，影片結果為以下連結中的 video 1,2,3

20 種類別音效：

https://drive.google.com/open?id=1y1z2Zro4AIyIwP9E4YthGgjQcdmt_wBW

測試影片配音結果：

https://drive.google.com/open?id=1IFdHlgDYlSo4akKTXUX5HmrFM7B4xqol

5.8 針對影片結果調整模型

對於生成出的音效，我們發現以下問題：

1. 有些片段不應該出現音效，但模型會將它分在最有可能的類別，代表每張圖片都會被配上

音效，並不符合預期。因此要設計能處理「其他類別」的機制。

2. 模型無法非常正確地分出圖片的類別，推測是結構不夠完善抑或少量訓練資料無法應付多變化的圖片。

5.9 其他類別（otherwise class）

處理其他類別最直覺的想法就是在類別中新增一個其他類別，並蒐集許多不屬於這 20 種類別的圖片作為該類別的訓練資料。因此我們在 unsplash[11] 網站上找到隨機的圖片加入其他類別的訓練資料，解決其他類別的問題。

值得一提的是，在選取空類類別時，我們特別加強一些模型較易認錯的類別。例如當模型在辨識尖叫時，卻將所有人臉的圖案都辨識成尖叫。因此我們在其他類別中特別加入一般人臉的圖樣，以期模型能認出其差別。

最終我們額外蒐集 201 張圖片作為其他類別的訓練資料，所有資料總數為 3005 筆。

5.10 遷移學習（Transfer Learning）

遷移學習是一種將已經訓練完成的模型，套用在不同神經網路上的技術。若選用同是圖片辨識的神經網路，即使辨識的類別不盡相同，已訓練過的神經網路依然包含有進行圖片辨識需要的**特徵提取**的資訊。換句話說，已訓練過的神經網路已學會如何擷取圖片的特徵，因此在訓練時只需要做些微調整（fine-tune），即可應用在特定的目標類別上。

本次實驗我們選用 Keras 公開的 ResNet152V2 模型以及在 imagenet[12] 上訓練的權重作為引入目標。

以圖 2為例，我們將前段的捲積區段稱為 (特徵) 提取網路（extraction network），為**特徵提取**的階段；最後的全連接層（fully connected layer）稱作決策網路（decision network），為決策的階段。Keras 提供的 ResNet 是針對 1000 個類別進行分類，若要將其調度到我們的目標上，則需重現建立、訓練決策網路並微調提取網路。故我們以以下方法進行遷移學習：

1. 載入 Keras 提供的 RseNet152V2

2. 去除最後的全連接層（保留提取網路），並建立新的決策網路（輸出為 20 類）

3. 固定提取網路，訓練新建的決策網路 100 次（epochs）

4. 訓練整個網路（fine-tune）100 次（epochs）

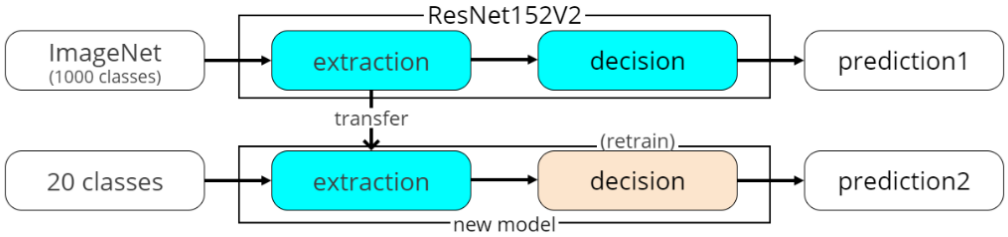

圖 7: 遷移學習示意圖

5.11 討論模型表現

我們畫出訓練的紀錄以及混淆矩陣，如圖 8,9所示。其中在圖 8在第 100 次（epochs）時正確率下降為 fune-tune 初期的影響。最終訓練達到 86%，與 5.4相比高出 10%。

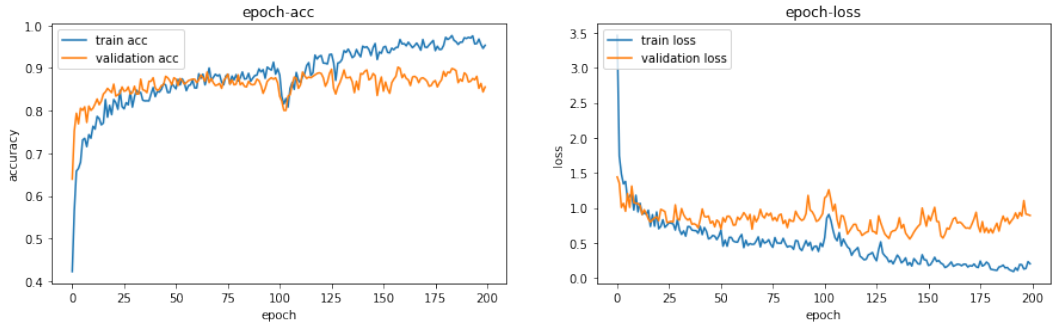

圖 8: 訓練結果

接收者操作特徵曲線（Receiver Operating Characteristic curve, ROC curve）[13]

接收者操作特徵曲線為二元分類問題時一種評估模型表現的工具。若將「有音效」訂為陽性，「無音效」訂為陰性，則定義：

- TPR（true positive rate）：在所有實際為陽性的樣本中，被正確地判斷為陽性之比率（陽性樣本辨識率）。

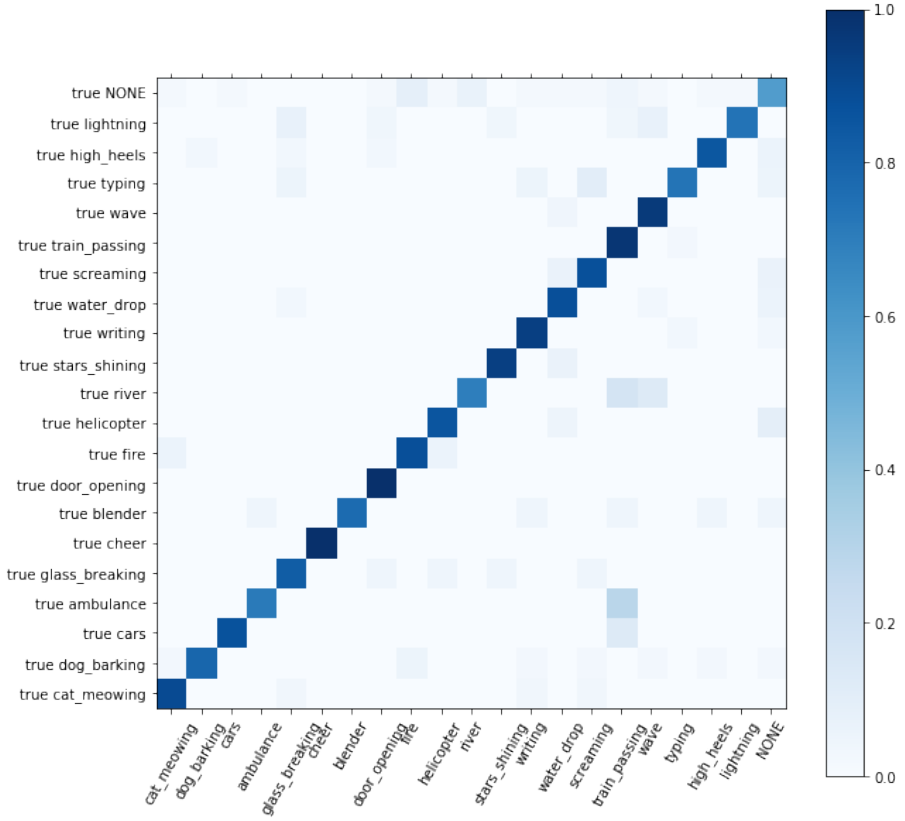

圖 9: 混淆矩陣

- FPR（false positive rate）：在所有實際為陰性的樣本中，被錯誤地判斷為陽性之比率（陰性樣本誤判率）。

在模型預測出的機率分布上設定不同的閾值（threshold）會造成不同的辨識結果，例如閾值設定為 0 則代表所有樣本皆被辨識為陽性，閾值設定為 1 代表所有樣本皆被辨識為陰性。

接收者操作特徵曲線即在閾值的變化下，將座標點 $(x = FPR, y = TPR)$ 在座標上變動軌跡而成的曲線。正常情形下，接收者操作特徵曲線應為從左下連到右上的曲線，而曲線越偏左上角（FPR 越小，TPR 越大），代表模型的分類效果越佳。客觀表達之，即曲線下面積（Area Under the Curve，AUC）越大代表模型的效果越佳。

綜上，我們將其他類別視為陰性，其餘類別視為陽性，繪製出接收者操作特徵曲線如圖10所示。

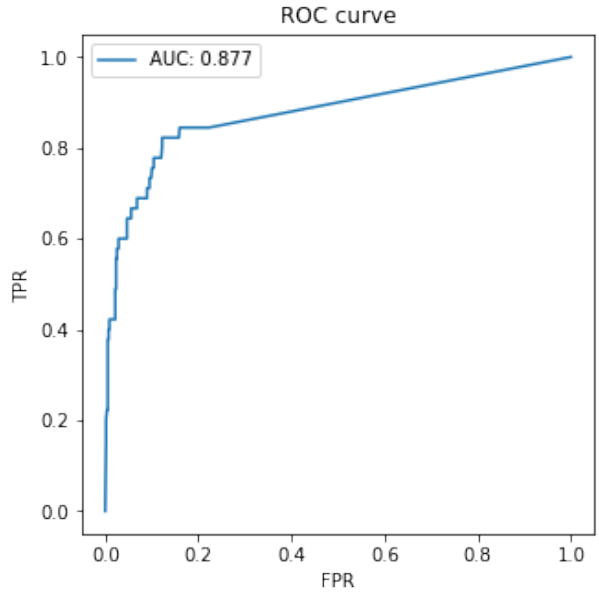

圖 10: 接收者操作特徵曲線

5.12 影片測試

我們以相同的三部影片測試，結果如以下連結中的 video 4,5,6。可看出配音效結果有大幅改善，但依然有部分片段出現不合理的配音，仍有改進的空間。

https://drive.google.com/open?id=1IFdHlgDYlSo4akKTXUX5HmrFM7B4xqol

6 研究結果

我們利用遷移學習的技術達到 86% 的圖片辨識率，並依據模型的預測結果設計演算法為影片配音效。在測試影片上可以看見一定的成效，但若要更趨近人們理想的結果，仍需要相當的改進。

7 未來展望

增加音效類別

目前支援的類別僅有 20 類，但常見的音效遠不止這些，因此希望可以蒐集更多類別的圖片。

以循環神經網路作為決策網路

儘管利用眾數濾波的技術處理標籤，預測結果依然會有短時間內不斷改變音效的情形。若以循環神經網路，同樣在每個時間點回傳該畫面的類別機率分布，如此在進行分類時考慮到時序問題，可以期望在一連串同標籤連續出現時，接下來出現的機率也會較高（趨於同標籤連續出現）。

除了上述的優點，我們也希望用循環神經網路代替「新增其他類別」以處理無音效問題。由前面測試影片可以看出其他類別的作用並不如預期的有效，我們推測「將其他類別視為一個類別」未必是最好的做法。因此若改以循環神經網路學習透過觀察連續畫面的類別機率分布，可以期望它在各類機率皆低、分布不穩定時將其分為其他類別。

新增音效屬性

每種音效都有各自的特性，由於 5.7中提到為求簡便，過長的連續標籤我們以重複播放音效的方式解決。但未必所以音效重複播放都是合理的，例如河水聲、歡呼、海浪聲屬於適合重複播放的；然而尖叫聲、玻璃破裂聲、尖叫聲就不適合重複持續播放。

因此在音效處理上我們可以將每個音效都配上適合的屬性。而屬性的類別未必只局限於重覆與不重複，將來收錄更多音效後，亦可能需要更多類別使音效以最適合的方式任意長短呈現。

音效平行播放

在我們演算法當中同一時間僅會有一種音效出現，然而若畫面中同時出現多種不同類別的物件，能讓這些音效同時出現應是較符合預期的結果。因此我們期望在配音效上能針對每個類別個別處理。我們提出的構想為針對每個類別都訓練一個二元分類模型（CNN+RNN），分類是否要配上該類別音效。其中在特徵提取階段可以適度的共用網路（共用同一部分的特徵提取網路）。然而此種方式在訓練資料的產生（手動標記影片）上較不易，需要大量的時間完成，因此目前較無法達成。

參考資料

[1] Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E. H., & Freeman,W. T. (2016). Visually indicated sounds.

[2] Google Colab：https://colab.research.google.com

[3] Keras：https://keras.io/

[4] `google_images_download`：https://pypi.org/project/google_images_download/

[5] `PIL`：https://pillow.readthedocs.io/en/stable/

[6] `matplotlib`：https://matplotlib.org/

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for imagerecognition.

[8] `openCV`：https://opencv.org/

[9] free sound：https://freesound.org/browse/tags/sound-effects/

[10] 測試影片：
https://youtu.be/hw_jOnb6Jr0
https://youtu.be/HAF0loQa9jo
https://youtu.be/SzHQkjnwsQw

[11] free image：https://unsplash.com/

[12] imagenet：http://www.image-net.org/

[13] ROC curve：https://zh.wikipedia.org/wiki/RO%E6%9B%B2%E7%BA%BF