

Interface combined of audience emotion and movie content

觀眾情緒、電影內容結合介面系統建置

組別：A25 組員：李佳璋、吳沛芸、賀文芸 指導教授：李祈均 教授

摘要

本專題想要實現一個能夠呈現觀眾情緒的介面。此介面可以呈現觀眾在看預告片時的各個情緒比例，並有回放功能使觀眾可以看到當時的情緒反應。介面的部分主要是以 PyQt 為架構來設置，錄製觀眾觀看預告片時的反應。錄製完成後的影片依時段切分成圖像，以 Emopy 這個預先訓練過的模型，對每張圖像預測不同情緒所佔比例。然後判斷觀眾當下主要為何種情緒，並輸出最能代表此情緒的圖像。

前言

影視產業日漸興盛，在網路上也出現許多網紅、youtuber 等會拍攝自己觀看影片內容的反應，而這類型的影片也會吸引許多觀眾的眼球。我們認為觀眾既然會對他人的反應感興趣，那對自己的情緒反應也會有一定程度的好奇吧？因此，我們希望可以做出一個能分析觀看者觀賞反應的系統，使觀看者能對觀賞中自身的情緒變化有進一步的了解。

原理分析與系統設計

我們的系統設計主要分成三部分，第一部分是影片的播放、錄製介面，第二部分是分析觀看者在看預告片過程中的情緒，第三部分是觀看者的情緒報告及觀影過程中的精采反應片段回顧，並在最後產生一張分析報表。

(一) 系統架構

1. 架構選擇

會選擇使用 PyQt 架構是希望我們的顯示介面可以用類似 app 的方式，使用者點選螢幕上的按鈕，進行系統操作並與增加互動性。由於 PyQt 有支援 signal 與 slot，因此我們可以設計出按鈕分別連接不同的指令，讓使用者達成互動體驗。播放影片的部分採用 ffmpeg，錄製影片的部分則採用 opencv。

實際操作流程如下，執行程式後，會先跳出一對話框，詢問使用者是否要播放預告片，使用者可以點選要播放-'Play Trailer'或不要播放-'Cancel'。當按下播放按鈕後，系統會自動播放電影預告片，同時錄製觀看者在看影片時的反應。若使用者選擇'Cancel'鈕，即不要播放影片，則顯示介面關閉，系統不會播放影片。選擇對話框右上角的'x'按鈕也同樣將顯示介面關閉，不播放影片。

當預告片播放完畢後，系統會按照當下的日期與時間建立資料夾，並把觀看者的影片存進該資料夾中。此時，介面會出現對話框，讓觀眾點選要播放何種情緒片段。選擇完後會播放觀眾該情緒的反應片段。

回放觀眾反應的部份我們也是利用 PyQt 這個架構去完成，並在按鈕的部分將文字改成相

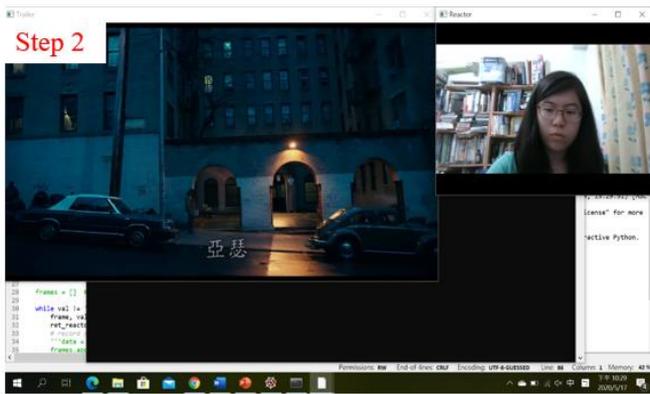
關情緒圖片。最後會輸出一張 pdf 檔的報表，報表的內容包含觀眾在觀看預告片這段期間的情緒反應的比例，以及最高情緒比例的一張代表照片。



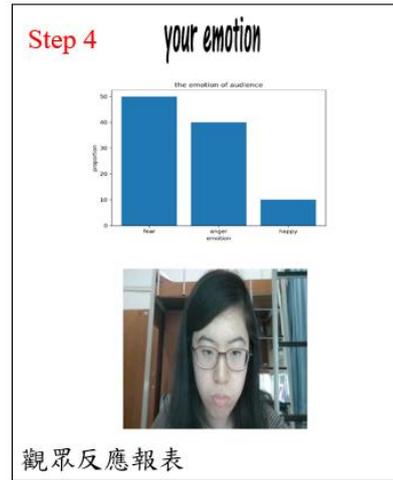
詢問使用者是否播放影片的對話框



情緒回放選擇按鈕



影片播放介面



觀眾反應報表

(二) 預測模型架構

1. 模型選擇

由於是建構系統本身，因此決定使用網路上現有，已預訓練過的模型再加以客製化成系統所需的模型。目前嘗試過的模型架構預訓練在 FER2013(Challenges in Representation Learning: Facial Expression Recognition Challenge)[2] 資料集上。

2. FER2013 資料集

Challenges in Representation Learning: Facial Expression Recognition Challenge

Learn facial expressions from an image
\$500 · 56 teams · 7 years ago

Overview Data Discussion Leaderboard Rules Team

Data Description

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

train.csv contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column.

The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples.

This dataset was prepared by [Pierre-Luc Carrier](#) and Aaron Courville, as part of an ongoing research project. They have graciously provided the workshop organizers with a preliminary version of their dataset to use for this contest.

An error occurred: Data not found

FER2013 資料集由 Training Set (28,709 張圖片) 加上 Test Set (3,589 張圖片) 圖片所構成，每一張都為 48 x 48 pixels 的灰階圖片，且 Training Set 中每一張都有對應情緒數字標籤，由 0 到 6 (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral)，此 Dataset 的 Training 目標是能分類出圖片的情緒。

3. Emopy[3] 模型

(1) 環境建構(主要套件說明)：

名稱	版本
python	3.6
keras	2.2.4
tensorflow	1.13.1

(2) 模型架構：

首先由兩層 Convolution，Extract 出 High level 的臉部特徵，再來接上 Max pooling Layer，讓 Model 更重視所辨識的臉部特徵，忽略掉臉部在圖片中的相對位置，重複此架構數層，嘗試取出臉部圖像中代表情緒的特徵。

(3) 模型應用：

由於此模型是針對單圖片做出情緒種類的可能性判斷，因此，為了能實際運用到此系統上，我們將上述系統所拍攝的影片，運用 OpenCV 做出 sampling，每秒(每0.5秒)取樣一次，便可得到許多圖像供判斷。

在 sampling 後，對每一張圖片都跑一次模型的預測，在此情境下，我們預想能被預測出最多次的情緒，便能代表觀影者在整部影片的主要情緒，最後在輸出時，輸出在此主情緒下預測最高的情緒圖片，作為代表觀影者本次觀賞的代表圖。

```
Initializing FER model parameters for target emotions: ['calm', 'anger', 'happiness']
2018-05-02 09:57:12.144168: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
Predicting on happy image...
anger: 44.1%
calm: 9.5%
happiness: 46.4%
Dominant emotion: happiness

Predicting on disgust image...
anger: 29.9%
calm: 16.0%
happiness: 54.1%
Dominant emotion: happiness

Predicting on anger image...
anger: 63.1%
calm: 22.9%
happiness: 14.0%
Dominant emotion: anger
```

(註)：對每張圖片都跑一次模型預測，紀錄數值，找出影片的主要情緒，再輸出比例最高的圖片。

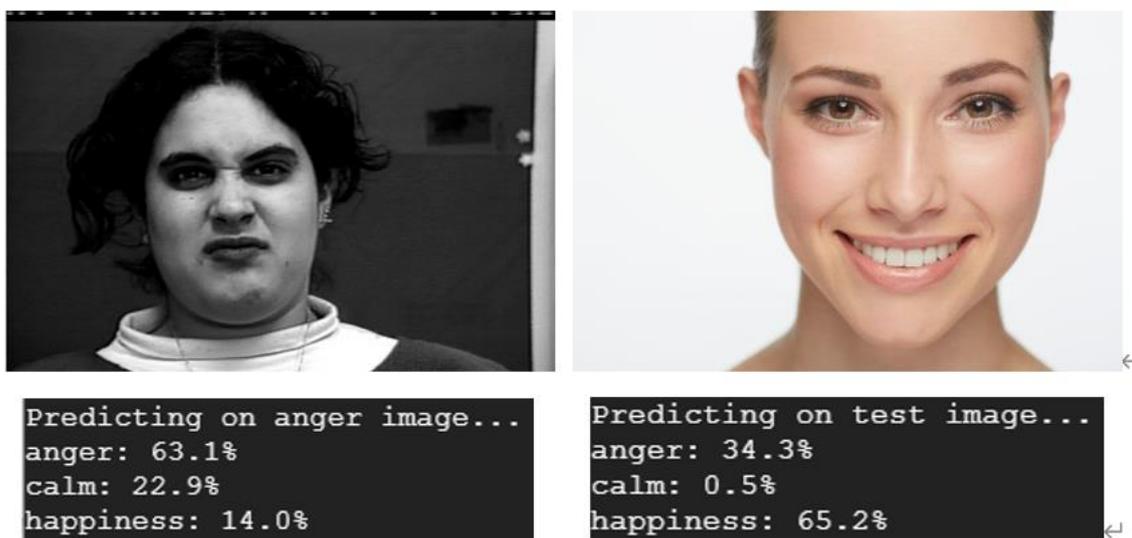
實驗結果

(一) 系統架構

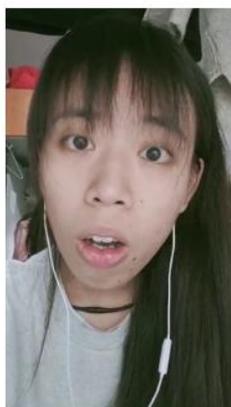
在 PyQt 架構下，我們實現了同步播放預告片與錄製觀看者反應的介面，以及觀眾反應回放功能，因此目前的系統是採用這個架構。

(二) 訓練模型

在最初的測試中，我們使用網路上蒐集的到的圖片，以作為分析。預測的結果相當不錯，單一情緒的辨識度都有達到六成以上。



再來我們錄自己的表情，做出特定情緒下的反應，並找出影片片段中所代表的情緒，下圖擷取自以 surprise 為情緒錄製成的影片。在此測試中，對於 surprise 情緒的分析是正確的，但對於 calm 情緒的分析則是失準的。



預測結果: surprise



預測結果: calm

結論

(一) 系統架構

能達成同步播放預告片與錄製反應影片的方式有許多種，愈複雜的方式不一定適用於所有情況，選擇較為熟悉的方式當遇到問題時比較容易找到解決辦法。本專題的影片播放、錄製介面最後採用 PyQt 架構搭配 ffmpeg 與 opencv 完成。

(二) 訓練模型

分析上述測試圖片，推測說模型分析失準的原因與所使用的預測圖片有很大的關係，在第一次測試的圖片與後期測試成功用的圖片來看，可看出背景部分在整張圖像中只佔很小的一部分。而預測失準的圖片，臉部只佔圖像的一半，推測是整張圖片中，臉部占比的比例需要大過一定程度，預測才會成功。

(三) 未來應用

藉由這個系統，可以讓電影製作者更加了解觀眾對該作品的情緒反應、觀眾的反應是否如預期，如：驚悚片是否真的讓觀眾感到恐懼。未來可提供影視相關從業者選材方面的參考。另外，也可以藉由此系統來預測未來票房表現。

參考文獻

- [1] Human Centered Multimedia, Social Signal Interpretation (SSI) Framework
<https://github.com/hcmlab/ssi>
- [2] Challenges in Representation Learning: Facial Expression Recognition Challenge
<https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>
- [3] A deep neural net toolkit for emotion analysis via Facial Expression Recognition (FER)
<https://github.com/thoughtworksarts/EmoPy>

心得感想

我們在尋找專題教授的時候，知道了李祈均教授的實驗室-人類行為訊息暨計算研究室。覺得這個實驗室能與人有更多的互動，也能透過實驗計算與分析許多人們的行為，是件非常有趣的事情，因此在聽完教授的介紹與討論過後，決定了 Movie: 觀眾情緒、電影內容結合介面系統建置這個專題。

在剛開始架設系統的時候，因為之前對 python 的接觸比較少，所以花了一點時間才慢慢上手，有時候看到不斷出現 error 的提醒訊息時，真的覺得非常頭痛。但是在成功做出一個功能的時候，例如成功播放預告片及同步錄影的時候，開心與成就感讓我們能繼續想辦法將其他的功能實現。然而我們在同步播放聲音及影像的部分遇到了頗大的困難，影像與聲音一直無法同步，使得我們耗費了一段時間在上面，卻找不到有效解決方法。只好尋求帶我們的學長協助，因此學習了另外一個架構-SSI。剛接觸 SSI 的時候覺得非常複雜，而且也並非我們

熟悉的 C 語言及 python，且光要使他成功運行就耗費了許多時間，也覺得很挫折。當能成功的播放預錄的影片及聲音的時候，我們覺得非常開心，以為我們快要成功完成同步播放及錄影的部分了，沒想到播放上網下載的預告片卻會出現 error，讓我們的心情一下子從天堂掉到谷底，並且在重複嘗試許多方法後，仍沒有找到有效的解決方法，且覺得這個架構實在有點複雜，許多裡面的 code 並不是真的每個都了解，因此我們決定還是繼續嘗試原本的架構，並嘗試找不同的套件來試。最後我們找到了 ffpyplayer 並解決了影片影像偏藍調的問題後，才成功的將影片與錄影同步。我們在解決這個問題後認為，或許有許多架構跟方法的功能是很強大的，但他背後的程式卻也會相對複雜，有時候選擇適當的工具比強大的工具對解決問題更有助益。

在訓練模型的時候，因為資料量龐大且環境建構複雜，所以也花了不少時間使他能成功運作，並在成功運作後，再找能將正確率提高的方法。剛訓練完的時候我們的模型對網路上的圖片是正確率蠻高的，然而在實際錄影我們自己的影片，並切成圖片後丟進去判斷時卻有許多的問題，例如明明是沒什麼情緒變化的表情，他卻判斷成害怕等等。讓我們花了一些時間探討原因，並在觀察比較了我們的圖片與 training data 的圖片後，發現背景佔過多的部分以及顏色的影像太大。所以我們將圖片灰階，並把圖片的背景切掉一些，來提供正確率。在探討為何正確率不高的過程中，我們也發現有時候很多面向的原因，都有可能導致最後的結果不同。

在這一年的專題中，教授並沒有硬性規定我們要什麼時候 meeting，什麼時候進度要到哪裡，只有在我們需要協助時提供幫助，給了我們很多的自由以及可以自己發揮的空間。而我們能自己規劃我們的系統有什麼功能，自己排定進度等，讓我們在完成一學年的專題外，也能有更多的討論及互動。