多媒體內容分析

Homework 2 Video Shot Change Detection

一、程式執行環境

Windows 11 企業版 21H2 build 22000.556

Python 3.9.7

Jupyter Lab 3.3.2 Package:

Opencv-python 4.5.5.62

Numpy 1.20.3

二、使用的visual features

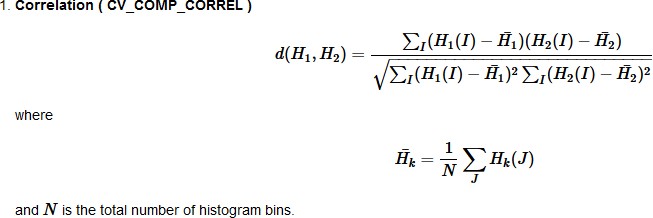
取每張frame 的灰階histogram

三、shot change detection 演算法

1. 將每張frame 分成 4 \* 4 的blocks，分別計算每個block 的 histogram。
2. 將每個frame 的各個部分的blocks 與前一個frame 的各個部分的blocks

的histogram 做比較，加總histogram 相似度小於 0.7 的blocks 數。

1. 相似度算法



(取自

https://docs.opencv.org/3.4/d8/dc8/tutorial\_histogram\_comparison.html)

1. 決定是否為shot change frame 有三種條件
   1. 相似度小於 0.7 的blocks 數>8 個
   2. 相似度小於 0.7 的blocks 數>6 個<=8 且前面連續 3 個frame 也是
   3. 相似度小於 0.7 的blocks 數>4 個<=6 且前面連續 6 個frame 也是

四、偵測效能（三部影片效能分別詳列）

1. news.mpg

|  |  |
| --- | --- |
| 執行結果 | 預期結果 |
| 73  195  235  301  370  452  861  1281 | 73  235  301  370  452  861  1281 |

* + 分析

因為此影片幾乎為hard cut，偵測不高，所以與預期結果大致符合。第 195 frame 似乎是影片突然變非常模糊的frame，在調整ˇ 參數之前的算法有時也會偵測到其他突然變非常模糊的frame

如：30、45、60 等。

1. climate.mp4

|  |  |
| --- | --- |
| 執行結果 | 預期結果 |
| 92  156  231  313  354  455~465  898~899  914  1020  1236  1400  1554 | 93  157  232  314  355  455~478  542~578  608~644  675~697  774~799  886~887  1021  1237  1401  1555  (frame 編號多了 1 號) |

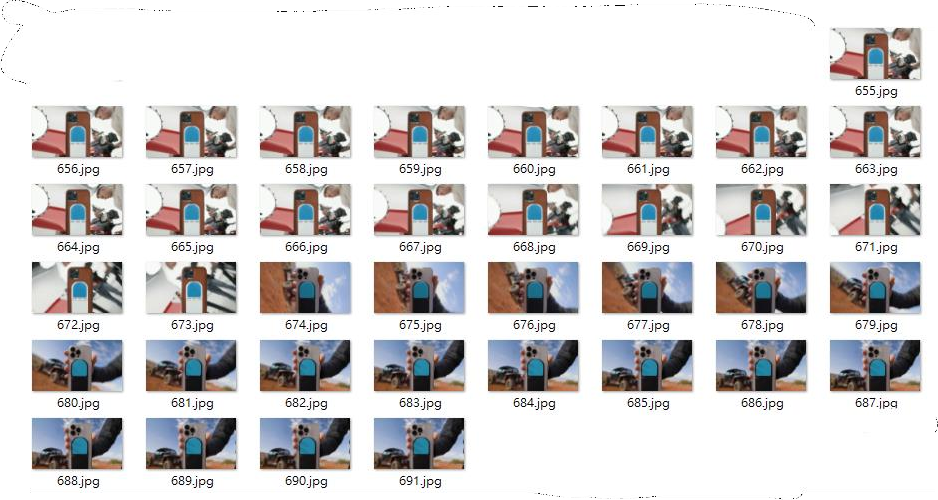
* + 分析

此影片有很多的淡入淡出的轉場動畫，蠻多過場後的素材灰階histogram長得差不多，造成此影片的偵測效能不好。Hard cut 的部分都有偵測出來，其他過渡轉場的部分則只抓出變化較劇烈的部分。可能需要分別對RGB的histogram作分析，才能取得比較好的結果。

1. ftfm.mp4

|  |  |
| --- | --- |
| 執行結果 | 預期結果 |
| 1~6  28  48  65  89  133  141~148  150~157  177  204~205  224  298~304  330  354  371  393  428  445~448  482  517  548  575  594~601  629  654  673  677  679~680  691  729  734 | 1~8  29  49  66  90  134  148~157  178  206  225  298~305  331  355  372  394  429  446~450  483  518  549  576  594~601  630  655  674  692  730  (frame 編號多了 1 號) |

* + 分析

此影片有很多複雜的轉場動畫，但是較前一部影片更為乾脆俐落點，轉場的時間長度都不長，所以都比較好偵測。除了hard cut，其他過渡轉場也大致都有偵測到，frame 的區間也大致正確。偵測不好的部分為

此片段可能是因為中間手機一直都在，擋住了一大部分的畫面， 導致算法有點誤認了有轉場的frame。