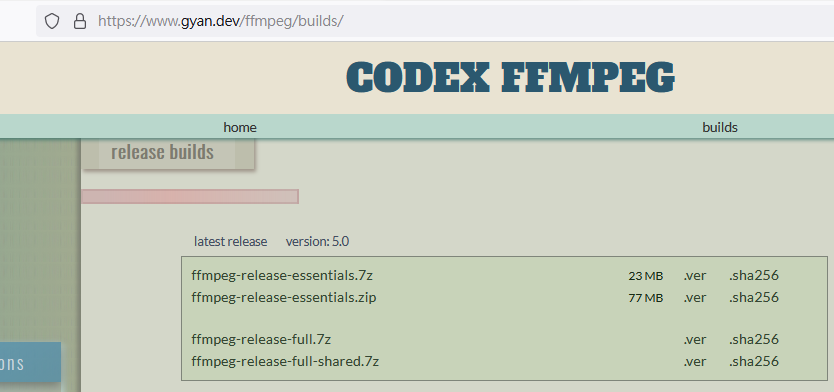
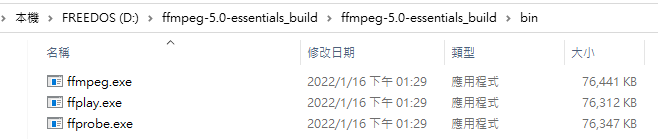
20220214在 win10 安裝並使用 ffmpeg 影音轉檔程式

1. 至[[WINDOW版下載處 https://www.gyan.dev/ffmpeg/builds/](https://www.gyan.dev/ffmpeg/builds/)]的release builds處下載[ffmpeg-release-essentials.zip](https://www.gyan.dev/ffmpeg/builds/ffmpeg-release-essentials.zip) 77 MB並解開至d:\



1. 將如下解開資料夾D:\ffmpeg-5.0-essentials\_build\ffmpeg-5.0-essentials\_build\bin 中的ffmpeg.exe 和ffplay.exe 拷至d:\



1. 備妥一個i.mp4影片檔

(可將<http://lfwiki.kmvs.km.edu.tw/lftree/uploads/output.mp4> 下載為d:\i.mp4)

1. 執行CMD(命令提示字元)並輸入如下ffmpeg指令，會出現本指令的版本、編譯組態與用法如下：

|  |
| --- |
|  |

1. 執行ffmpeg -i i.mp4 -c:v libx264 -c:a aac o.mp4

以影像編碼採用 H.264 、音訊編碼採用 AAC將 i.mp4 轉為 o.mp4，  
並 執行 ffplay o.MP4 來播放新產生的影片檔  
並再執行DIR \*.MP4 結果如下：  
註1: 進階音訊編碼（英語：Advanced Audio Coding，AAC）是有損音訊壓縮的專利數位音訊編碼標準，由Fraunhofer IIS、杜比實驗室、貝爾實驗室、Sony、Nokia等公司共同開發。

註2: H.264，也稱作MPEG-4 AVC（Advanced Video Codec，高級影像編碼），是一種視訊壓縮標準，同時也是一種被廣泛使用的高精度影像的錄製、壓縮和發佈格式，採基於運動補償的視訊編碼標準。

|  |
| --- |
| D:\>ffmpeg -i i.mp4 -c:v libx264 -c:a aac o.mp4  :  [D:\](file:///D:/)>ffplay o.MP4  :  D:\>DIR \*.MP4  :  2022/02/11 下午 02:02 6,757,887 i.mp4  2022/02/11 下午 02:17 15,354,465 o.mp4  2 個檔案 22,112,352 位元組 |

1. 同(5)增加如下-b:v 1M -b:a 256k 選項來調整bitRate(位元率)，

執行ffmpeg -i i.mp4 -c:v libx264 -c:a aac -b:v 1M -b:a 256k o.mp4

並 執行 ffplay o.MP4 來播放新產生的影片檔

並再執行DIR \*.MP4，結果如下(o.mp4檔案已降為 3.6 MB，可是畫質很差)：

|  |
| --- |
| D:\>ffmpeg -i i.mp4 -c:v libx264 -c:a aac -b:v 1M -b:a 256k o.mp4  :  [D:\](file:///D:/)>ffplay o.MP4  :  D:\>DIR \*.MP4  :  2022/02/11 下午 02:02 6,757,887 i.mp4  2022/02/11 下午 02:29 3,637,048 o.mp4  2 個檔案 10,394,935 位元組 |

1. 同(6)改如下-b:v 4M選項來調整bitRate(位元率)，

執行ffmpeg -i i.mp4 -c:v libx264 -c:a aac -b:v 4M -b:a 256k o.mp4

並 執行 ffplay o.MP4 來播放新產生的影片檔

並再執行DIR \*.MP4，結果如下(o.mp4檔案昇為 13MB，可是畫質不會變更好)：

|  |
| --- |
| D:\>ffmpeg -i i.mp4 -c:v libx264 -c:a aac -b:v 4M -b:a 256k o.mp4  :  [D:\](file:///D:/)>ffplay o.MP4  :  D:\>DIR \*.MP4  :  2022/02/11 下午 02:02 6,757,887 i.mp4  2022/02/11 下午 02:39 13,059,644 o.mp4  2 個檔案 19,817,531 位元組 |

(8)用指令ffmpeg -i I.mp4 -ss 00:00:00.0 -t 00:00:19.0 -c:v libx264 -c:a aac -b:v 3M -b:a 256k o2.mp4   
將影片輸入檔I.mp4在0秒之後以3M的bitrate截取計19秒長的影片內容為輸出檔o2.mp4

(9)用指令ffmpeg -i i.mp4 -f image2 -ss 00:00:17.0 -t 0.001 b.jpg

自給定的輸入影片的第17秒處開始0.001秒的連續播放畫面截取一個畫面為b.jpg

|  |
| --- |
|  |

(10)用指令ffmpeg -i I.mp4 -f gif -ss 00:00:17.0 -frames 10 a.gif

將給定的輸入影片的第17秒處開始的10個連續播放畫面截取為a.gif動畫檔

|  |
| --- |
|  |

(11) 用指令ffmpeg -i i.mp4 -vf crop=2/3\*in\_w:2/3\*in\_h -ss 00:00:00.0 -t 00:00:20.0 -c:v libx264 -c:a aac -b:v 3M -b:a 256k c1.mp4

將給定的輸入影片截切留下中間寬高各皆2/3的矩形區畫面成為c1.mp4影片檔。

(11b)用指令ffmpeg -i i.mp4 -vf crop=0.6\*iw:0.6\*ih:0.2\*iw:0.2\*ih c2.mp4

將給定的i.mp4輸入影片，以(0.2\*iw,0,2\*ih)為左上角截切出寬、高皆為原輸入影片0.6倍的矩形區畫面成為c2.mp4影片檔，iw及ih分別 為輸入影片的寬及高。

(11c)用指令ffplay -vf crop=0.6\*iw:0.6\*ih:0.2\*iw:0.2\*ih i.mp4 可預覽(11b)的輸出影片。

(12) 用如下指令ffmpeg -r 1/5 -i "treehrt%2d.jpg" -stream\_loop -1 -i eagle.mp3 -shortest -r 10 d.mp4

可將給定的22張照片(treehrt01.jpg~treehrt22.jpg)以每張圖5秒、並以輸出影格速率10fps及 eagle.mp3為循環音效來產生d.mp4影片檔(每張圖將依序持續50個影格在輸出影片上，輸出影片將以每秒10個影格來播放)

|  |
| --- |
| D:\treehrt>ffmpeg -r 1/5 -i "treehrt%2d.jpg" -stream\_loop -1 -i eagle.mp3 -shortest -r 10 d.mp4  :  Input #0, image2, from 'treehrt%2d.jpg':  Duration: 00:00:00.88, start: 0.000000, bitrate: N/A  Stream #0:0: Video: mjpeg (Baseline), yuvj420p(pc, bt470bg/unknown/unknown), 998x666 [SAR 96:96 DAR 499:333], 25 fps, 25 tbr, 25 tbn  [mp3 @ 000001789b40e540] Estimating duration from bitrate, this may be inaccurate  Input #1, mp3, from 'eagle.mp3':  Duration: 00:00:19.77, start: 0.000000, bitrate: 127 kb/s  Stream #1:0: Audio: mp3, 44100 Hz, stereo, fltp, 128 kb/s  File 'd.mp4' already exists. Overwrite? [y/N] y  Stream mapping:  Stream #0:0 -> #0:0 (mjpeg (native) -> h264 (libx264))  Stream #1:0 -> #0:1 (mp3 (mp3float) -> aac (native))  Press [q] to stop, [?] for help  [image2 @ 000001789b40b500] Thread message queue blocking; consider raising the thread\_queue\_size option (current value: 8)  [libx264 @ 000001789b40fec0] using SAR=1/1  [libx264 @ 000001789b40fec0] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2  [libx264 @ 000001789b40fec0] profile High, level 3.1, 4:2:0, 8-bit  :  Output #0, mp4, to 'd.mp4':  Metadata:  encoder : Lavf59.16.100  Stream #0:0: Video: h264 (avc1 / 0x31637661), yuvj420p(pc, bt470bg/unknown/unknown, progressive), 998x666 [SAR 96:96 DAR 499:333], q=2-31, 10 fps, 10240 tbn  Metadata:  encoder : Lavc59.18.100 libx264  Side data:  cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv\_delay: N/A  Stream #0:1: Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 128 kb/s  Metadata:  encoder : Lavc59.18.100 aac  :  [swscaler @ 000001789b6a90c0] [swscaler @ 000001789c1fe3c0] deprecated pixel format used, make sure you did set range correctly  More than 1000 frames duplicated  frame= 1099 fps=140 q=-1.0 Lsize= 8331kB time=00:01:49.90 bitrate= 621.0kbits/s dup=1077 drop=0 speed= 14x  video:6546kB audio:1736kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.593922%  [libx264 @ 000001789b40fec0] frame I:22 Avg QP:13.45 size:302833  :  [libx264 @ 000001789b40fec0] kb/s:487.87  [aac @ 000001789b411bc0] Qavg: 691.562  D:\treehrt> |

(13) 用選項 -an -filter:v "setpts=1/5\*PTS" 可將給定輸入影片加速5倍

如 ffmpeg -i i.mp4 -an **-filter:v "setpts=1/5\*PTS"** e.mp4

參考文章：

1. 常用ffmpeg指令用法參考文章：<https://lnpcd.blogspot.com/2012/09/ffmpeg.html>
2. Ffmpeg官網線上說明：<https://ffmpeg.org/ffmpeg.html>

