

# 聲音檔，影像檔，和 Video 檔的讀與寫 (by Python)

作者：蔡昌廷

丁建均老師編修

國立台灣大學電信工程學研究所

# 一. 聲音檔的讀寫

可以先安裝幾個模組

```
pip install numpy
```

```
pip install scipy
```

```
pip install matplotlib # plot
```

```
pip install pipwin
```

```
pipwin install simpleaudio # vocal files
```

```
pipwin install pyaudio
```

後面將說明使用 Python 讀檔，畫出頻譜，撥放聲音，製作音檔，錄音的方法

## I-A. 讀音訊檔

要先import 相關模組： `import wave`

讀取音檔：

```
wavfile = wave.open('C:/WINDOWS/Media/Alarm01.wav', 'rb')
```

獲得音檔取樣頻率和音訊長度：

```
fs = wavfile.getframerate()    # sampling frequency
```

```
num_frame = wavfile.getnframes() # length of the vocal signal
```

```
>>> fs
```

```
22050
```

```
>>> num_frame
```

```
122868
```

## 讀取波形與數據

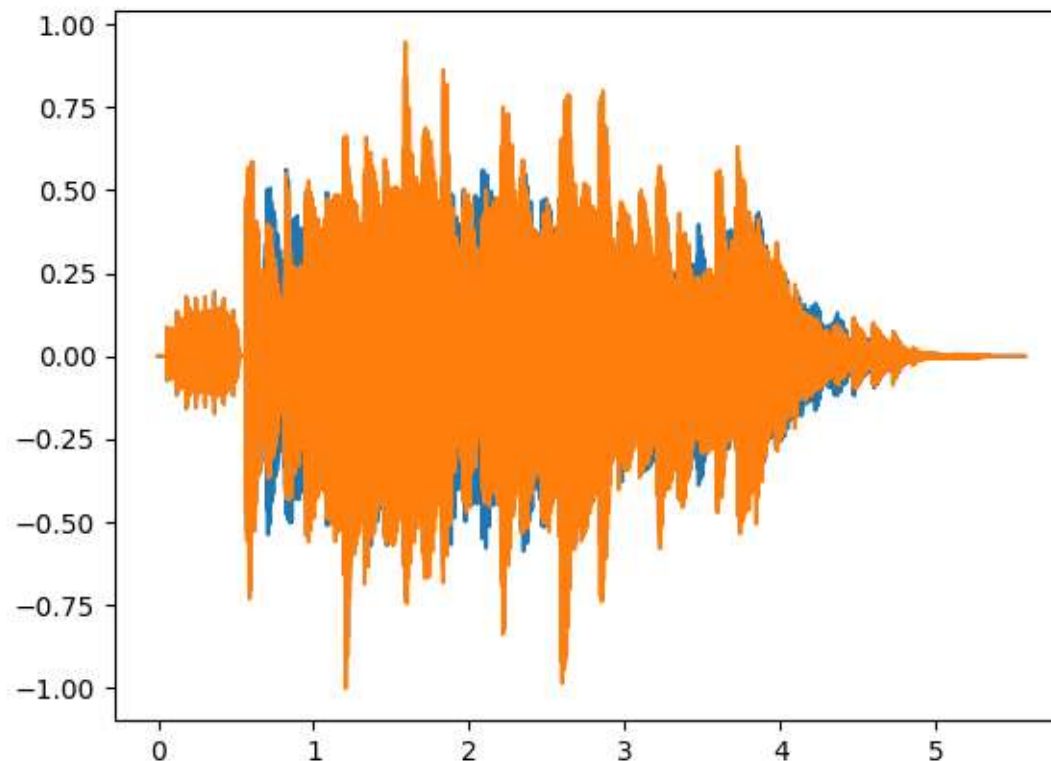
要先import 相關模組：import numpy as np

- `str_data = wavefile.readframes(num_frame)`
- `wave_data = np.frombuffer(str_data, dtype=np.int16)`  
# 轉成整數型態
- `wave_data = wave_data / max(abs(wave_data))` # normalization
- `n_channel = 2`
- `wave_data = np.reshape(wave_data, (num_frame, n_channel))`  
# 若為雙聲道音檔需要做 reshape

## 畫出音訊波形圖

要先import 相關模組：`import matplotlib.pyplot as plt`

- `time = np.arange(0, num_frame)*1/fs`
- `plt.plot(time, wave_data)`
- `plt.show()`

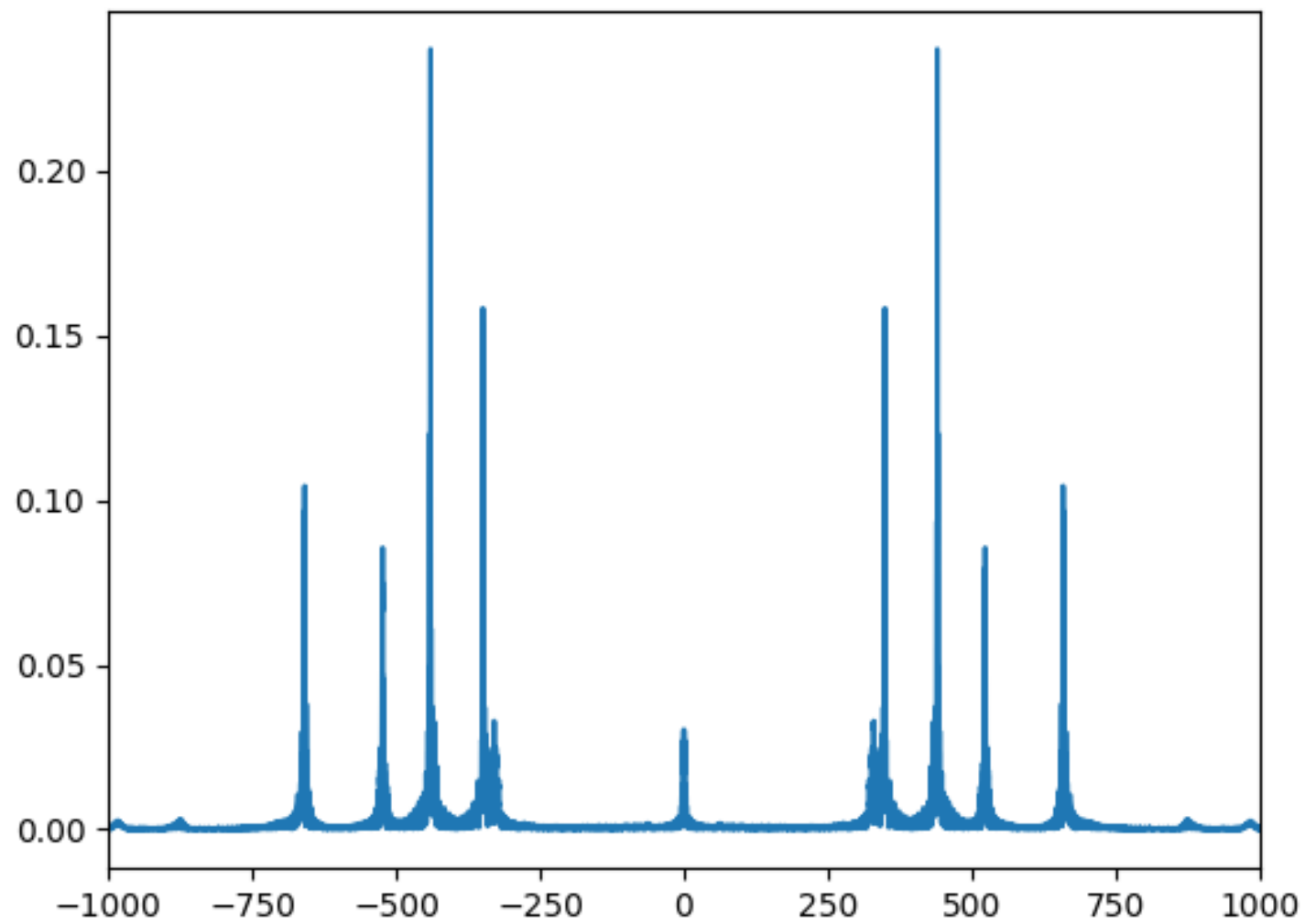


## 1-B. 畫出頻譜

要先import 相關模組： `from scipy.fftpack import fft`

- `fft_data = abs(fft(wave_data[:,1]))/fs` # only choose the 1<sup>st</sup> channel  
# 注意要乘上 1/fs
- `n0=int(np.ceil(num_frame/2))`
- `fft_data1=np.concatenate([fft_data[n0:num_frame],fft_data[0:n0]])`  
# 將頻譜後面一半移到前面
- `freq=np.concatenate([range(n0-num_frame,0),range(0,n0)])*fs/num_frame`  
# 頻率軸跟著調整
- `plt.plot(freq,fft_data1)`
- `plt.xlim(-1000,1000)` # 限制頻率的顯示範圍
- `plt.show()` # 如後圖

plt.show() 執行結果



## I-C. 播放聲音

要先import 相關模組： `import simpleaudio as sa`

- `n_bytes = 2` # using two bytes to record a data
- `wave_data = (2**15-1)* wave_data`  
# change the range to  $-2^{15} \sim 2^{15}$
- `wave_data = wave_data.astype(np.int16)`
- `play_obj = sa.play_buffer(wave_data, n_channel, n_bytes, fs)`
- `play_obj.wait_done()`



## 1-D. 製作音檔

- `f = wave.open('testing.wav', 'wb')`
- `f.setnchannels(2)` # 設定聲道數
- `f.setsampwidth(2)` # 每個 samples 有幾個位元組
- `f.setframerate(fs)` # 設定取樣頻率
- `f.writeframes(wave_data.tobytes())`
- `f.close()`

## 1-E. 錄音

要先import 相關模組： `import pyaudio`

範例程式

```
import pyaudio
pa=pyaudio.PyAudio()
fs = 44100
chunk = 1024
stream = pa.open(format=pyaudio.paInt16, channels=1, rate=fs,
input=True, frames_per_buffer=chunk)

vocal=[]
count=0
```

```
while count<200: #控制錄音時間
    audio = stream.read(chunk) #一次性錄音取樣位元組大小
    vocal.append(audio)
    count +=1

save_wave_file('testrecord.wav',vocal)
stream.close()
```

參考

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/491427/>

## 二. 影像檔的讀寫

可以先安裝幾個模組

```
pip install numpy
```

```
pip install matplotlib
```

後面將說明使用 Python 讀圖，畫圖，製作圖檔的方法

## II-A 讀取影像檔

```
import cv2
```

```
image = cv2.imread('D:/Pic/peppers.bmp')
```

注意

(1) 寫入圖片若為彩色圖片，需要注意cv2預設channel為BGR,

`image[:, :, 0] => B, image[:, :, 1] => G, image[:, :, 2] => R`

(2) 若讀檔讀不出來，有時要將路徑的 \ 改為 /

(3) `image.shape` 可以看出影像之大小

```
>>> image.shape
```

```
(512, 512, 3)
```

(4) 可讀 jpg, bmp, png 檔，但不能讀 gif 檔

## II-B 顯示影像

### Case 1: 圖的值格式為 int

以下的指令要配合使用 (彩色，灰階皆可)

```
cv2.imshow('test', image) # 以 test 為顯示的圖的名稱
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

若要顯示灰階圖，亦可用以下之指令

```
import matplotlib.pyplot as plt
```

```
plt.imshow(image)
```

```
plt.show()
```

若要用 `plt.imshow` 來顯示彩色圖，要先將 BGR 的順序轉回 RGB，將第二行改為

```
plt.imshow(image[:, :, [2, 1, 0]])
```

## Case 2: 圖的值格式為 double (非整數)

此時要先除以255，再將圖顯示出來

Example:

```
import cv2
image = cv2.imread('D:/Pic/peppers.bmp')
Image1 = image*0.5 + 127.5 # lighten the image
cv2.imshow('test', image) # int 不用除255
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imshow('test', image1/255) # 非整數要除255
cv2.waitKey(0)
cv2.destroyAllWindows()
```



## II-C 寫入圖片檔

```
cv2.imwrite('D:/Pic/jpg', image)
```

注意

(1) 寫入圖片若為彩色圖片，需要注意

`image[:, :, 0] => B`, `image[:, :, 1] => G`, `image[:, :, 2] => R`

(2) 若用 `cv2.imwrite('D:\Pic\jpg', image)` 可能無法存檔，要將 `\` 改為 `/`

(3) 若是使用 `plt.imshow` 和 `plot.show()` 來顯示，可以用右下角的 “save the figure” 來存檔



### 三. Video 檔的讀寫

可以先安裝幾個模組

```
pip install numpy
```

```
pip install matplotlib
```

後面將說明使用 Python 讀 video，製作 video 的方法

## III-A 讀取並播放影片

範例程式(讀取並顯示)

```
import cv2
cap = cv2.VideoCapture('test.avi')
while cap.isOpened(): # 持續讀取 video
    ret, frame = cap.read() # frame 包含目前 frame 的資訊
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    cv2.imshow('frame', frame)
    cv2.waitKey(1) # 可以調整放映速度, 數字大則越慢
    if cv2.waitKey(1) == ord('q'): # 按下q鍵退出
        break
cap.release()
cv2.destroyAllWindows()
```

參考資料：<https://www.kancloud.cn/aollo/aolloopencv/260405>

## 範例程式(讀取並將每個 frames 一一存檔)

```
import cv2
import matplotlib.pyplot as plt
cap = cv2.VideoCapture('test.avi')
while cap.isOpened():
    ret, frame = cap.read()
    # if frame is read correctly ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    plt.imshow(frame)
    plt.show()    # 此時可按右下角“save the figure”來將每個 frame 存檔
cap.release()
cv2.destroyAllWindows()
```

## III-B 製作 Video 檔

範例程式(讀取一個 video 檔之後，將當中的內容上下相反後，存成另一個 video 檔)

```
import numpy as np
import cv2
cap = cv2.VideoCapture('test.mp4')

# Define the codec and create VideoWriter object
fourcc = cv2.VideoWriter_fourcc(*'XVID') # 指定編碼格式
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

out = cv2.VideoWriter('output1.mp4', fourcc, 20.0, (width,height))
# 參數依序為 output file name, 編碼格式, frame per second (fps), 寬與高
```

(續)

```
while(cap.isOpened()):  
    ret, frame = cap.read()  
    if ret==True:  
        frame = cv2.flip(frame,0) # 上下相反  
  
        # write the flipped frame  
        out.write(frame)  
        cv2.imshow('frame',frame)  
        if cv2.waitKey(1) == ord('q'):  
            break  
    else:  
        break  
cap.release()  
out.release()  
cv2.destroyAllWindows()
```

參考資料：<https://www.kancloud.cn/aollo/aolloopencv/260405>