

105061254 林士平 數位訊號處理實驗報告 Lab6

1. Abstract/Introduction

VAD(Voice Activity Detection)是語音辨識最根本的技術，任何的語音辨識一開始一定會有一個方塊(block)是在做 VAD。VAD 的功能是利用音檔在不同時間有不同能量的特性來去掉音訊中我們不關心的部分(例如：雜訊、過小的聲音)，如此一來才能針對我們真正關心的部分來做後續的語音處理。

VAD 在語音通信系統的應用列舉如下：在多媒體應用領域，VAD 允許系統同時進行音頻和數據應用；在通用移動通訊系統（UMTS）中，它可以控制降低平均比特率，並提高整體的語音編碼質量；數位移動廣播、DSVD 或語音儲存中，都需要一種不連續傳輸的語音編碼參數。這樣可以降低平均功耗，提升傳輸的平均比特率並提升晶片的存儲能力。(本段節自維基百科)

本次的實驗便要走一次 VAD 的流程，採用 Energy threshold 的方式，每次擷取一小段聲音，當這一小段聲音 $> \text{threshold}$ 便保留下來， $< \text{threshold}$ 則予以捨去。藉由此方式，希望能將嬰兒哭聲檔分割成若干段只有音訊而沒有雜訊的小音檔。

2. Goals of this lab

- (1) Do Energy-based VAD method on baby cry audio file.
- (2) Plot start-end lines in Time Domain.
- (3) Plot threshold lines in Energy Domain.
- (4) Write baby cry audio segments stream into audio file.

3. Method

範例程式檔中，需要調整的地方有三個，第一個是 frameSize 和 overLap，overLap 會影響到解析度的大小，frameSize – overLap 則會影響到 step 的大小，step 的大小和 time complexity 有關係。經過反覆的測試我決定選用 frameSize = 700，overLap = 500。

第二個是 threshold 的選用，利用做出來的 threshold line 圖來 try and error，最後我決定使用 $\text{threshold} = \text{np.sqrt}(\max(\text{Audio_Energy})) * 95$ 。

第三個是做完音檔切割後，決定是否要把切割後的片段寫出的部分，如果該片段長度過小，代表雖然該片段能量比 threshold 大但仍為雜訊，不應該寫出，最後我選 $\text{len}(\text{item}) > 10$ ，以長度 10 為基準。

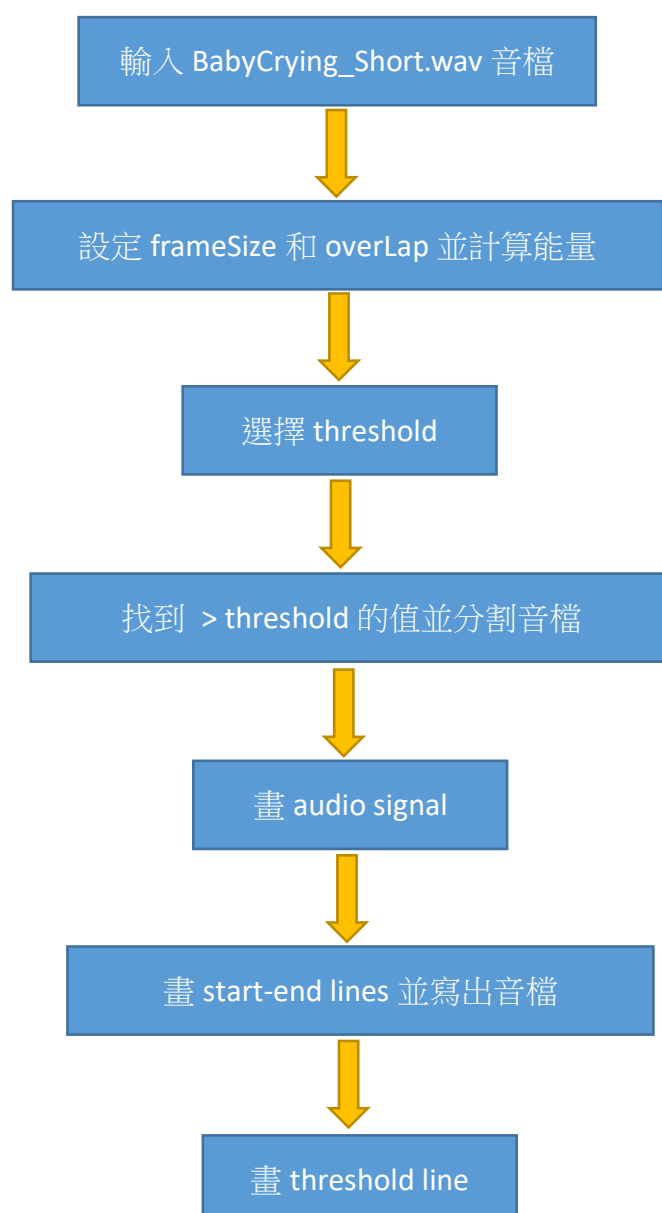
4. Pseudo code

以下是本次 Lab 程式碼重點：

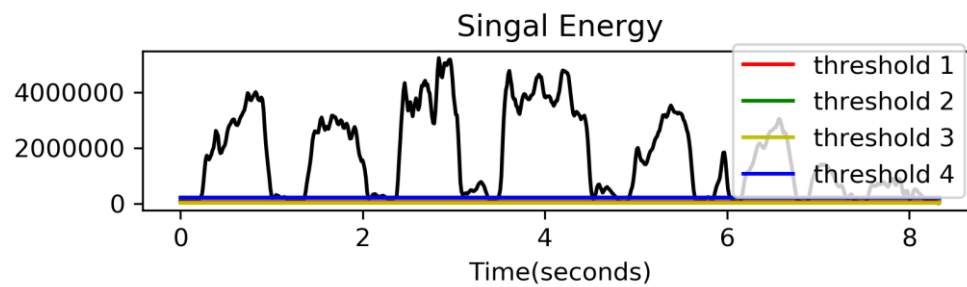
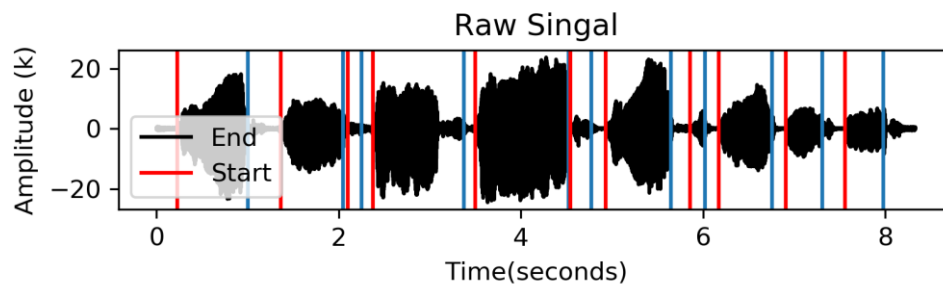
<p>(0) 定義 CalEnergy 類別</p>	<pre>def CalEnergy(audioData, frameSize, overLap): """Signal Array length""" wlen = len(audioData) """Calculate Step Size""" step = frameSize - overLap frameNum = int(math.ceil(wlen*1.0/step)) Energy = np.zeros((frameNum,1)) for i in range(frameNum): curFrame = audioData[np.arange(i*step,min(i*step+frameSize,wlen))] curFrame = curFrame - np.median(curFrame) # zero-justified """To calculate Energy""" Energy[i] = np.sum(np.sqrt(curFrame**2)) return Energy</pre> <p>說明：計算能量，使用方均根和代替絕對值和。</p>
<p>(1) 輸入 BabyCrying_Short.wav 音檔</p>	<pre>SampleRate, audioData= wav.read(InputAudioFile,'r')</pre>
<p>(2) 設定 frameSize 和 overLap 並計算能量</p>	<pre>"""Setting frame size and overlab Size.""" frameSize = 700 overLap = 500 Audio_Energy = CalEnergy(audioData,frameSize,overLap)</pre>
<p>(3) 選擇 threshold</p>	<pre>threshold4 = np.sqrt(max(Audio_Energy))*95 """Choose one threshold""" threshold=threshold4</pre> <p>說明：在此我選擇 threshold4 來作為 Energy threshold，經過反覆測試後選擇*95。</p>
<p>(4) 找到 > threshold 的值並分割音檔</p>	<pre>Filter_Array = np.where(Audio_Energy>threshold)[0] Split_Array = consecutive(Filter_Array)</pre>

(5) 畫圖並輸出分割後的音檔	<pre> for item in Split_Array: if len(item) > 10: start_line=item[0]*(len(audioData)*1.0/len(Audio_Energy)/SampleRate) plt.axvline(x=start_line,c='r') end_line=item[-1]*(len(audioData)*1.0/len(Audio_Energy)/SampleRate) plt.axvline(x=end_line) i=i+1 </pre> <p>說明：本段是畫 start end line 的程式碼，此部分重點是要作單位的換算，找出該片段的 start_line 和 end_line 實際對應到的時間位置，並且注意一開始 <i>if len(item) > 10</i>: 是為了避免 item 的長度太小，長度太小基本上就是雜訊。</p>
-----------------	--

5. Flow chart



6. Results



7. Reference

(1) 語音活性檢定(維基百科)

<https://zh.wikipedia.org/wiki/%E8%AF%AD%E9%9F%B3%E6%B4%BB%E6%80%A7%E6%A3%80%E6%B5%8B>