

103062315_report

Q1、

Discuss how you determine filter:

在呼叫時會輸入所要使用的 window function，這題用的是 Blackmann，並且輸入選擇的 filter，有三種可以選，分別是 Low-pass、High-pass、Band-pass，其中 Band-pass 需要特別輸入範圍，比較後找出適當的 fcutoff 跟 filter。

```
[outputSignal1,outputFilter1] = my_filter(y_input, fs, 601, 'Blackmann', 'Low-pass', 250);
```

How you implement the filter and convolutions to separate the mixed song:

Filter 的部分參照投影片第 76 頁，分別算出 Low-pass、High-pass、Band-pass filter，由於 N 為奇數，所以要取 round，並在 for 迴圈中上界跟下界分別取 ceil 跟 floor(負取 ceil，正取 floor)，code 如下

```
if strcmp(filterName,'Low-pass') == 1
    for n=ceil(-N/2):floor(N/2)
        if(n==0)outputFilter(middle)=1;
        else    outputFilter(n+middle)=sin((wc*n))/(pi*n);
        end
        outputFilter(middle)=2*fcutoff;
    end
elseif strcmp(filterName,'High-pass') == 1
    for n=ceil(-N/2):floor(N/2)
        if(n==0)outputFilter(middle)=1;
        else    outputFilter(n+middle)=-sin((wc*n))/(pi*n);
        end
        outputFilter(middle)=1-2*fcutoff;
    end
elseif strcmp(filterName,'Band-pass') == 1
    for n=ceil(-N/2):floor(N/2)
        if(n==0)
            outputFilter(middle)=1;
        else    outputFilter(n+middle)=sin(2*pi*fcutoff(2)*n)-sin(2*pi*fcutoff(1)*n)/(pi*n);
        end
        outputFilter(middle)=2*(fcutoff(2)-fcutoff(1));
    end
end
```

並且參照投影片 79 頁算出 window function，由於這裡取值是從 1-N，所以講義上的公式要將加+號改成減號，而講義的方法適用於取值從 -N/2-N/2，這只是公式的週期性所導致。

```

elseif strcmp(windowName, 'Blackmann') == 1
    for n=1:N
        outputFilter(n)=outputFilter(n)*(0.42-0.5*cos((2*pi*n)/(N-1))+0.08*cos((4*pi*n)/(N-1)));
    end
end
end

```

最後去做 convolution，對於小於 0 的則不處理。

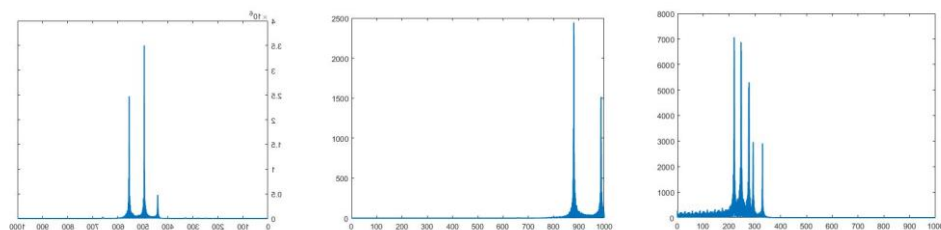
```

for n=1:length(inputSignal)
    for k=1:N
        a=n-k;
        if(a<1)
            h=0;
        else h=inputSignal(a);
        end
        outputSignal(n)=outputSignal(n)+outputFilter(k)*h;
    end
end
end

```

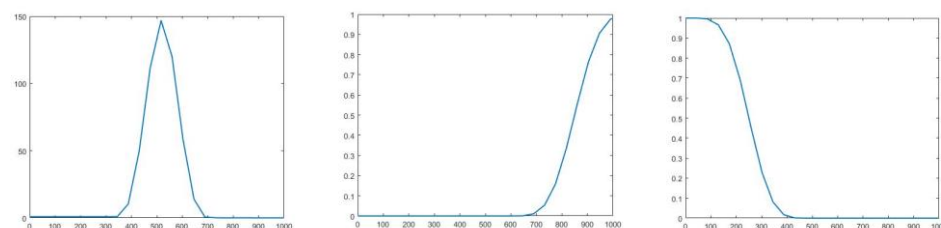
Compare spectrum and shape of the filter:

1、 Filter Signal Spectrum:



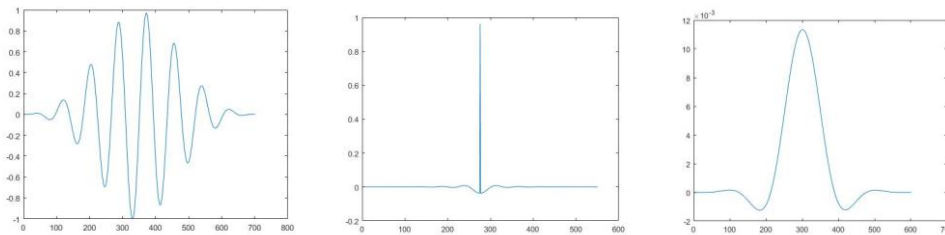
從左至右分別是 Bandpass、Highpass、Lowpass 後的 signal spectrum，我們可以看到 bandpass 左右兩邊都被濾掉了，留下中間界於 fcutoff 的訊號；至於 Highpass 我們能發現位於右邊的高頻的訊號被留了下來，濾掉了低頻的訊號，反之 Lowpass 則相反，去高頻留低頻。

2、 Filter Spectrum(Frequency Domain):



從左至右分別是 Bandpass、Highpass、Lowpass 的 filter 的 spectrum，bandpass 呈現左右對稱的山峰型，而 highpass、lowpass 則成了兩個相對的坡型，Highpass 的圖形中低頻的值為 0，僅留下高頻，Lowpass 則相反。

3、 Filter (Time Domain):

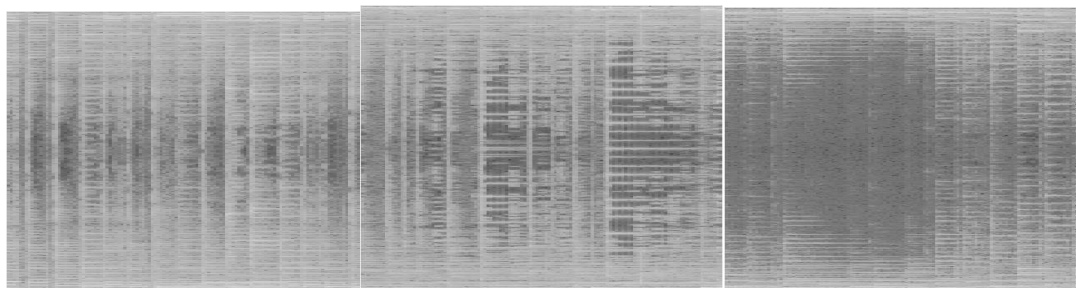


從左至右分別是 Bandpass、Highpass、Lowpass 的 filter 在 time domain 的圖，觀察可發現 Highpass 的圖跟 Lowpass 基本上是相反的，這次因為在 highpass 跟 lowpass 的公式上是相差一個負號的，並且由於是單一的 sin 函數，左右兩端的值會漸小，因此跟 bandpass 比起來 highpass 跟 lowpass 會呈現有一個區間特別突出，其餘為 0 的現象，而 bandpass 則因為是兩 sin 函數組成，所以波型變化較大。

Q2、

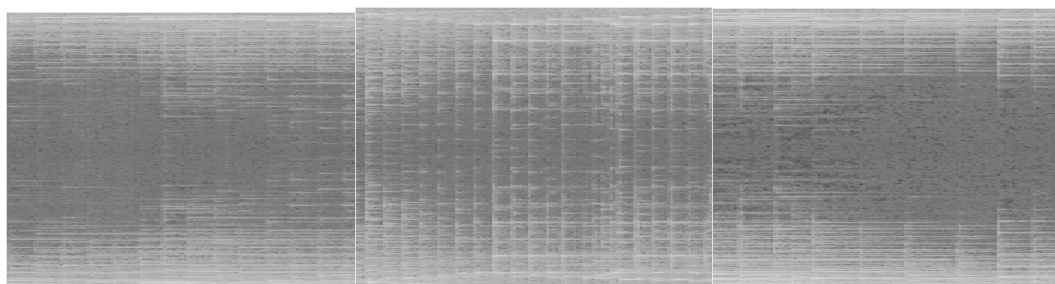
Instrument:

Guitar:



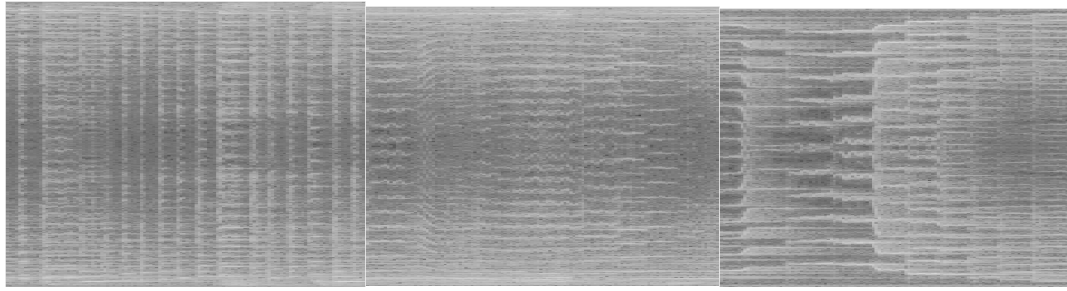
跟鋼琴都有呈現格子狀，然而 Guitar 的長條狀較為密，但每個長條的分節較為粗，整體成垂直狀的長條格子分布，格子狀明顯。

Piano:



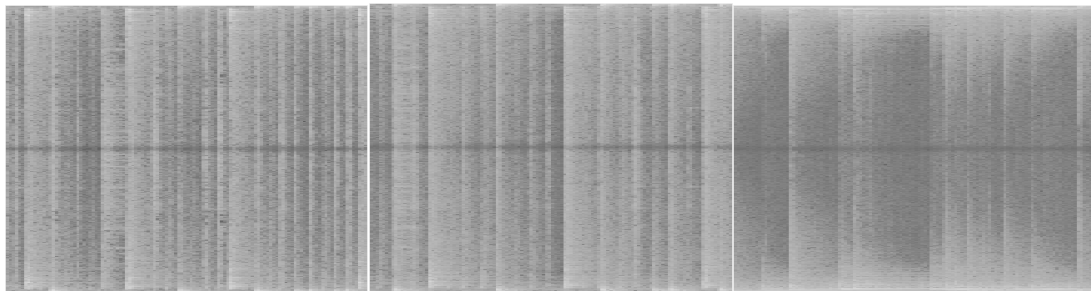
一樣具有格子狀，但跟吉他相比下長條分布較為稀疏，但每條的分節更為細，推測可能因為音色的音域的關係，鋼琴擁有較多的音調，所以比較水平的部分較吉他來得細，然而和弦部分可能是表現在垂直上，所以吉他跟小提琴的垂直部分普遍較細。

Violin:



整體看起來呈現水平走向，有明顯的波浪狀，跟吉他的圖有點相似，或許是因為兩者皆為弦樂器所致。

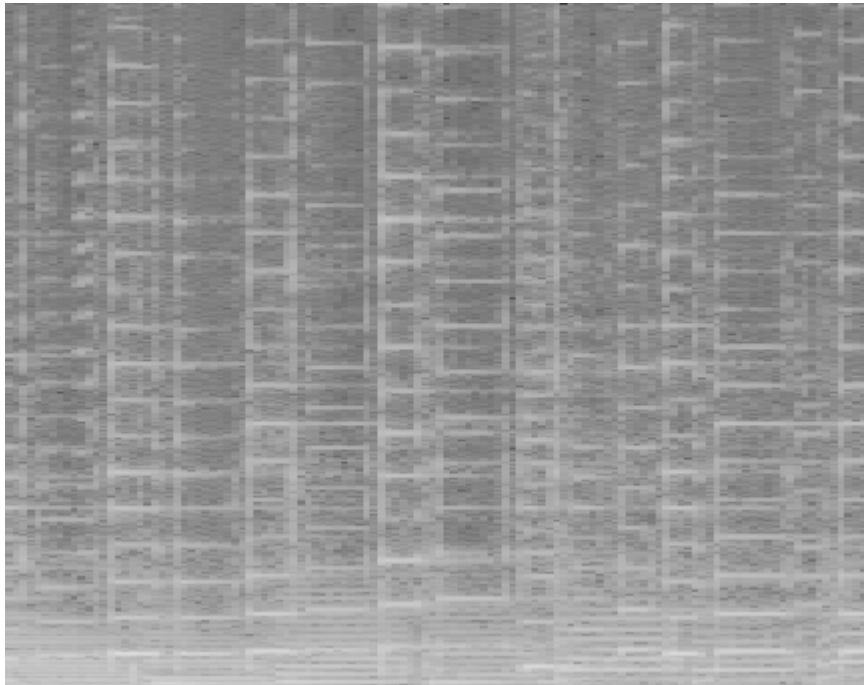
Drum:



呈現垂直走向，跟上面三種比起來較為單一，猜測這可能是反應在音域上，鼓相較之下，跟前三者比起來音色較為單一，所以在圖形上的變化也比前三者少。

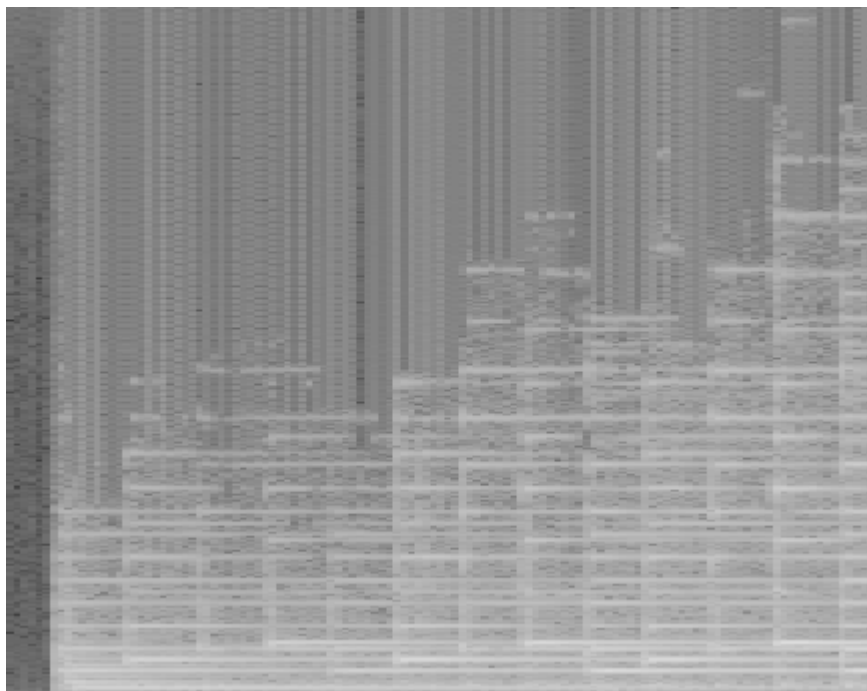
Test spectrogram classify

Test Spectrogram1:



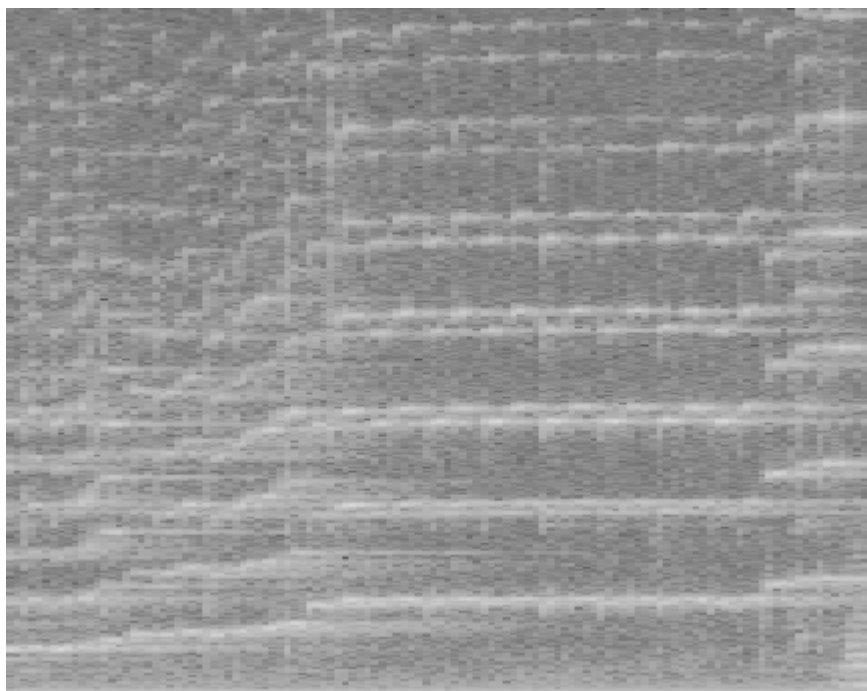
細垂直，寬水平，格子狀明顯可能為吉他。

Test Spectrogram 2:



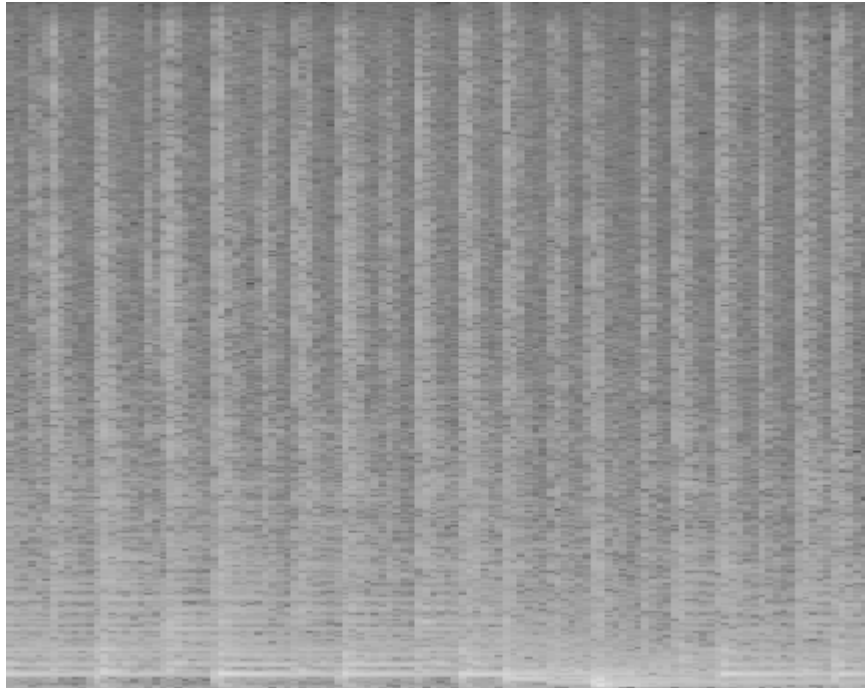
細水平，可能為鋼琴。

Test Spectrogram 3:



水平走向，波浪狀應該是小提琴。

Test Spectrogram 4:



垂直單一，應該是鼓。

Short-time Fourier transform:

一開始先選擇 window function，這裡我用的是 Hanning，code 如下

```
]for n=1:segment_duration
    w(n)=0.5-0.5*cos((2*pi*n)/segment_duration);
-end
```

segment_duration 就是 window 的大小，也是每個區間的大小，因此我們要算一個的範圍就會是區間大小減去彼此重複的部分，直到 input_signal 的 size-segment_duration，對於每個區間採"."的方式來套用 window function，之後做 fft，再將它聚集到一個 col=segment_duration 二維矩陣 S，透過 row 去紀錄有幾個區間，這裡算出來是 120 個。

```
]for k=1 : segment_duration-segment_overlap : N-segment_duration

    s=x(k:k+segment_duration-1);

    t=s.*w;
    sf=fft(t);
    S(:,row)=sf;
    T(:,row)=(k+k+segment_duration-1)/2/samplerate;
    row=row+1;
-end
```

F 是週期性頻率，參考以下網址得知 F 跟 T。

<https://www.mathworks.com/matlabcentral/fileexchange/45197-short-time-fourier->

[transformation--stft--with-matlab-implementation](#)

傅立葉轉換是在假設頻率不會隨時間而變的一種呈現訊號自身能量密度的函數，然後事實上像是音樂訊號和語音訊號這種頻率會隨時間改變的函數無法單靠傅立葉轉換得到時間資訊，所以才需要用 **STFT** 的方式來獲取時間資訊，其中 **T** 為時間，透過 **T** 跟 **F** 作時頻分析，在這種頻率隨時間改變的訊號上是有利的。

```
-F=(0:row-1)*samplerate/segment_duration;
```