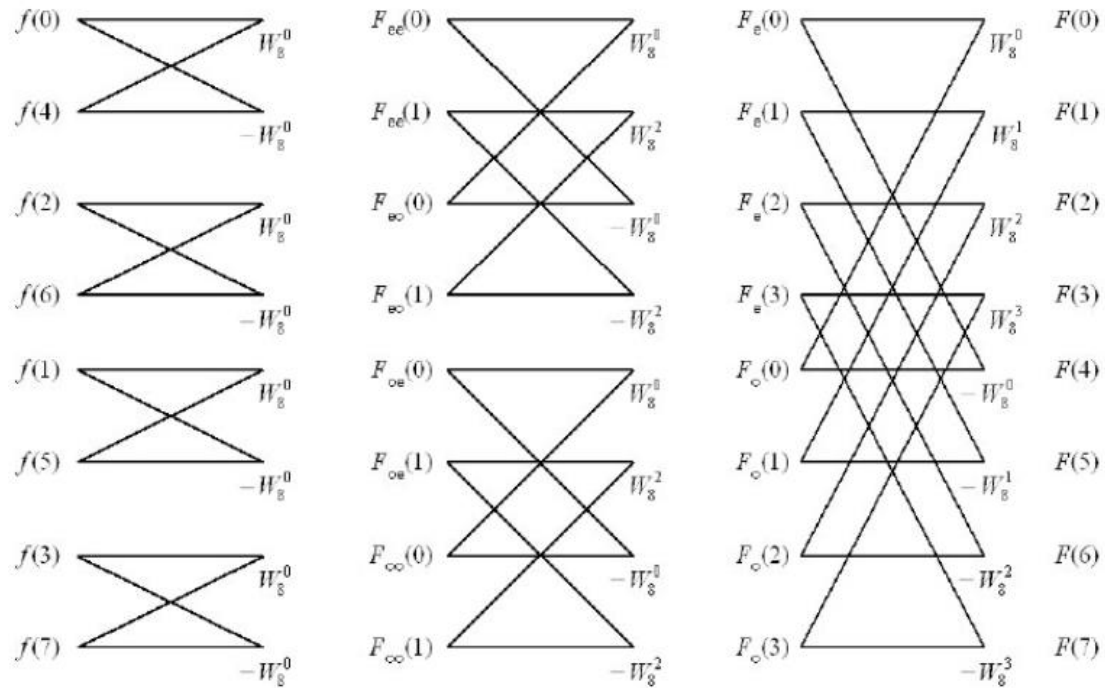


# 影像處理 Lab3 Report

姓名: 王領崧 學號: 107062107

## 1) Proj.04-01 Two-Dimensional Fast Fourier Transform

FFT 實作方法:



以上面這張圖( $n=8$ )為例，可以知道從左到右總共有三層計算，在每一層中會分成數個區塊(第一層 4 個、第二層 2 個....)，然後每個區塊的 elements 會再分成兩半，兩側相同位置的數字為一組進行計算，產生出同樣位置的結果，如此一層一層的計算下去。

因此我們需要跑一個三層 for 迴圈。

首先宣告數個變數: layers(表示總共有幾層計算)、inst(代表進行的是 FFT 還是 IFFT)、time(代表該層有幾個區塊)、partition(代表一個區塊有幾個數字 初始值為 2)、pos\_neg(代表一個區塊內一組的數量=partition/2)、checkpoint(每一層的 input)、temp(每一層的 output)。

然後將 checkpoint 設定為經過 bit-reverse 後的  $f()$  值，將 partition 設為 2。

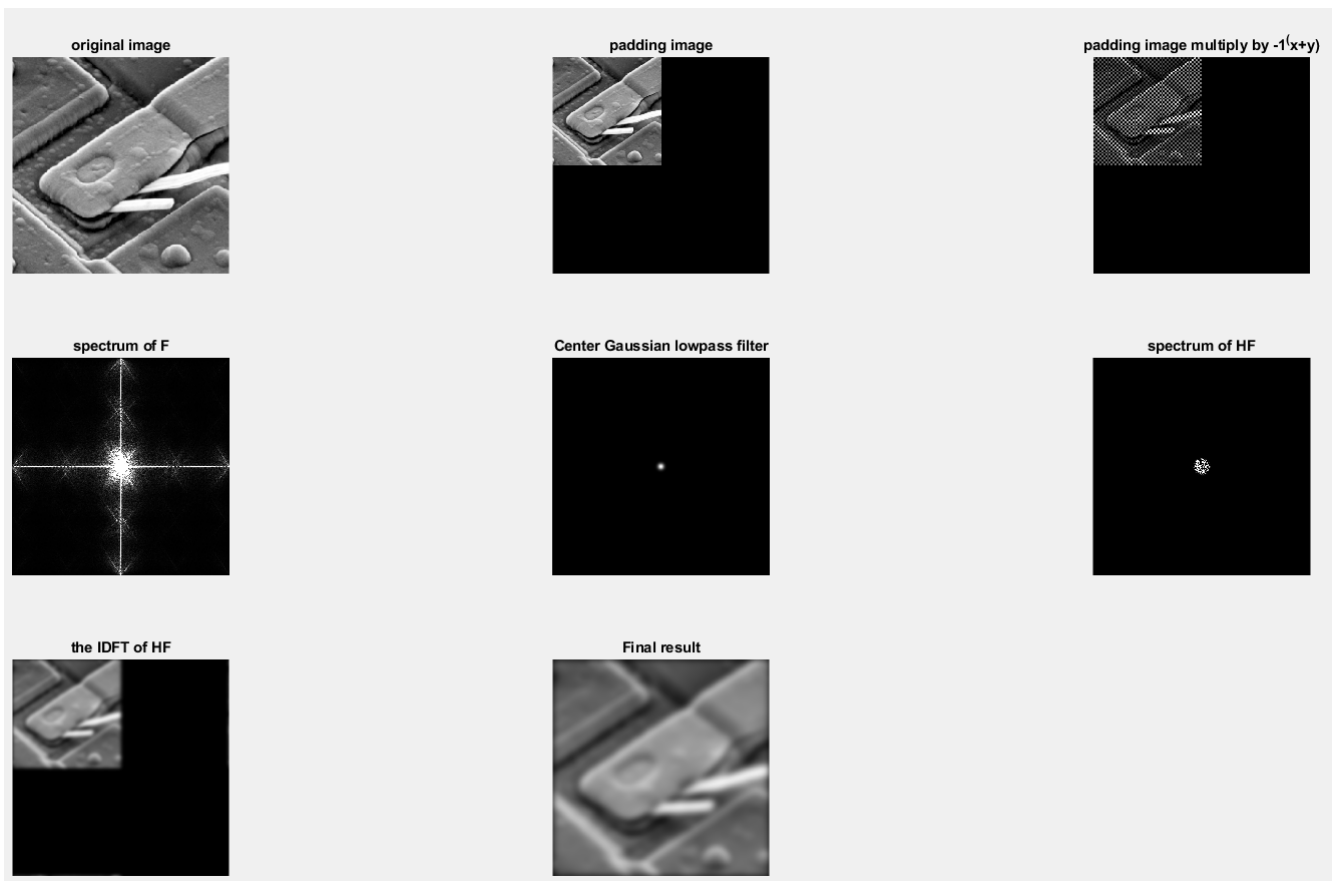
接著開始進行三層迴圈的執行，最外圈是 layers 的次數，而 layers 的值為  $\log_2(\text{input-size})$ ，表示總共有幾層要跑。接著在每一層中，也就是第二個迴圈就是 time 的次數，表示一層中有幾個區塊要進行計算，time 的值會等於  $\text{input-size}/\text{partition}$ 。最後最裡面的那層，為實際計算，第三個迴圈的部分，會是

pos\_neg 的次數(pos\_neg 為  $\text{partition}/2$ )，因為我們一次計算是取兩半中同樣位置的兩個數，因此 for loop 不用跑整個 partition，計算的部分就是取出兩個位置的值(利用 checkpoint)，使在上半部的 temp 等於上半部的 checkpoint+ 下半部的 checkpoint\*W 權重，下半部的 temp 則是改為-，就完成了一組數字的計算，然後利用迴圈跑過每一組，至每個區塊，最後到每一層就完成 FFT。在每一層計算完成後，要將 partition 加倍，並且要將 checkpoint 設為 temp，使得上一層的 output 為這一層的 input。

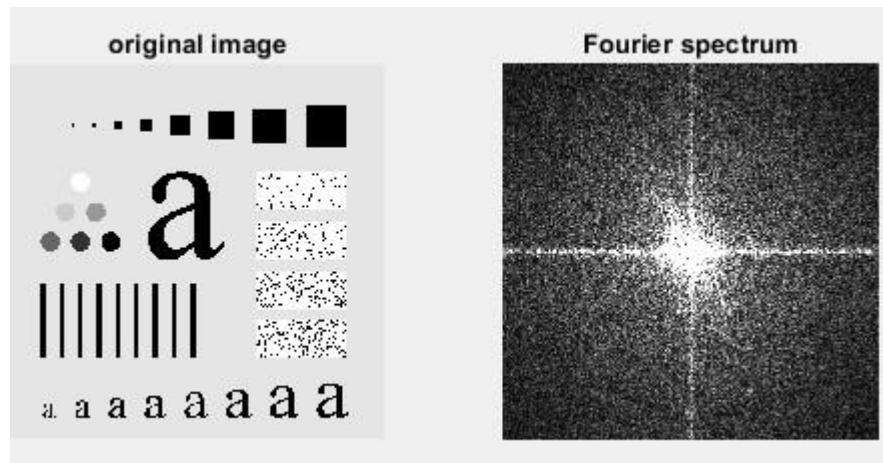
IFFT 與 FFT 基本上是一樣的，只需要將 W 權重中虛數  $-1j$  改成  $1j$ ，以及算出結果後要進行平均即可。

我利用 inst 這個讀進來的參數來分辨進行的是 FFT 還是 IFFT。

底下為 matlab code 結果圖:



## 2) Proj.04-02 – Fourier Spectrum and Average Value



上圖為原圖與其頻譜圖

```
>> project04_02
      {'image_mean: '}      {'0.367'}

      {'DFT_mean: '}      {'0.367'}
```

上圖為 mean 的比較

分析: 由原圖所得的 mean 會與 spectrum 中心點得到的 mean 是相同。因為單純從傅立葉轉換的公式來看時,可以發現  $F(0,0)$  所得到的 mean 會是原圖的 mean, 但是我們在傅立葉轉換前, 會先乘上  $-1^{(x+y)}$  做平移, 因此原本在 spectrum 的左上會被移到中心點來。

### 3) Proj.04-03 Lowpass Filtering



由左至右由上至下分別是原圖、cutoff frequency = 10 結果圖、cutoff frequency = 30 結果圖、cutoff frequency = 60 結果圖、cutoff frequency = 160 結果圖、cutoff frequency = 460 結果圖。

#### 4) Proj.04-04 Highpass Filtering



由左至右分別為原圖、cutoff frequency = 60 結果圖、cutoff frequency = 160 結果圖。