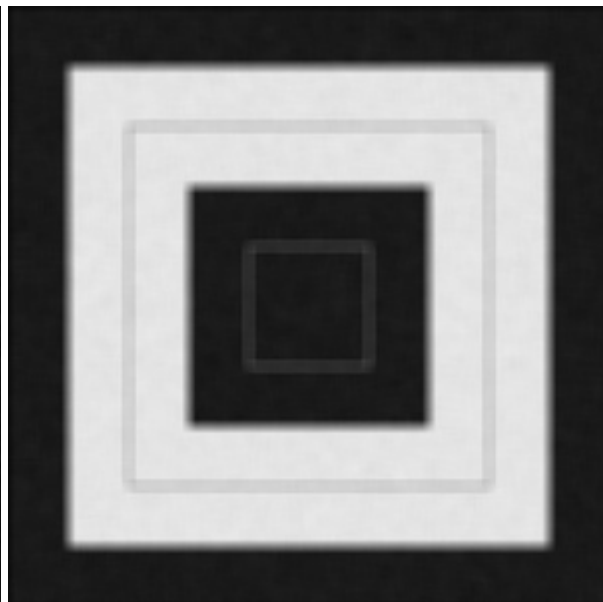
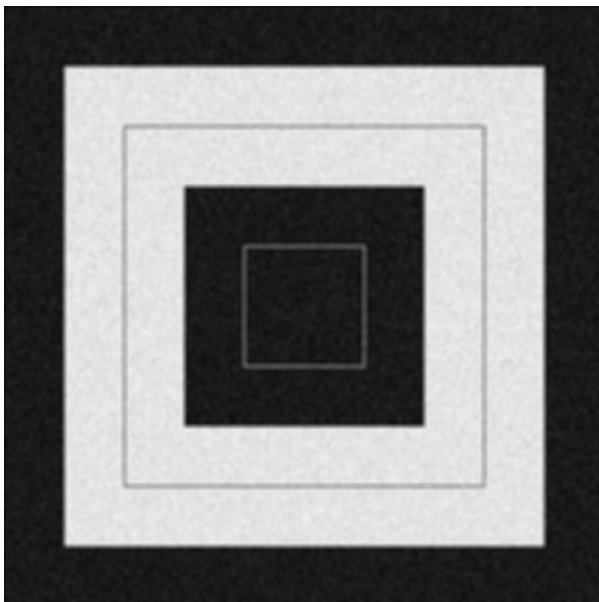


# Report 110590018 劉承翰

---

## Q1 MeanFilter

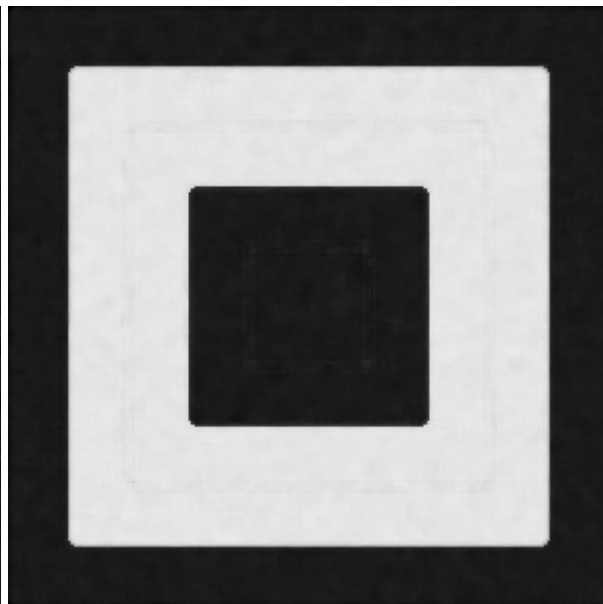
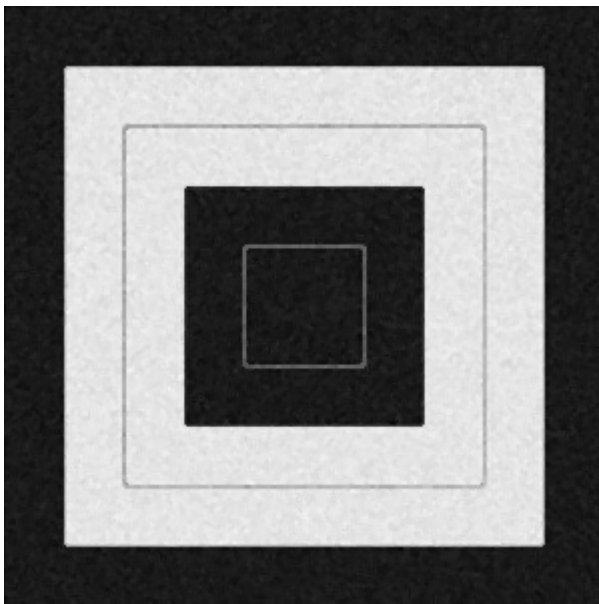




1. 透過`np.pad`來將圖片邊界擴展補0(`filtered_image`)
  2. 尋訪原始照片的pixel，並算出filter覆蓋值的平均，最後將平均值寫進`filterd_image`對照的座標
- 效果 具有降噪功能，但對於salt-and-pepper noise的效果很差(img2)

## Q2 MedianFilter



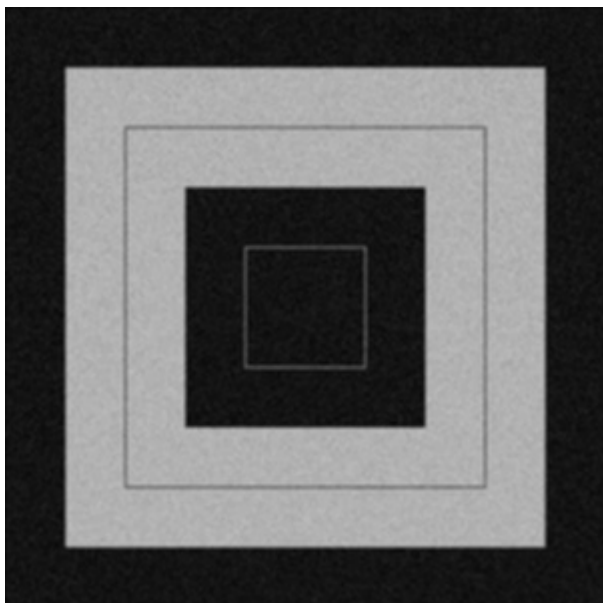


1.透過`np.pad`來將圖片邊界擴展補0(`filtered_image`) 2.尋訪原始照片的pixel，並重新排序被filter覆蓋區域的值，最後將中值謝進`filterd_image`對照的座標

- 效果

對於salt-and-pepper noise有著良好的降噪效果(img2)，保留更多的邊緣

### Q3 GaussianFilter



1. 透過`np.pad`來將圖片邊界擴展補0(`filtered_image`)

2. 透過公式 $G(x,y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}}$ 並使filter中心點為(0,0)，接著算出在filter中所對應的值，最後將算出來的結果全部相加。
3. 將相加的結果，寫進filtered\_image所對應的座標

- 效果

在去除噪音與保留細節平衡的比較好

## 心得

相較於上次作業，這次簡單很多:D，只要基礎的矩陣運算就好。但實作的時候遇到了一個小問題，我看照片內容以為本身就是grayscale，但實際上卻不是，讓我想說我照著公式去實作，為甚麼輸出的圖片都還是全黑的==，最後發現是輸入格式的問題:D。