

這次作業實作以不同型態的 filter 做灰階影像處理，並探討其中差異。

步驟如下：

對灰階圖像以要求 kernel 大小做 Mean/Median/Gaussian filter，並重複對前次結果再 filter 處理共 7 次，依序輸出圖像。

➤ Describe the main part of your method：

```
// 用 kernelSize*kernelSize 卷積回傳鄰近的像素點
vector<uchar> getConvolutionList(const Mat& source, int row, int col, int kernelSize)
```

以 source 圖像搜尋(row, col)像素點為中心 kernelSize^2 大小的視窗中所有的灰階像素值，如果視窗在圖像邊界，超出邊界的值設為 0（零填充方法）。

```
// 取得卷積範圍內的 Mean 值
uchar getMeanValue(vector<uchar> pixellist)
```

利用 getConvolutionList() 所得到的 pixellist 計算原像素經過 Mean filter（取平均值）後的像素值。

```
// 快速排序函数
void quickSort(vector<uchar>& nums, int left, int right)
```

以常見的快速排序方法排序 nums 的 vector，為 Median filter 步驟要求。

```
// 取得卷積範圍內排序後的 Median 值
uchar getMedianValue(vector<uchar> pixellist)
```

利用 getConvolutionList() 所得到的 pixellist，經過 quickSort 排序，計算原像素經過 Median filter（取中值）後的像素值。

```
// 依照給定的 kernel 大小和標準化參數給出 kernel 權重值
void setGaussianKernel(int kernelSize, double standardDeviation)
```

以 kernelSize^2 的視窗大小和 standardDeviation 參數設定 Gaussian Kernel。

```
// 取得特定方法過濾圖像
void getFilterImageByMode(Mat& source, Mat& dest, const string& mode, int kernelSize)
```

根據提供的原圖像 source 以 mode 方法的 filter 和視窗大小 kernelSize^2 輸出圖像至 dest，利用函式指標依照 mode 指向特定的 get{mode}Value() 以不同方法處理像素值，而 mode = gaussian 會需要先呼叫過 setGaussianKernel()。

➤ Bonus: Discusses the result after repeat 7 times

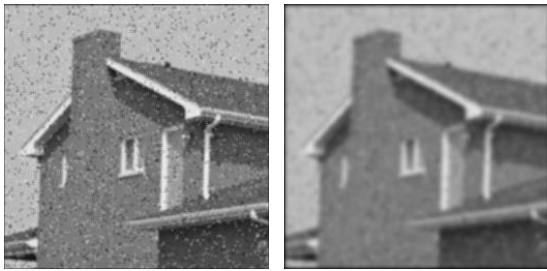
- Mean filter 3*3 能得到比較平均的變更值，在去雜訊中還是有些雜訊沒被濾除，這跟 kernel 太小有關。
- Mean filter 7*7 能得到比 Mean filter 3*3 更平均的變更值，且沒有明顯雜訊，但圖像會看起來更模糊，許多細節被濾除，這跟 kernel 太大有關。
- Median filter 3*3 能得到比較合理的變更值，在去雜訊有顯著的效果，灰階差距較大的地方能看到強烈的邊緣，kernel 並不會太大，是蠻理想的處理方式。
- Median filter 7*7 的效果與 Median filter 3*3 相似，去雜訊時圖像細節被嚴重忽略，較不理想的處理結果，這跟 kernel 太大有關。
- Gaussian filter 5*5，我使用的 $\sigma = 1.414$ ，是在逐漸調整下得出比上述 filter 效果還要好的參數值，特色是能盡量保留細節、圖像不會過度模糊，5*5 的 kernel 有較佳的去雜訊效果。

➤ Result images (total = 140 pics, document imports 40 pics, 1 times and 7 times)

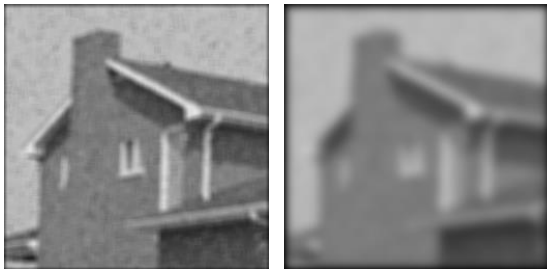
At next page

House256_noise (repeat 1, repeat 7)

- Mean 3*3



- Mean 7*7



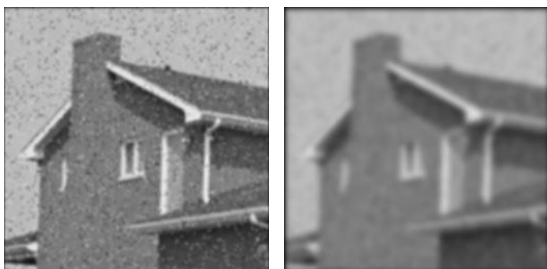
- Median 3*3



- Median 7*7



- Gaussian 5*5 $\sigma = 1.414$

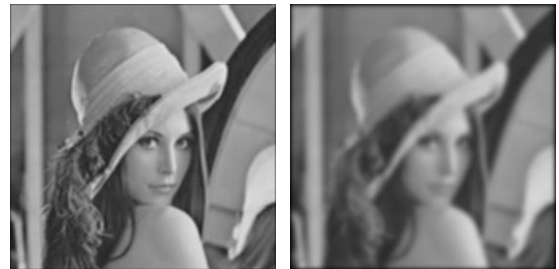


Lena_gray (repeat 1, repeat 7)

- Mean 3*3



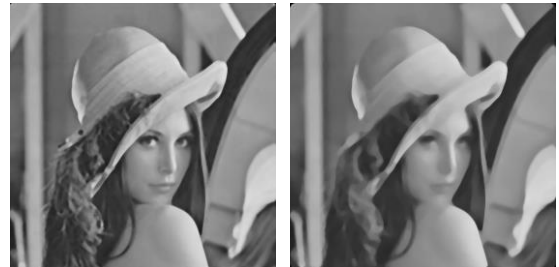
- Mean 7*7



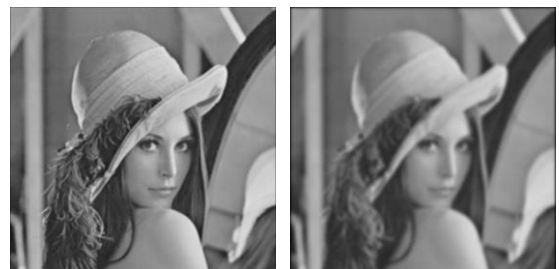
- Median 3*3



- Median 7*7

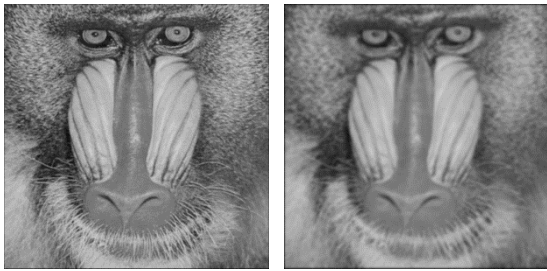


- Gaussian 5*5 $\sigma = 1.414$

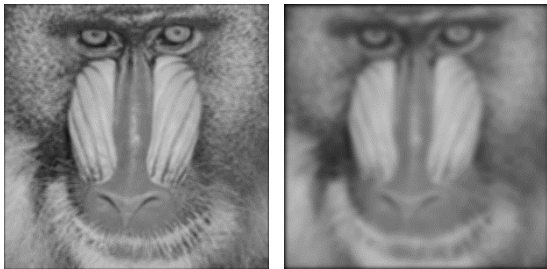


Mandrill_gray (repeat 1, repeat 7)

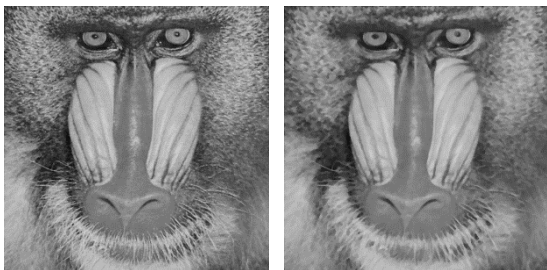
- Mean 3*3



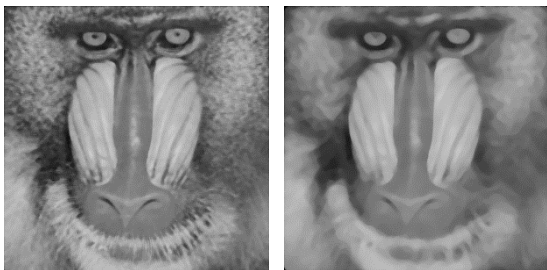
- Mean 7*7



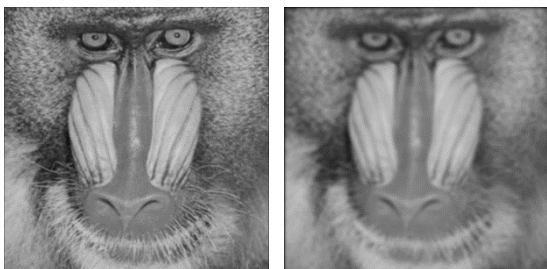
- Median 3*3



- Median 7*7



- Gaussian 5*5 $\sigma = 1.414$

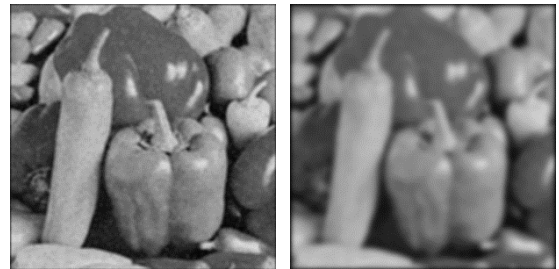


Peppers_noise (repeat 1, repeat 7)

- Mean 3*3



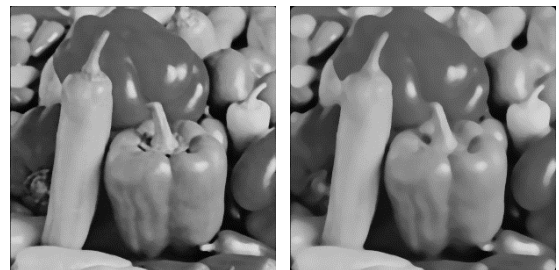
- Mean 7*7



- Median 3*3



- Median 7*7



- Gaussian 5*5 $\sigma = 1.414$

