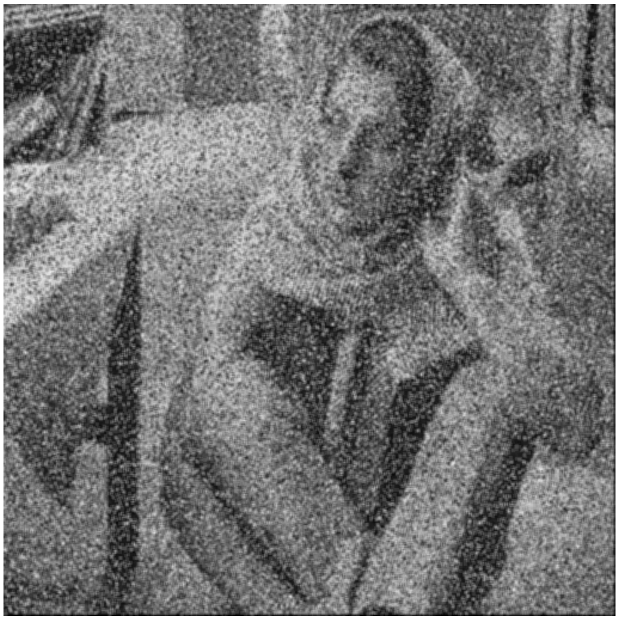
Computer Vision hw2

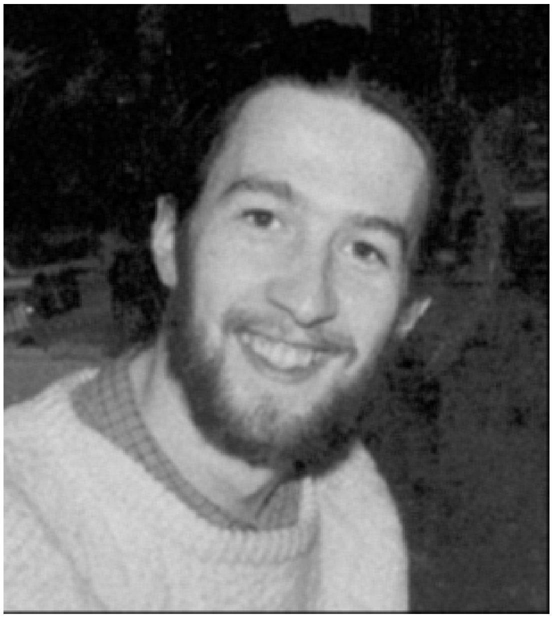
資工碩一 112598037 洪子翔

Original Image

Q1

Mean Filter
The function takes a mean kernel then zero padding, and does convolution on every pixel.
As you can see, because of the mean filtering does some kind of blur, the noise on the left image wasn’t removed at all, only became dimmer. The right image through the mean filter got smoother, some detail was removed.

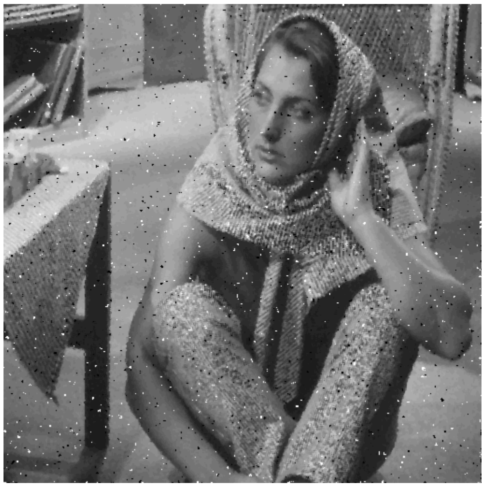


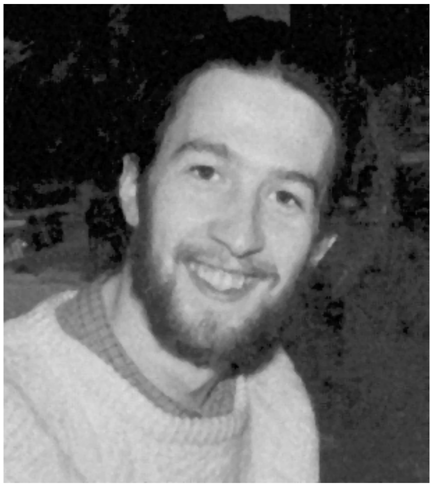


Q2

Median Filter

In question 2, the method is almost the same as question 1, but changes the core calculation to a different way. So it also could but not necessarily take a kernel as the input to set filter size, so value inside isn't important. The method also starts with zero padding, and does convolution on every pixel. The way to process the median filter I used is to do bubble sort on kernel size image data, then pick the one on the medium. In this approach, the left picture's noise has been removed a lot, because using a median filter could ignore the peak value. The right one does something like question 1, but a lil bit sharper.

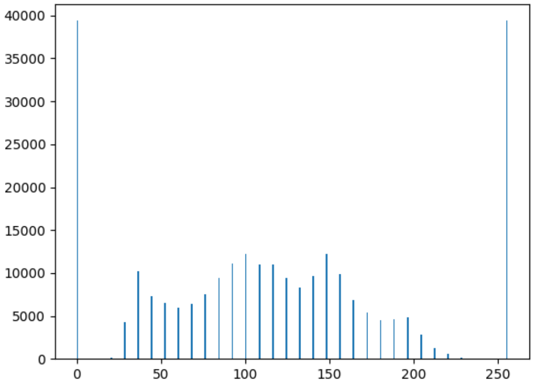




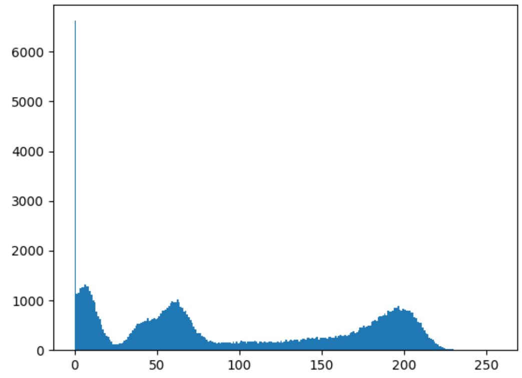
Q3

Question 3 is to draw histograms on all images. Because using `**matplotlib**` to draw is allowed, the only thing I need to do is to transform the image data to a one dimensional list, then using the function in matplotlib that could create a histogram.

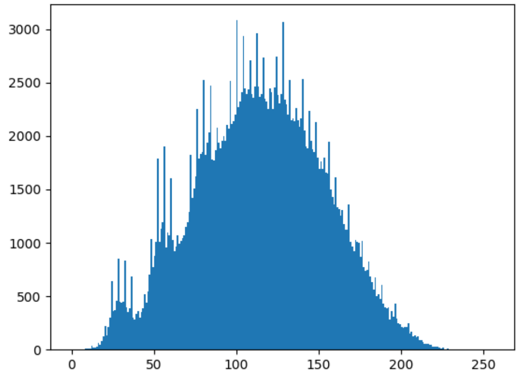
Original image noise1:
There are a lot of impulse value, and some pixel value doesn't exist on it.



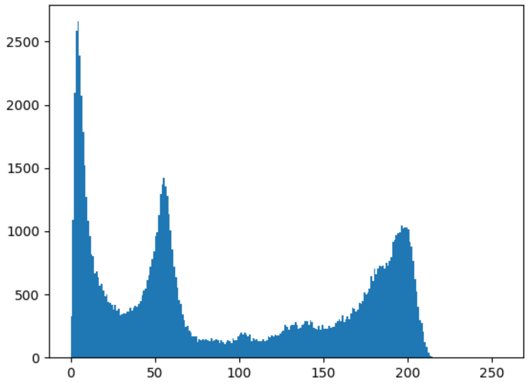
Original image noise2:
Lots of 0s.



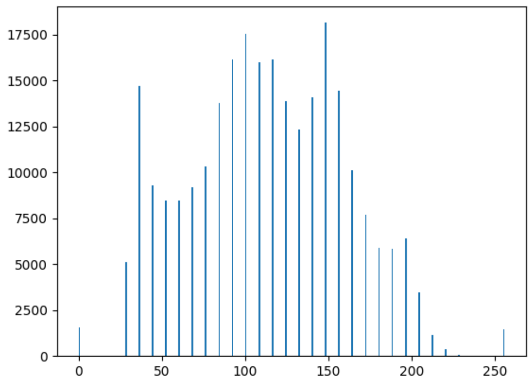
After mean filter:
After calculate the mean value, all of the impulse value was affected, and there is no gap between values.



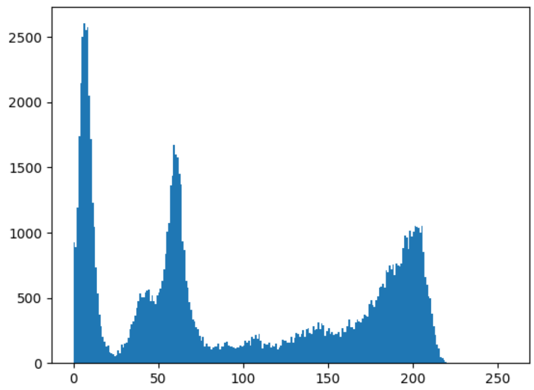
After mean filter:
Just make the distribution of the value on pixels more equally.



After median filter:
Because median filter only pick the median number near by the pixel to become new value, although the peak value could be Removed, the gaps between values retained



After median filter:
This histogram look almost the same as the one above…


2500
2000
1500
1000
500
0
0
50
100
150
200
250