**Computer Vision**

NTUT 資訊工程所111598070 賴俊霖

**Homework 2**

**1. Mean filter** (save as : output1.png)

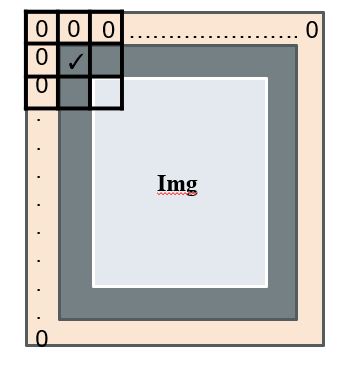
In meanFilter(img),you can see

Using n,m,\_ = img.shape to get length,width and channel’s data

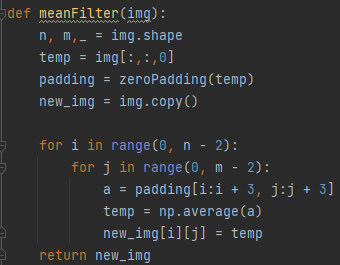
And img[:,:,0] to get a single channel and store in temp.

Then we copy an original img to take as a base.

We using zeroPadding to make the temp like pic below.



And we get the 3x3 of zeroPadding we created and use np.average(a) to get the Avg and replace the center Value of the position a which exactly the new\_img[i][j] we copied.



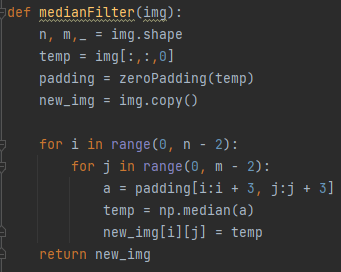
Images conversion as follow



2. **Median filter** (save as : output2.png)

The logic of Median filter is the same as Mean filter.

The only difference is using np.median(a) not np.average.



Images conversion as follow



3. **Image histogram** : Count the number of each pixel intensity.

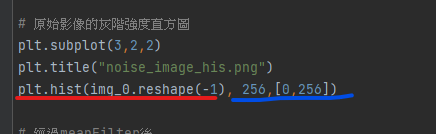
(save as : noise\_image\_his.png, output1\_his.png, output2\_his.png)

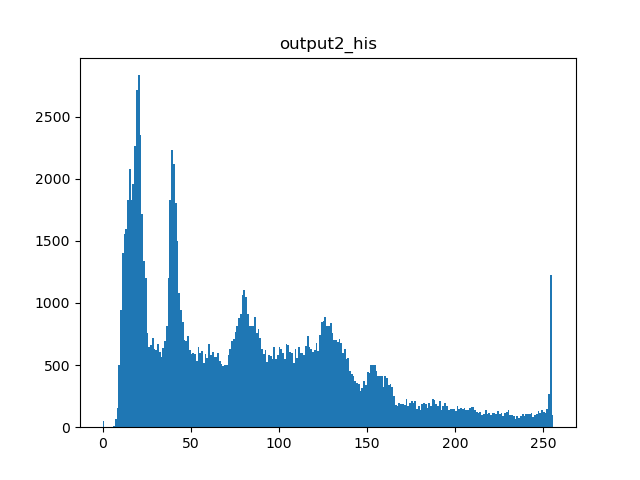
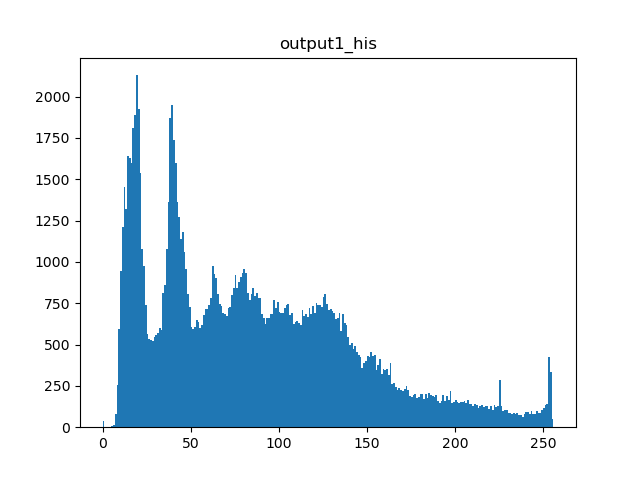
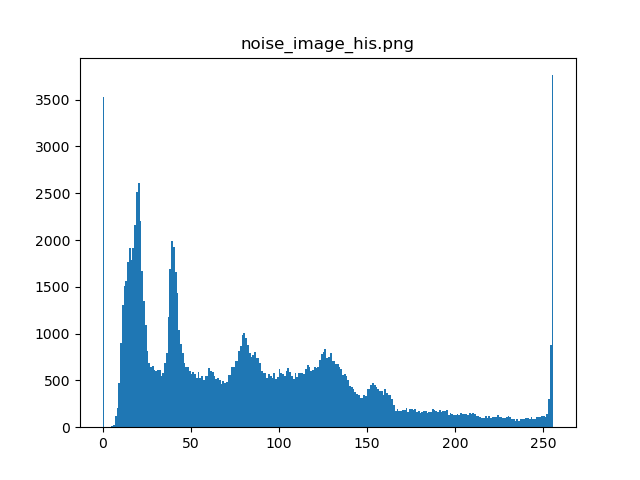
In Python ,we can easily get the hist using

img = img[:,:,0] to get single channel ,So that the statistic does not triple.

reshape(-1) is for converting 3D matrix to 1D list.

And we fixed the Minimum value and Maximum of coordinates.





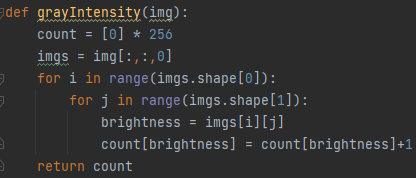
Instead of relying on the above approach, we can write our own as grayIntensity(img)

Count = [0] \* 256 to declare a 1-dimensional list with all 0's

Imgs = img[:,:,0] as mentioned before avoiding 3 times the number of counts.

We visited every point of this image and go through 0~255, then go to the count we created , use the number we just got as the index to do incremental.

Then return count to make the histogram.



4. **Compare** : noise\_image\_his.png, output1\_his.png, output2\_his.png

(Text description)

According to the original histogram, we can find that the pepper salt noise problem causes our values at 0 and 255 to be unusually high.

Using both the mean filter and the median filter, we can find that the part at 0 drops significantly and is quite close to 0.

Based on the previous output of Median Filter, we can also know that the image seems to be more suitable for using Median Filter as a denoise filter .

**5. 補充 zeroPadding**

In zeroPadding, first we created a bigger 2D matrix is full of zero, and stored the original pixel values to the new img from i=1 to (shape[0]-1) or (j=1 to shape[1]-1)

