Computer Vision Home Work 2

Name:許晉偉

Studentid:610415145

program and method

Q1: **Mean filter (save as : output1.png)**

```
: # mean filter
  def Meanfilter(x,y,noiseimg):
      mean_value=0
      for i in range(0,3):
          for j in range(0,3):
              mean_value=mean_value+noiseimg[x+i,y+j,0]
      mean_value=mean_value//9
      return mean_value

  for i in range(0,Height-2):
          for j in range(0,width-2):
              Mean_img[i,j,0]=Meanfilter(i,j,img)
  cv2.imwrite('output1.png',Mean_img)
```

The Meanfilter function use loop to get the noiseimg[x+0][y+0]~[x+2][y+2]value and add it up,before divide nine and return the value.

```
for i in range(0,Height-2):
        for j in range(0,width-2):
            Mean_img[i,j,0]=Meanfilter(i,j,img)
cv2.imwrite('output1.png',Mean_img)
```

The loop is use in get every pixel in the original picture and tranfer Variable for Meanfilter function,and store the value for new image when Meanfilter function return the value, and output the image.

Q2: Median filter (save as : output2.png)

```
# median filter
def Medianfilter(x,y,sourceimg):

    for i in range(0,3):
            for j in range(0,3):
                a[i,j]=sourceimg[x+i,y+j,0]

    for n in range(0,8):
        for i in range(0,3):
            for j in range(0,3):
                nc=i
                nr=j+1
                if(j==2):
                    nc=(i+1)
                    nr=0
                    if(nc==3):
                        break
                if(a[i,j]>a[nc,nr]):
                    temp=a[i,j]
                    a[i,j]=a[nc,nr]
                    a[nc,nr]=temp
    return(a[1,1])
for i in range(0,Height-2):
        for j in range(0,width-2):
            Median_img[i,j,0]=Medianfilter(i,j,img)
cv2.imwrite('output2.png',Median_img)
```

In the Medianfilter function ,first I create a temp array to store the value of the original picture, because it seems like a pointer, if I direct sort it, the original picture will be broken.

```
    for n in range(0,8):
        for i in range(0,3):
            for j in range(0,3):
                nc=i
                nr=j+1
                if(j==2):
                    nc=(i+1)
                    nr=0
                    if(nc==3):
                        break
                if(a[i,j]>a[nc,nr]):
                    temp=a[i,j]
                    a[i,j]=a[nc,nr]
                    a[nc,nr]=temp
    return(a[1,1])
```

After create a temp array, I do the 2-D bubble sort, The first loop is used in control sorting times and the other loop is used in get the value of the temp array[0][0]~[2][2],and sort it ,The variables i and j are the current position, nc and nr are the position of the next element, So they compare each other to finish sorting and return the middle element to the main function.

```
for i in range(0,Height-2):
        for j in range(0,width-2):
            Median_img[i,j,0]=Medianfilter(i,j,img)
cv2.imwrite('output2.png',Median_img)
```

The loop is use in get every pixel in the original picture and transfer Variable for Medianfilter function and store the value for new image when Medianfilter function return the value and output the image.

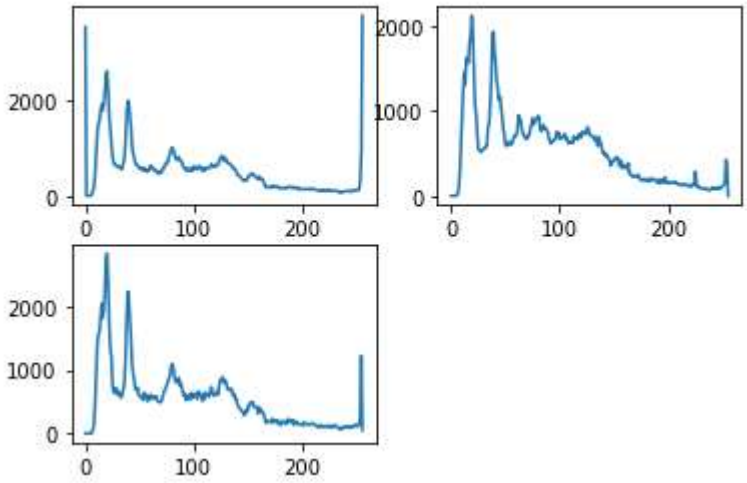
Q3:Image histogram

```
for i in range(0,Height):
            for j in range(0,width):
                a=img[i,j,0]
                noise_image_his[a]=noise_image_his[a]+1
for i in range(0,Height-2):
            for j in range(0,width-2):
                out1=Mean_img[i,j,0]
                out2=Median_img[i,j,0]
                output1_his[out1]=output1_his[out1]+1
                output2_his[out2]=output2_his[out2]+1
num_list = [i for i in range(0, 256)]

plt.plot(num_list,noise_image_his)
plt.savefig("noise_image_his.png")
plt.close()
plt.plot(num_list,output1_his)
plt.savefig("output1_his.png")
plt.close()
plt.plot(num_list,output2_his)
plt.savefig("output2_his.png")
plt.close()
```

The image between original and filtered have different size, so I need to use two loops to deal with it, The two loops doing the same way that Count the number of each pixel intensity, for example if detected the pixel intensity is zero then the array[0] will be add one, When I read array, I will know how many pixel intensity on the image, The array can represent y axis and create a list of 0~256 as x axis and plot it to become Image histogram and save the image.

Q4: Compare : noise_image_his.png, output1_his.png, output2_his.png

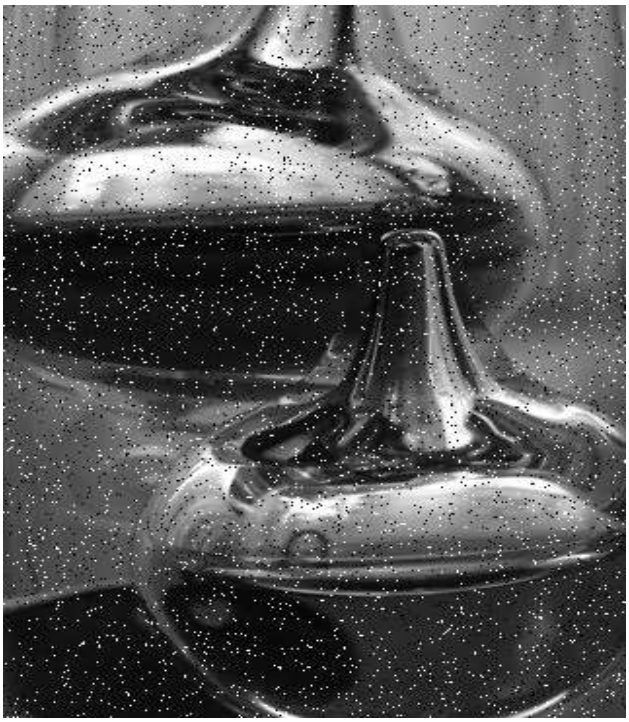


Fig(upper left corner): noise_image_his.png
Fig(upper right corner): output1_his.png
Fig(lower left corner): output2_his.png

In the image histogram , I found that the median filter is similar to the original image, but the mean filter doesn't, I think it is because the median filter is to take the pixels in the original image, but the mean filter is to get the average value of the original image , so the mean filter destroys the original image ,Causing the image becomes blurred after the mean filter, but the median filter doesn't.

Result images

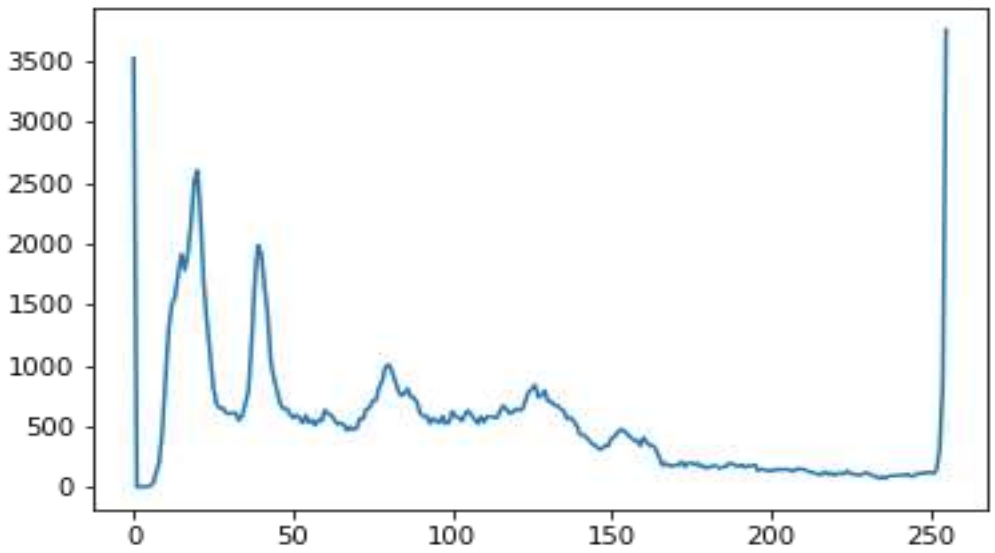
1.noise_image



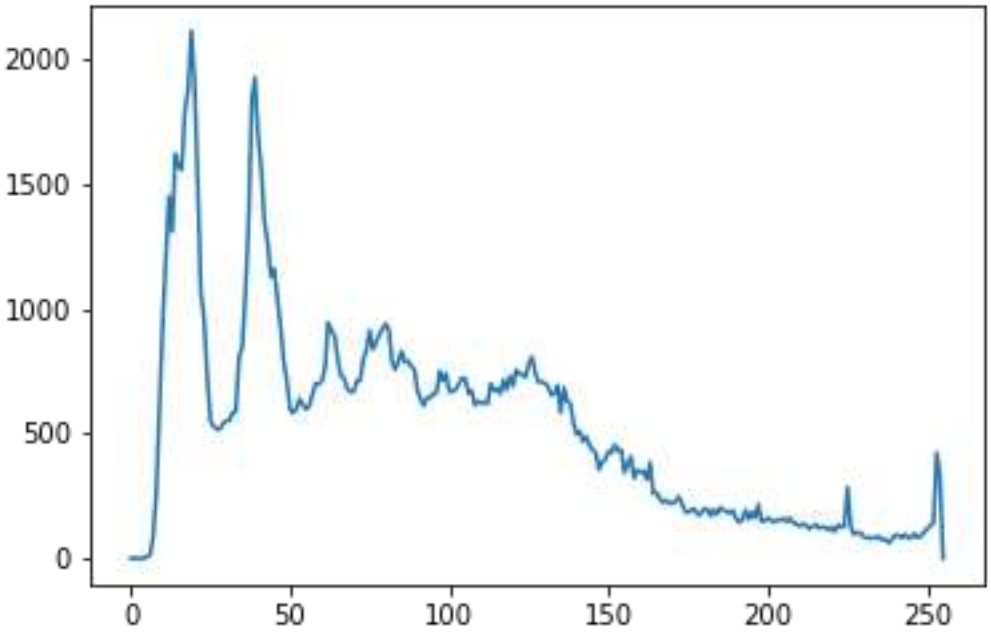
2. output1.png

3. output2.png

4. noise_image_his.png



5. output1_his.png



6. output2_his.png