

1. Input image



2.

(a) average filter



(b) median filter



3. Unsharp masking

$k = 0.5$

(a)



(b)



source code:

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib.image as mpimg
4  import cv2
5
6  def averageProcess(img, startx, starty, xnum, ynum):
7      cnt = 0
8      for i in range(ynum):
9          for j in range(xnum):
10             cnt += int(img[starty + i][startx + j][0])
11      res = cnt // 9
12      return [res, res, res]
13
14
15  def medianProcess(img, startx, starty, xnum, ynum):
16      temp = []
17      for i in range(ynum):
18          for j in range(xnum):
19             temp.append(int(img[starty + i][startx + j][0]))
20      temp.sort()
21      ret = temp[(len(temp) // 2)]
22      return [ret, ret, ret]
23
24
25  # read the image
26  img = mpimg.imread("input.jpg")
27
28  plt.imshow(img)
29  # disable axis
```

```

30 plt.axis('off')
31 # show the image
32 plt.show()
33
34 y = len(img)
35 x = len(img[0])
36
37 averageFilter = []
38 medianFilter = []
39
40 # get filter data
41 for i in range(y):
42
43     averageTemp = []
44     medianTemp = []
45
46     for j in range(x):
47         if i == 0 :
48             if j == 0:
49                 averageTemp.append(averageProcess(img, j, i, 2, 2))
50                 medianTemp.append(medianProcess(img, j, i, 2, 2))
51             elif j == x - 1:
52                 averageTemp.append(averageProcess(img, j - 1, i, 2, 2))
53                 medianTemp.append(medianProcess(img, j - 1, i, 2, 2))
54             else:
55                 averageTemp.append(averageProcess(img, j - 1, i, 3, 2))
56                 medianTemp.append(medianProcess(img, j - 1, i, 3, 2))
57         elif i == y - 1:
58             if j == 0:
59                 averageTemp.append(averageProcess(img, j, i - 1, 2, 2))
60                 medianTemp.append(medianProcess(img, j, i - 1, 2, 2))
61             elif j == x - 1:
62                 averageTemp.append(averageProcess(img, j - 1, i - 1, 2, 2))
63                 medianTemp.append(medianProcess(img, j - 1, i - 1, 2, 2))
64             else:
65                 averageTemp.append(averageProcess(img, j - 1, i - 1, 3, 2))
66                 medianTemp.append(medianProcess(img, j - 1, i - 1, 3, 2))
67         else :
68             if j == 0:
69                 averageTemp.append(averageProcess(img, j, i - 1, 2, 3))
70                 medianTemp.append(medianProcess(img, j, i - 1, 2, 3))
71             elif j == x - 1:
72                 averageTemp.append(averageProcess(img, j - 1, i - 1, 2, 3))
73                 medianTemp.append(medianProcess(img, j - 1, i - 1, 2, 3))
74             else:
75                 averageTemp.append(averageProcess(img, j - 1, i - 1, 3, 3))
76                 medianTemp.append(medianProcess(img, j - 1, i - 1, 3, 3))
77
78     averageFilter.append(averageTemp)
79     medianFilter.append(medianTemp)
80
81 # show average filter
82 plt.imshow(averageFilter)
83 # disable axis
84 plt.axis('off')
85 # show the image
86 plt.show()
87

```

```
88 # show median filter
89 plt.imshow(medianFilter)
90 # disable axis
91 plt.axis('off')
92 # show the image
93 plt.show()
94
95 npAverageFilter = np.array(averageFilter)
96 unsharpByAverage = np rint(np rint(img - 0.5 * npAverageFilter) / 0.5)
97 unsharpByAverage = unsharpByAverage.astype(int)
98 unsharpByAverage = np.clip(unsharpByAverage, 0, 255)
99
100 # show unsharp by average filter image
101 plt.imshow(unsharpByAverage)
102 # disable axis
103 plt.axis('off')
104 # show the image
105 plt.show()
106 # print('done')
107
108 npMedianFilter = np.array(medianFilter)
109 unsharpByMedian = np rint(np rint(img - 0.5 * npMedianFilter) / 0.5)
110 unsharpByMedian = unsharpByMedian.astype(int)
111 unsharpByMedian = np.clip(unsharpByMedian, 0, 255)
112
113 # show unsharp by median filter image
114 plt.imshow(unsharpByMedian)
115 # disable axis
116 plt.axis('off')
117 # show the image
118 plt.show()
119 # print('done')
```

Comment

這次作業花比較多時間在處理形態上的問題，因為 k 是 float，在 show 圖的時候要記得把運算完的 np array 裡面變成 int 且要 clip 進 0 到 255 內。