

```
68 #dilation_gray
69 img = cv2.imread("lena.bmp")
70 height=img.shape[0]
71 width=img.shape[1]
72 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
73 cv2.imwrite("gray.bmp", gray)
74 gray.flags.writeable = True
75 kernel=np.array([[0,1,1,1,0],
76                  [1,1,1,1,1],
77                  [1,1,1,1,1],
78                  [1,1,1,1,1],
79                  [0,1,1,1,0]])
80 dilation=np.zeros((height,width),dtype=int)
81
82 for i in range(height):
83     for j in range(width):
84         tmp=0
85         for x in range(-2,3,1):
86             for y in range(-2,3,1):
87                 if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
88                     if kernel[x+2][y+2]==1 and gray[i+x][j+y]+kernel[x+2][y+2]>tmp:
89                         tmp=gray[i+x][j+y]+kernel[x+2][y+2]
90                     dilation[i][j]=tmp
91
92 dilation_img=Image.fromarray(dilation)
93 dilation_img=dilation_img.convert("L")
94 dilation_img.save("dilation.bmp")
95
96
```



dilation

```

100 #erosion_gray
101 img = cv2.imread("lena.bmp")
102 height=img.shape[0]
103 width=img.shape[1]
104 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
105 cv2.imwrite("gray.bmp", gray)
106 gray.flags.writeable = True
107 kernel=np.array([[0,1,1,1,0],
108                 [1,1,1,1,1],
109                 [1,1,1,1,1],
110                 [1,1,1,1,1],
111                 [0,1,1,1,0]])
112 kernel_sum=0
113 for i in range(kernel.shape[0]):
114     for j in range(kernel.shape[1]):
115         kernel_sum=kernel_sum+kernel[i][j]
116 erosion=np.zeros((height,width),dtype=int)
117 for i in range(height):
118     for j in range(width):
119         tmp=gray[i][j]
120         for x in range(-2,3,1):
121             for y in range(-2,3,1):
122
123                 if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
124                     if kernel[x+2][y+2]==1 and gray[i+x][j+y]-kernel[x+2][y+2]<tmp:
125                         tmp=gray[i+x][j+y]-kernel[x+2][y+2]
126                     erosion[i][j]=tmp
127                 else:
128                     tmp=0
129                     erosion[i][j]=tmp
130 erosion_img=Image.fromarray(erosion)
131 erosion_img=erosion_img.convert("L")
132 erosion_img.save("erosion.bmp")
133
134

```



erosion

```

139 #opening_gray
140 img = cv2.imread("Lena.bmp")
141 height=img.shape[0]
142 width=img.shape[1]
143 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
144 cv2.imwrite("gray.bmp", gray)
145 gray.flags.writeable = True
146 kernel=np.array([[0,1,1,1,0],
147                 [1,1,1,1,1],
148                 [1,1,1,1,1],
149                 [1,1,1,1,1],
150                 [0,1,1,1,0]])
151 kernel_sum=0
152 for i in range(kernel.shape[0]):
153     for j in range(kernel.shape[1]):
154         kernel_sum=kernel_sum+kernel[i][j]
155 erosion=np.zeros((height,width),dtype=int)
156 for i in range(height):
157     for j in range(width):
158         tmp=gray[i][j]
159         for x in range(-2,3,1):
160             for y in range(-2,3,1):
161                 if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
162                     if kernel[x+2][y+2]==1 and gray[i+x][j+y]-kernel[x+2][y+2]<tmp:
163                         tmp=gray[i+x][j+y]-kernel[x+2][y+2]
164                         erosion[i][j]=tmp
165                 else:
166                     tmp=0
167                     erosion[i][j]=tmp
168
169 kernel=np.array([[0,1,1,1,0],
170                 [1,1,1,1,1],
171                 [1,1,1,1,1],
172                 [1,1,1,1,1],
173                 [0,1,1,1,0]])
174 opening=np.zeros((height,width),dtype=int)
175 for i in range(height):
176     for j in range(width):
177         tmp=0
178         for x in range(-2,3,1):
179             for y in range(-2,3,1):
180                 if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
181                     if kernel[x+2][y+2]==1 and erosion[i+x][j+y]+kernel[x+2][y+2]>tmp:
182                         tmp=erosion[i+x][j+y]+kernel[x+2][y+2]
183                     opening[i][j]=tmp
184 opening_img=Image.fromarray(opening)
185 opening_img=opening_img.convert("L")
186 opening_img.save("opening.bmp")
187

```



opening

```

192 #closing_gray
193 img = cv2.imread("Lena.bmp")
194 height=img.shape[0]
195 width=img.shape[1]
196 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
197 cv2.imwrite("gray.bmp", gray)
198 gray.flags.writeable = True
199 kernel=np.array([[0,1,1,1,0],
200                 [1,1,1,1,1],
201                 [1,1,1,1,1],
202                 [1,1,1,1,1],
203                 [0,1,1,1,0]])
204 dilation=np.zeros((height,width),dtype=int)
205 for i in range(height):
206     for j in range(width):
207         tmp=0
208         for x in range(-2,3,1):
209             for y in range(-2,3,1):
210                 if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
211                     if kernel[x+2][y+2]==1 and gray[i+x][j+y]+kernel[x+2][y+2]>tmp:
212                         tmp=gray[i+x][j+y]+kernel[x+2][y+2]
213                     dilation[i][j]=tmp
214
215 kernel=np.array([[0,1,1,1,0],
216                 [1,1,1,1,1],
217                 [1,1,1,1,1],
218                 [1,1,1,1,1],
219                 [0,1,1,1,0]])
220 kernel_sum=0
221 for i in range(kernel.shape[0]):
222     for j in range(kernel.shape[1]):
223         kernel_sum=kernel_sum+kernel[i][j]
224 closing=np.zeros((height,width),dtype=int)
225 for i in range(height):
226     for j in range(width):
227         tmp=dilation[i][j]
228         for x in range(-2,3,1):
229             for y in range(-2,3,1):
230                 if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
231                     if kernel[x+2][y+2]==1 and dilation[i+x][j+y]-kernel[x+2][y+2]<tmp:
232                         tmp=dilation[i+x][j+y]-kernel[x+2][y+2]
233                     closing[i][j]=tmp
234             else:
235                 tmp=0
236         closing[i][j]=tmp
237 closing_img=Image.fromarray(closing)
238 closing_img=closing_img.convert("L")
239 closing_img.save("closing.bmp")
240

```



closing