

```

1 import sys
2 import cv2 as cv
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import math
6
7 path = "C:/Users/Raiyan/Desktop/myout/bfly.png"
8
9 img = cv.imread(path)
10 img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
11
12 rows = img.shape[0]
13 cols = img.shape[1]
14 MN = rows * cols
15 L = 256
16
17 def show_img(image, title):
18     plt.imshow(image, 'gray')
19     plt.title(f'{title} image')
20     plt.show()
21
22 # flatten, , range
23 def show_hist(image, title):
24     plt.hist(image.ravel(), 256, [0, 256])
25     plt.title(f'{title} histogram')
26     plt.show()
27
28 def make_gauss(mean, sigma):
29
30     gauss = np.zeros(256, np.float32)
31     for i in range(256):
32         temp = math.exp( -1.0 * (i - mean) * (i - mean) / (sigma * sigma) )
33         temp2 = 1.0 / ( math.sqrt( 2 * np.pi) * sigma )
34         gauss[i] = temp * temp2
35
36     return gauss
37
38 show_img(img, "input")
39 show_hist(img, "input")
40
41 gauss_1 = make_gauss(80.0, 20.0)
42 gauss_2 = make_gauss(150.0, 50.0)
43
44 plt.plot(gauss_1)
45 plt.title("Gaussian function 1")
46 plt.show()
47
48 plt.plot(gauss_2)
49 plt.title("Gaussian function 2")
50 plt.show()
51
52 gauss = gauss_1 + gauss_2
53
54 plt.plot(gauss)
55 plt.title("Final Gaussian function")
56 plt.show()
57
58 freq_gauss = gauss
59 sum_gauss = gauss.sum()
60
61 cdf_gauss = pdf_gauss = np.zeros(256, np.float32)
62 pdf_gauss = freq_gauss / sum_gauss
63 cdf_gauss[0] = 0.0
64
65 for i in range(1, 255):
66     cdf_gauss[i] = cdf_gauss[i-1] + pdf_gauss[i]
67
68
69 for i in range(256):
70     cdf_gauss[i] = round(cdf_gauss[i]*255)
71
72 #####
73
74 cdf = pdf = freq = np.zeros(256, np.int32)
75
76 for i in range(rows):
77     for j in range(cols):

```

```
78         freq[ int(img[i][j]) ] += 1
79
80 pdf = freq / MN
81 cdf = pdf.cumsum()
82
83 for i in range(256):
84     cdf[i] = round(cdf[i] * 255.0)
85
86 #####
87
88 for i in range(rows):
89     for j in range(cols):
90
91         m_dist = sys.maxsize
92         f_px = px = int(img[i][j])
93         from_cdf = cdf[px]
94
95         for each_val in range(256):
96             x = cdf_gauss[ each_val ] - from_cdf
97             x = x * -1 if x < 0 else x
98             if x < m_dist:
99                 m_dist = x
100             f_px = each_val
101
102         img[i][j] = from_cdf = f_px
103
104 show_img(img, "enhanced")
105 show_hist(img, "output")
106
107
```