

```

1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 def get_pixel(img, center, x, y):
6     new_value = 0
7     try:
8         if img[x][y] >= center:
9             new_value = 1
10    except:
11        pass
12    return new_value
13
14 def lbp_calculated_pixel(img, x, y):
15     '''
16     64 | 128 | 1
17     -----
18     32 | 0 | 2
19     -----
20     16 | 8 | 4
21     '''
22
23     center = img[x][y]
24     val_ar = []
25     val_ar.append(get_pixel(img, center, x-1, y+1)) # top_right
26     val_ar.append(get_pixel(img, center, x, y+1)) # right
27     val_ar.append(get_pixel(img, center, x+1, y+1)) # bottom_right
28     val_ar.append(get_pixel(img, center, x+1, y)) # bottom
29     val_ar.append(get_pixel(img, center, x+1, y-1)) # bottom_left
30     val_ar.append(get_pixel(img, center, x, y-1)) # left
31     val_ar.append(get_pixel(img, center, x-1, y-1)) # top_left
32     val_ar.append(get_pixel(img, center, x-1, y)) # top
33
34     power_val = [1, 2, 4, 8, 16, 32, 64, 128]
35     val = 0
36     for i in range(len(val_ar)):
37         val += val_ar[i] * power_val[i]
38     return val
39
40
41 def show_output(output_list):
42     output_list_len = len(output_list)
43     figure = plt.figure()
44     for i in range(output_list_len):
45         current_dict = output_list[i]
46         current_img = current_dict["img"]
47         current_xlabel = current_dict["xlabel"]
48         current_ylabel = current_dict["ylabel"]
49         current_xtick = current_dict["xtick"]
50         current_ytick = current_dict["ytick"]
51         current_title = current_dict["title"]
52         current_type = current_dict["type"]
53         current_plot = figure.add_subplot(1, output_list_len, i+1)
54         if current_type == "gray":
55             current_plot.imshow(current_img, cmap = plt.get_cmap('gray'))
56             current_plot.set_title(current_title)
57             current_plot.set_xticks(current_xtick)
58             current_plot.set_yticks(current_ytick)
59             current_plot.set_xlabel(current_xlabel)
60             current_plot.set_ylabel(current_ylabel)
61         elif current_type == "histogram":
62             current_plot.plot(current_img, color = "black")
63             current_plot.set_xlim([0,260])
64             current_plot.set_title(current_title)
65             current_plot.set_xlabel(current_xlabel)
66             current_plot.set_ylabel(current_ylabel)
67             ytick_list = [int(i) for i in current_plot.get_yticks()]
68             current_plot.set_yticklabels(ytick_list,rotation = 90)
69

```

```

70     plt.show()
71
72 def main():
73     img_bgr = cv2.imread('images/color_ball.jpg')
74     height, width, channel = img_bgr.shape
75     img_gray = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)
76
77     img_lbp = np.zeros((height, width, 3), np.uint8)
78     for i in range(0, height):
79         for j in range(0, width):
80             img_lbp[i, j] = lbp_calculated_pixel(img_gray, i, j)
81     hist_lbp = cv2.calcHist([img_lbp], [0], None, [256], [0, 256])
82     output_list = []
83     output_list.append({
84         "img": img_gray,
85         "xlabel": "",
86         "ylabel": "",
87         "xtick": [],
88         "ytick": [],
89         "title": "Gray Image",
90         "type": "gray"
91     })
92     output_list.append({
93         "img": img_lbp,
94         "xlabel": "",
95         "ylabel": "",
96         "xtick": [],
97         "ytick": [],
98         "title": "LBP Image",
99         "type": "gray"
100    })
101    output_list.append({
102        "img": hist_lbp,
103        "xlabel": "Bins",
104        "ylabel": "Number of pixels",
105        "xtick": None,
106        "ytick": None,
107        "title": "Histogram(LBP)",
108        "type": "histogram"
109    })
110
111    show_output(output_list)
112
113    cv2.waitKey(0)
114    cv2.destroyAllWindows()
115    print("LBP Program is finished")
116
117 if __name__ == '__main__':
118     main()

```