

```
1 import cv2
2 import numpy as np
3 import os
4
5 # PS1-2 (1) User input feature
6 user_input = input("Please name your input color
7 file: ")
8 file_directory = os.getcwd()
9 image_location = os.path.join(file_directory,
10 user_input)
11 if os.path.exists(image_location):
12     print(f"Your '{user_input}' image loaded
13     successfully.")
14 else:
15     print(f"Error: unable to load your input image.
16     \nPlease make sure '{user_input}' is in the correct
17     directory.")
18     exit()
19
20 emphasize_choice = input("Please indicate your
21 emphasize region, 'dark' or 'bright'? : ")
22 if emphasize_choice == 'dark':
23     print(f"You choose '{emphasize_choice}' as your
24     emphasize region .")
25 elif emphasize_choice == 'bright':
26     print(f"You choose '{emphasize_choice}' as your
27     emphasize region .")
28 else:
29     print("Error: unable execute region emphasis.")
30     exit()
31
32 # PS1-2 (1) Open and load image files
33 if user_input == 'circuit.png':
34     circuit_color = cv2.imread(user_input)
35     cv2.imshow('Circuit Color Image', circuit_color
36 )
37     cv2.waitKey(0)
38     cv2.destroyAllWindows('Circuit Color Image')
39 elif user_input == 'crack.png':
40     crack_color = cv2.imread(user_input)
41     cv2.imshow('Crack Color Image', crack_color)
```

```

33     cv2.waitKey(0)
34     cv2.destroyAllWindows('Crack Color Image')
35
36 # PS1-2 (2) Convert color image to grayscale
37 if user_input == 'circuit.png':
38     circuit_color = cv2.imread(user_input)
39     circuit_grey = cv2.cvtColor(circuit_color, cv2.
        COLOR_BGR2GRAY)
40     cv2.imshow('Circuit Grey Image', circuit_grey)
41     cv2.waitKey(0)
42     cv2.destroyAllWindows('Circuit Grey Image')
43     cv2.imwrite('circuit_greyscale.png',
        circuit_grey)
44 elif user_input == 'crack.png':
45     circuit_color = cv2.imread(user_input)
46     crack_grey = cv2.cvtColor(circuit_color, cv2.
        COLOR_BGR2GRAY)
47     cv2.imshow('Crack Grey Image', crack_grey)
48     cv2.waitKey(0)
49     cv2.destroyAllWindows('Circuit Grey Image')
50     cv2.imwrite('crack_greyscale.png', crack_grey)
51
52 # PS1-2 (3) Convert greyscale image to binary
53 if user_input == 'circuit.png':
54     circuit_grey = cv2.imread('circuit_greyscale.
        png', cv2.IMREAD_GRAYSCALE)
55     _, circuit_binary = cv2.threshold(circuit_grey
        , 95, 255, cv2.THRESH_BINARY)
56     cv2.imshow('Circuit Binary Image',
        circuit_binary)
57     cv2.waitKey(0)
58     cv2.destroyAllWindows('Circuit Binary Image')
59     cv2.imwrite('circuit_binary.png',
        circuit_binary)
60 elif user_input == 'crack.png':
61     crack_grey = cv2.imread('crack_greyscale.png',
        cv2.IMREAD_GRAYSCALE)
62     _, crack_binary = cv2.threshold(crack_grey, 180
        , 255, cv2.THRESH_BINARY)
63     cv2.imshow('Crack Binary Image', crack_binary)
64     cv2.waitKey(0)

```

```

65     cv2.destroyAllWindows('Crack Binary Image')
66     cv2.imwrite('crack_binary.png', crack_binary)
67
68 # PS1-2 (4) Painting image pixel
69 # Circuit image color conversion
70 if user_input == 'circuit.png':
71     circuit_color = cv2.imread(user_input)
72     circuit_color_hsv = cv2.cvtColor(circuit_color
    , cv2.COLOR_BGR2HSV)
73     lower_green = np.array([10, 100, 100])
74     upper_green = np.array([20, 100, 100])
75     lower_yellow = np.array([10, 85, 100])
76     upper_yellow = np.array([70, 255, 255])
77
78     mask_green = cv2.inRange(circuit_color_hsv,
    lower_green, upper_green)
79     mask_yellow = cv2.inRange(circuit_color_hsv,
    lower_yellow, upper_yellow)
80     mask_green_yellow_circuit = cv2.bitwise_or(
    mask_green, mask_yellow)
81     mask_red_circuit = np.zeros_like(
    mask_green_yellow_circuit)
82     if emphasize_choice == 'bright':
83         mask_red_circuit[mask_green_yellow_circuit
    > 0] = 255
84         circuit_output_image_bright =
    circuit_color.copy()
85         circuit_output_image_bright[np.where(
    mask_red_circuit == 255)] = [0, 0, 255]
86
87         cv2.imshow('Circuit Output Image (Bright)'
    , circuit_output_image_bright)
88         cv2.imwrite('circuit_output.png',
    circuit_output_image_bright)
89         cv2.waitKey(0)
90         cv2.destroyAllWindows('Circuit Output Image (
    Bright)')
91     elif emphasize_choice == 'dark':
92         mask_red_circuit[mask_green_yellow_circuit
    > 0] = 255
93         circuit_output_image_dark = circuit_color.

```

```

93 copy()
94     circuit_output_image_dark[np.where(
    mask_red_circuit != 255)] = [0, 0, 255]
95
96     cv2.imshow('Circuit Output Image (Dark)',
    circuit_output_image_dark)
97     cv2.imwrite('circuit_output.png',
    circuit_output_image_dark)
98     cv2.waitKey(0)
99     cv2.destroyWindow('Circuit Output Image (
    Dark)')
100
101 # Crack image color conversion
102 elif user_input == 'crack.png':
103     crack_color = cv2.imread(user_input)
104     crack_color_hsv = cv2.cvtColor(crack_color,
    cv2.COLOR_BGR2HSV)
105     lower_black = np.array([0, 0, 0])
106     upper_black = np.array([180, 180, 180])
107
108     mask_black = cv2.inRange(crack_color_hsv,
    lower_black, upper_black)
109     mask_black_crack = cv2.bitwise_or(mask_black,
    mask_black)
110     mask_red_crack = np.zeros_like(
    mask_black_crack)
111     if emphasize_choice == 'bright':
112         mask_red_crack[mask_black_crack > 0] = 255
113         crack_output_image_bright = crack_color.
    copy()
114         crack_output_image_bright[np.where(
    mask_black_crack != 255)] = [0, 0, 255]
115
116         cv2.imshow('Crack Output Image (Bright)',
    crack_output_image_bright)
117         cv2.imwrite('crack_output.png',
    crack_output_image_bright)
118         cv2.waitKey(0)
119         cv2.destroyWindow('Crack Output Image (
    Bright)')
120     elif emphasize_choice == 'dark':

```

```
121         mask_black_crack[mask_black_crack > 0] =  
255  
122         crack_output_image_dark = crack_color.copy  
123         crack_output_image_dark[np.where(  
mask_black_crack == 255)] = [0, 0, 255]  
124  
125         cv2.imshow('Crack Output Image (Dark)',  
crack_output_image_dark)  
126         cv2.imwrite('crack_output.png',  
crack_output_image_dark)  
127         cv2.waitKey(0)  
128         cv2.destroyWindow('Crack Output Image (  
Dark)')  
129 exit()
```