

画像処理

175751C 宮城孝明

2019 年 11 月 13 日

目次

1	課題 1	2
1.1	ソースコード	2
1.2	実行結果	2
2	課題 2	2
2.1	ソースコード	2
2.2	実行結果	4
3	課題 3	4
3.1	ソースコード	4
3.2	実行結果	6
4	課題 4	6
4.1	ソースコード	6
4.2	実行結果	8
5	課題 5	8
5.1	ソースコード	8
5.2	実行結果	8
6	課題 6	9
6.1	ソースコード	9
6.2	実行結果	10

1 課題 1

1.1 ソースコード

Listing 1 課題 1

```
1 from PIL import Image
2 from matplotlib import pyplot as plt
3
4 def ImageShow():
5     im = Image.open("sample.jpg")
6     plt.imshow(im)
7     plt.pause(10)
8     plt.clf()
9     print("clf() 起動しました")
10    plt.pause(5)
11    ImageShow()
12
13 if __name__ == "__main__":
14     ImageShow()
```

1.2 実行結果

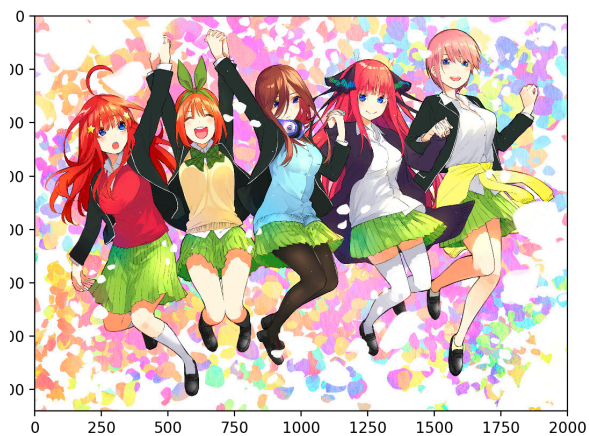


図 1 0 10 秒後と 15 秒

図 2 11 15 秒

2 課題 2

2.1 ソースコード

Listing 2 課題 3

```

1 from matplotlib import pyplot as plt
2 from PIL import Image
3 from matplotlib.backend_bases import MouseEvent
4 import cv2
5
6 class mouseParam:
7     def __init__(self, input_img_name):
8         #マウス入力用のパラメータ
9         self.mouseEvent = {"x":None, "y":None, "event":None, "flags":None}
10        #マウス入力の設定
11        cv2.setMouseCallback(input_img_name, self.__CallBackFunc, None)
12
13        #コールバック関数
14        def __CallBackFunc(self, eventType, x, y, flags, userdata):
15
16            self.mouseEvent["x"] = x
17            self.mouseEvent["y"] = y
18            self.mouseEvent["event"] = eventType
19            self.mouseEvent["flags"] = flags
20
21        #マウス入力用のパラメータを返すための関数
22        def getData(self):
23            return self.mouseEvent
24
25        #マウスイベントを返す関数
26        def getEvent(self):
27            return self.mouseEvent["event"]
28
29        #マウスフラグを返す関数
30        def getFlags(self):
31            return self.mouseEvent["flags"]
32
33        #の座標を返す関数x
34        def getX(self):
35            return self.mouseEvent["x"]
36
37        #の座標を返す関数y
38        def getY(self):
39            return self.mouseEvent["y"]
40
41        #との座標を返す関数xy
42        def getPos(self):
43            return (self.mouseEvent["x"], self.mouseEvent["y"])
44
45
46 if __name__ == "__main__":
47     #入力画像
48     read = cv2.imread("sample1.png")
49
50     #表示する名Window
51     window_name = "input_window"
52

```

```

53     #画像の表示
54     cv2.imshow(window_name, read)
55
56     #コールバックの設定
57     mouseData = mouseParam(window_name)
58
59     while 1:
60         cv2.waitKey(20)
61         #左クリックがあったら表示
62         if mouseData.getEvent() == cv2.EVENT_MOUSEMOVE:
63             print(mouseData.getPos())
64         #右クリックがあったら終了
65         elif mouseData.getEvent() == cv2.EVENT_RBUTTONDOWN:
66             break;
67
68     cv2.destroyAllWindows()
69     print("Finished")

```

2.2 実行結果



```

(62, 132)
(62, 132)
(62, 132)
(62, 132)
(62, 132)
(62, 132)
(62, 132)
(62, 132)
(62, 132)

```

図3 課題2の図

3 課題3

3.1 ソースコード

Listing 3 課題3

```

1 import cv2
2
3 class mouseParam:

```

```

4     def __init__(self, input_img_name):
5         #マウス入力用のパラメータ
6         self.mouseEvent = {"x":None, "y":None, "event":None, "flags":None}
7         #マウス入力の設定
8         cv2.setMouseCallback(input_img_name, self.__CallBackFunc, None)
9
10        #コールバック関数
11        def __CallBackFunc(self, eventType, x, y, flags, userdata):
12
13            self.mouseEvent["x"] = x
14            self.mouseEvent["y"] = y
15            self.mouseEvent["event"] = eventType
16            self.mouseEvent["flags"] = flags
17
18        #マウス入力用のパラメータを返すための関数
19        def getData(self):
20            return self.mouseEvent
21
22        #マウスイベントを返す関数
23        def getEvent(self):
24            return self.mouseEvent["event"]
25
26        #マウスフラグを返す関数
27        def getFlags(self):
28            return self.mouseEvent["flags"]
29
30        #の座標を返す関数x
31        def getX(self):
32            return self.mouseEvent["x"]
33
34        #の座標を返す関数y
35        def getY(self):
36            return self.mouseEvent["y"]
37
38        #との座標を返す関数xy
39        def getPos(self):
40            return (self.mouseEvent["x"], self.mouseEvent["y"])
41
42
43    if __name__ == "__main__":
44        #入力画像
45        read = cv2.imread("sample1.png")
46
47        #表示する名Window
48        window_name = "input_window"
49
50        #画像の表示
51        cv2.imshow(window_name, read)
52
53        #コールバックの設定
54        mouseData = mouseParam(window_name)
55
56        while 1:

```

```

57         cv2.waitKey(20)
58         #左クリックがあったら表示
59         if mouseData.getEvent() == cv2.EVENT_LBUTTONDOWN:
60             print(mouseData.getPos())
61         #右クリックがあったら終了
62         elif mouseData.getEvent() == cv2.EVENT_RBUTTONDOWN:
63             break;
64
65     cv2.destroyAllWindows()
66     print("Finished")

```

3.2 実行結果

```

(107, 67)
(364, 199)
(364, 199)
(364, 199)
(364, 199)
(364, 199)

```

図 4 課題 2 の図

4 課題 4

4.1 ソースコード

Listing 4 課題 4

```

1  import cv2
2
3
4  class mouseParam:
5      def __init__(self, input_img_name):
6          #マウス入力用のパラメータ
7          self.mouseEvent = {"x":None, "y":None, "event":None, "flags":None}
8          #マウス入力の設定
9          cv2.setMouseCallback(input_img_name, self.__CallBackFunc, None)
10
11     #コールバック関数
12     def __CallBackFunc(self, eventType, x, y, flags, userdata):
13
14         self.mouseEvent["x"] = x
15         self.mouseEvent["y"] = y
16         self.mouseEvent["event"] = eventType
17         self.mouseEvent["flags"] = flags
18

```

```

19     #マウス入力用のパラメータを返すための関数
20     def getData(self):
21         return self.mouseEvent
22
23     #マウスイベントを返す関数
24     def getEvent(self):
25         return self.mouseEvent["event"]
26
27     #マウスフラグを返す関数
28     def getFlags(self):
29         return self.mouseEvent["flags"]
30
31     #の座標を返す関数 $x$ 
32     def getX(self):
33         return self.mouseEvent["x"]
34
35     #の座標を返す関数 $y$ 
36     def getY(self):
37         return self.mouseEvent["y"]
38
39     #との座標を返す関数 $xy$ 
40     def getPos(self):
41         return (self.mouseEvent["x"], self.mouseEvent["y"])
42
43
44 if __name__ == "__main__":
45     #入力画像
46     read = cv2.imread("sample1.png")
47
48     #表示する名 $Window$ 
49     window_name = "input_□window"
50
51     #画像の表示
52     cv2.imshow(window_name, read)
53
54     #コールバックの設定
55     mouseData = mouseParam(window_name)
56
57     while 1:
58         cv2.waitKey(20)
59         #左クリックがあったら表示
60         if mouseData.getEvent() == cv2.EVENT_LBUTTONDOWN:
61             px = read[mouseData.getPos()]
62             print(px)
63             #右クリックがあったら終了
64             elif mouseData.getEvent() == cv2.EVENT_RBUTTONDOWN:
65                 break;
66
67     cv2.destroyAllWindows()
68     print("Finished")

```


4.2 実行結果

```
[128 128 128]
[31 30 38]
[255 255 255]
[255 255 255]
[255 255 255]
[49 37 37]
```

図 5 課題 2 の図

5 課題 5

5.1 ソースコード

Listing 5 課題 5

```
1 import os
2 import cv2
3
4 def main():
5     url = "sample.jpg"
6     im = cv2.imread(url)
7     print(im.shape)
8
9 if __name__ == "__main__":
10     main()
```

5.2 実行結果

```
(1480, 2000, 3)
```

図 6 課題 2 の図

6 課題 6

6.1 ソースコード

Listing 6 課題 6

```
1 import cv2
2
3
4 class mouseParam:
5     def __init__(self, input_img_name):
6         #マウス入力用のパラメータ
7         self.mouseEvent = {"x":None, "y":None, "event":None, "flags":None}
8         #マウス入力の設定
9         cv2.setMouseCallback(input_img_name, self.__CallBackFunc, None)
10
11     #コールバック関数
12     def __CallBackFunc(self, eventType, x, y, flags, userdata):
13
14         self.mouseEvent["x"] = x
15         self.mouseEvent["y"] = y
16         self.mouseEvent["event"] = eventType
17         self.mouseEvent["flags"] = flags
18
19     #マウス入力用のパラメータを返すための関数
20     def getData(self):
21         return self.mouseEvent
22
23     #マウスイベントを返す関数
24     def getEvent(self):
25         return self.mouseEvent["event"]
26
27     #マウスフラグを返す関数
28     def getFlags(self):
29         return self.mouseEvent["flags"]
30
31     #の座標を返す関数x
32     def getX(self):
33         return self.mouseEvent["x"]
34
35     #の座標を返す関数y
36     def getY(self):
37         return self.mouseEvent["y"]
38
39     #との座標を返す関数xy
40     def getPos(self):
41         return (self.mouseEvent["x"], self.mouseEvent["y"])
42
43
44 if __name__ == "__main__":
45     #入力画像
```

```

46 read = cv2.imread("sample1.png")
47
48 pxList=[]
49 count=0
50 #表示する名Window
51 window_name = "input_□window"
52
53 #画像の表示
54 cv2.imshow(window_name, read)
55
56 #コールバックの設定
57 mouseData = mouseParam(window_name)
58
59 while 1:
60     cv2.waitKey(20)
61     #左クリックがあったら表示
62     if mouseData.getEvent() == cv2.EVENT_LBUTTONDOWN:
63         count+=1
64         print(mouseData.getPos())
65         pxList.append(mouseData.getPos())
66         if count==2:
67             print("回目2")
68             count=0
69             print(pxList[0],pxList[1])
70             cv2.rectangle(read,pxList[0],pxList[1],(0,255,255),thickness=1)
71             print("実行されているはずだ")
72             pxList=[]
73             cv2.imwrite("sample2.png",read)
74             #右クリックがあったら終了
75             elif mouseData.getEvent() == cv2.EVENT_RBUTTONDOWN:
76                 break;
77 cv2.destroyAllWindows()
78 print("Finished")

```

6.2 実行結果

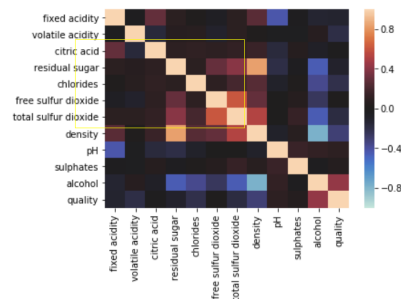


図7 課題2の図