

```

1  import cv2
2  import numpy as np
3  import sys
4
5  # read arguments
6  if(len(sys.argv) != 7) :
7      print(sys.argv[0], ":takes 6 arguments. Not", len(sys.argv)-1)
8      print("Expecting arguments: w1 h1 w2 h2 ImageIn ImageOut.")
9      print("Example:", sys.argv[0], "0.2 0.1 0.8 0.5 fruits.jpg out.png")
10     sys.exit()
11
12     w1 = float(sys.argv[1])
13     h1 = float(sys.argv[2])
14     w2 = float(sys.argv[3])
15     h2 = float(sys.argv[4])
16     name_input = sys.argv[5]
17     name_output = sys.argv[6]
18
19     # check the correctness of the input parameters
20     if(w1<0 or h1<0 or w2<=w1 or h2<=h1 or w2>1 or h2>1) :
21         print("arguments must satisfy 0<=w1<w2<=1, 0<=h1<h2<=1")
22         sys.exit()
23
24     # read image
25     inputImage = cv2.imread(name_input, cv2.IMREAD_COLOR)
26     if(inputImage is None) :
27         print(sys.argv[0], ":Failed to read image from:", name_input)
28         sys.exit()
29     cv2.imshow("input image:" + name_input, inputImage)
30
31     # check for color image and change w1, w2, h1, h2 to pixel locations
32     rows, cols, bands = inputImage.shape
33     if(bands != 3) :
34         print("Input image is not a standard color image:", inputImage)
35         sys.exit()
36
37     W1 = round(w1*(cols-1))
38     H1 = round(h1*(rows-1))
39     W2 = round(w2*(cols-1))
40     H2 = round(h2*(rows-1))
41
42     # The transformation should be applied only to
43     # the pixels in the W1,W2,H1,H2 range.
44     # The following code goes over these pixels
45
46     tmp1 = np.copy(inputImage)
47     for i in range(H1, H2+1) :
48         for j in range(W1, W2+1) :
49             b, g, r = inputImage[i, j]
50             gray = round(0.3*r + 0.6*g + 0.1*b + 0.5)
51             tmp1[i, j] = [gray, gray, gray]
52     cv2.imshow("replace_gray", tmp1)

```

```
53
54 # Slicing can be used for similar things
55 # In this example the red channel is zeroed out
56 tmp2 = np.copy(inputImage)
57 window_height = H2 - H1 + 1
58 window_width = W2 - W1 + 1
59 window = np.zeros([window_height, window_width], dtype=np.uint8)
60 tmp2[H1: H2+1, W1: W2+1, 2] = window
61 cv2.imshow("remove_red", tmp2)
62
63 # saving the output - save the gray window image
64 cv2.imwrite(name_output, tmp1)
65
66 # wait for key to exit
67 cv2.waitKey(0)
68 cv2.destroyAllWindows()
```