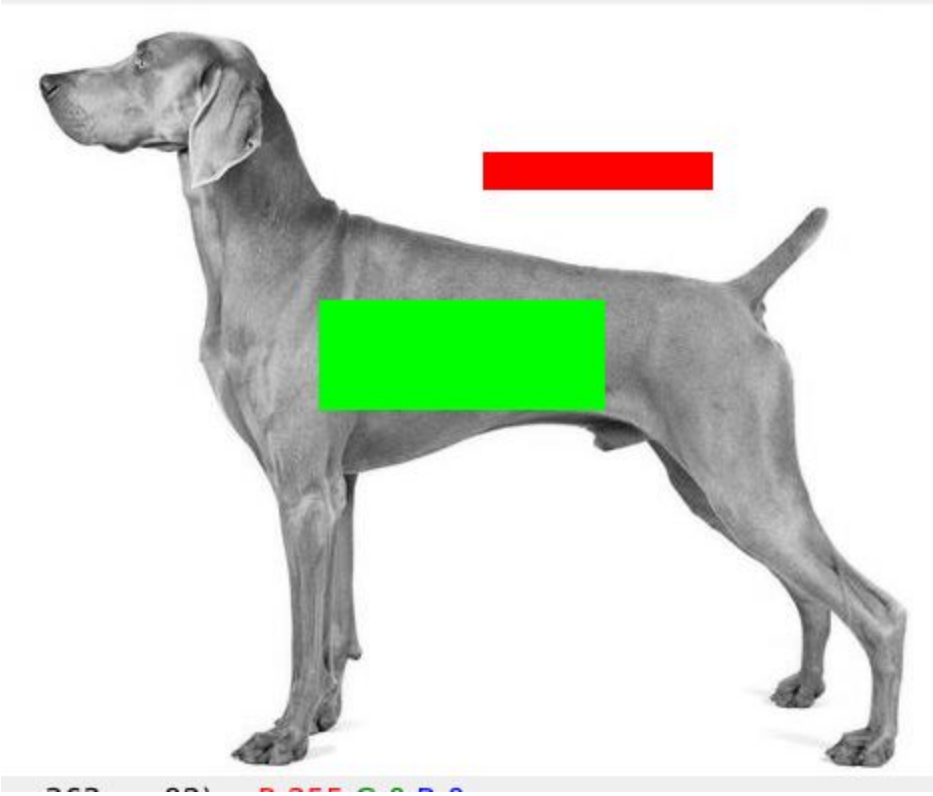
HW-7

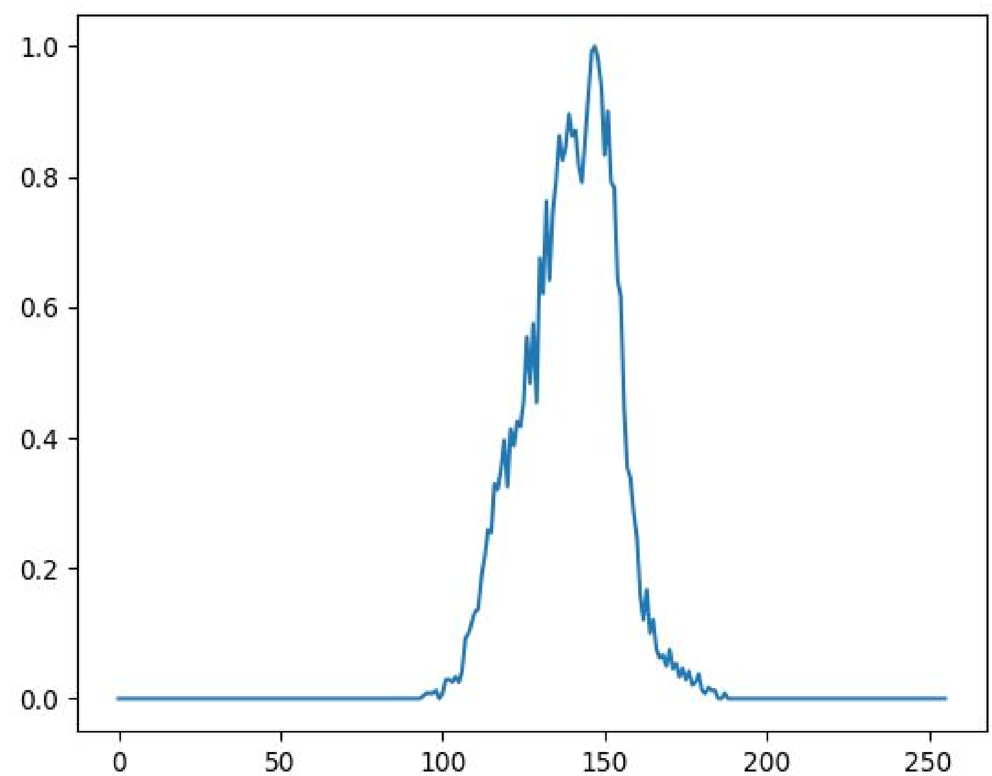
Input

Green = Foreground patch
Red = Background patch



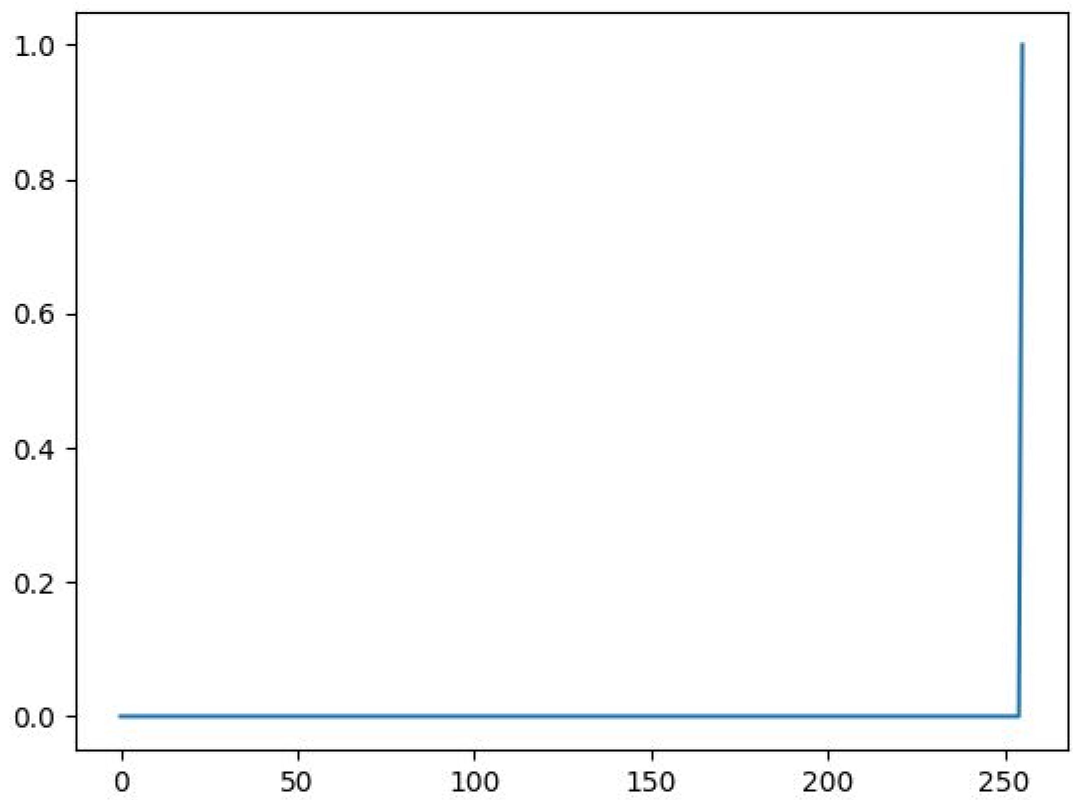
FG-map

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.004166667, 0.00833333, 0.00833333, 0.00833333, 0.0125, 0.0, 0.00833333, 0.029166667, 0.029166667, 0.025, 0.0333333, 0.025, 0.041666664, 0.09166666, 0.1, 0.11666667, 0.1333333, 0.1375, 0.1875, 0.21666667, 0.2583336, 0.25416665, 0.32916666, 0.3208336, 0.35, 0.395833, 0.325, 0.4125, 0.3875, 0.425, 0.4166667, 0.45416666, 0.5541667, 0.4833334, 0.575, 0.45416666, 0.675, 0.620833, 0.7625, 0.6416667, 0.745833, 0.7916666, 0.8625, 0.825, 0.845833, 0.895834, 0.8625, 0.870833, 0.8166667, 0.7916666, 0.858333, 0.9291667, 0.9916667, 1.0, 0.9791666, 0.9375, 0.833334, 0.9, 0.7916666, 0.783333, 0.6416667, 0.6166667, 0.4458336, 0.3541667, 0.3375, 0.283333, 0.2458332, 0.1583333, 0.1208333, 0.16666666, 0.1, 0.1208333, 0.075, 0.0625, 0.06666667, 0.05, 0.075, 0.0458333, 0.05416667, 0.0333333, 0.0458333, 0.029166667, 0.041666664, 0.02083332, 0.025, 0.0375, 0.0125, 0.00833333, 0.016666666, 0.0125, 0.0125, 0.0, 0.0, 0.00833333, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]



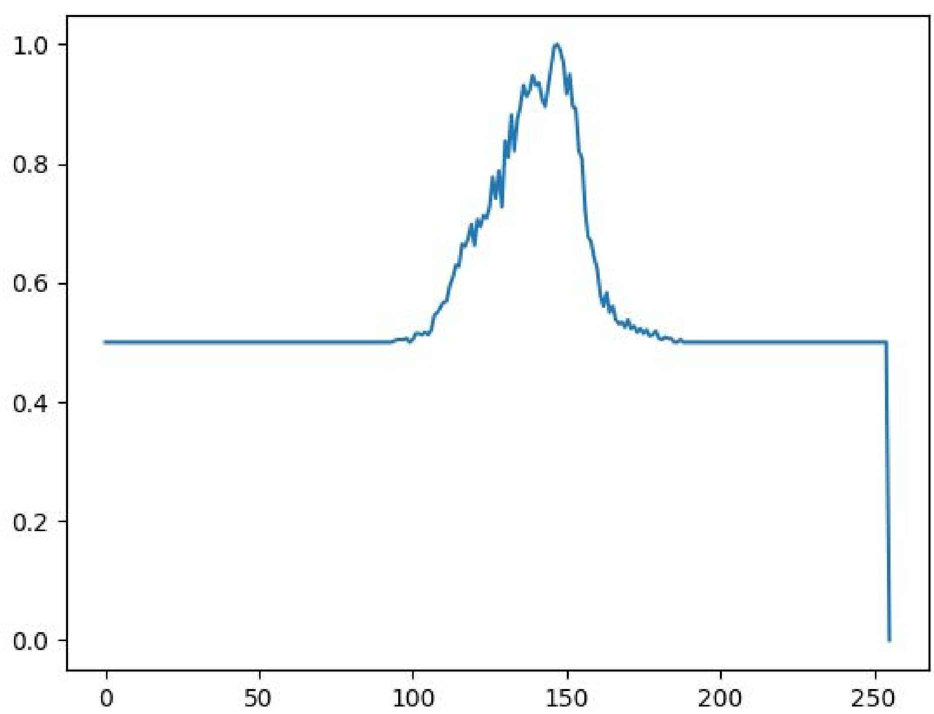
BG-Map

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0]

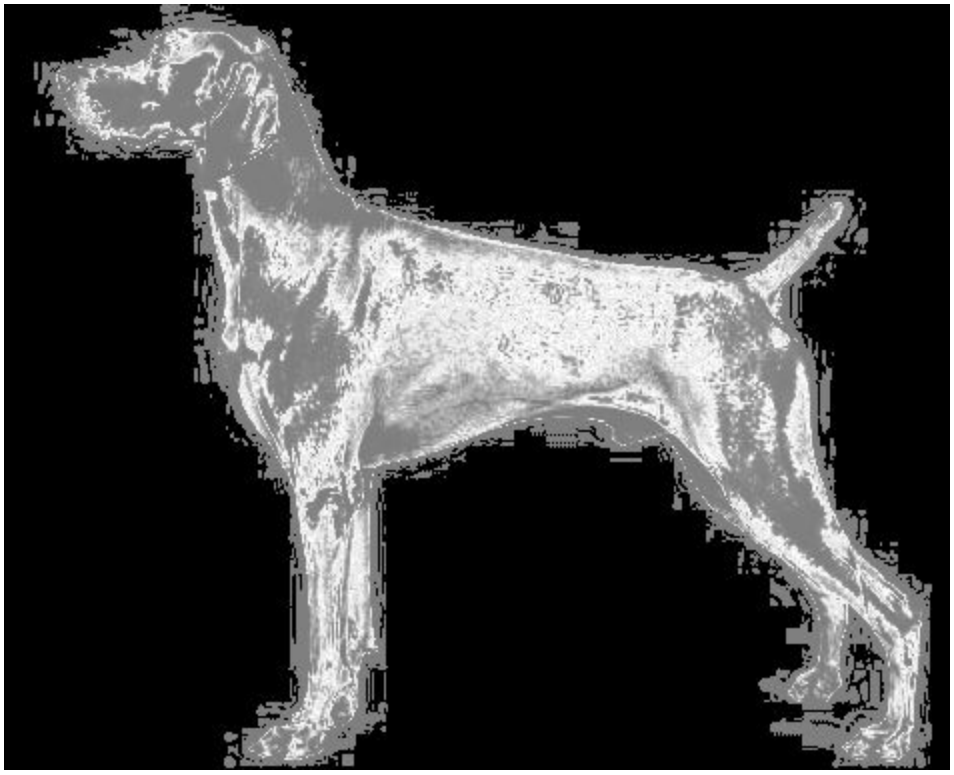


Salience-Map

[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.502083, 0.5041667, 0.5041667, 0.5041667, 0.50625, 0.5, 0.5041667, 0.514583,
0.514583, 0.5125, 0.5166667, 0.5125, 0.520834, 0.545833, 0.55, 0.558333, 0.5666667, 0.56875,
0.59375, 0.608334, 0.6291667, 0.627083, 0.664583, 0.6604167, 0.675, 0.6979166, 0.6625,
0.70625, 0.69375, 0.7125, 0.708334, 0.727083, 0.777083, 0.7416667, 0.7875, 0.727083, 0.8375,
0.8104167, 0.88125, 0.820833, 0.8729167, 0.895833, 0.93125, 0.9125, 0.9229166, 0.9479167,
0.93125, 0.9354167, 0.908333, 0.895833, 0.9291667, 0.964583, 0.995833, 1.0, 0.989583, 0.96875,
0.9166667, 0.95, 0.895833, 0.8916666, 0.820833, 0.808333, 0.7229167, 0.677084, 0.66875,
0.6416666, 0.6229167, 0.5791666, 0.5604167, 0.583334, 0.55, 0.5604167, 0.5375, 0.53125,
0.533333, 0.525, 0.5375, 0.5229167, 0.527083, 0.5166667, 0.5229167, 0.514583, 0.520834,
0.5104166, 0.5125, 0.51875, 0.50625, 0.5041667, 0.508333, 0.50625, 0.50625, 0.5, 0.5, 0.5041667,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.0]



Output



Code

```
import cv2
import sys
import copy
import numpy as np
import itertools
import time


f_x1, f_y1, b_x1, b_y1, f_x2, f_y2, b_x2, b_y2 = -1, -1, -1, -1, -1, -1,
-1, -1
drawing = False # true if mouse is pressed
ix, iy = -1, -1
count = 0
color_of_patch = (0, 255, 0)


def get_saliency_map(fg_map, bg_map):
    return [(i + (1 - j)) / 2 for i, j in zip(fg_map, bg_map)]


def get_likelihood_map(grayImage, x1, y1, x2, y2):
    x1, y1, x2, y2 = min(x1, x2), min(y1, y2), max(x1, x2), max(y1, y2)
    rows, cols = grayImage.shape
    frequency = [0] * 256

    if x1 < 0: x1 = 0
    if x2 < 0: x2 = 0
    if y1 < 0: y1 = 0
    if y2 < 0: y2 = 0
    if x1 > cols: x1 = cols
    if x2 > cols: x2 = cols
    if y1 > rows: y1 = rows
    if y2 > rows: y2 = rows

    for i in range(y1, y2):
       for j in range(x1, x2):
             frequency[grayImage[i][j]] += 1

    max_frequency = max(frequency)
    if max_frequency != 0:
       frequency = [i / max_frequency for i in frequency]
```

```
    return frequency


def take_patches_from_user(originalImage):
    # mouse callback function
    def draw_2_patches(event, x, y, flags, param):

        global ix, iy, drawing, mode, count, color_of_patch, f_x1, f_y1,
b_x1, b_y1, f_x2, f_y2, b_x2, b_y2

        if event == cv2.EVENT_LBUTTONDOWN:
              drawing = True
              ix, iy = x, y

        elif event == cv2.EVENT_MOUSEMOVE:
              if drawing == True:
                    cv2.rectangle(originalImage, (ix, iy), (x, y),
color_of_patch, -1)

        elif event == cv2.EVENT_LBUTTONUP:
              drawing = False
              cv2.rectangle(originalImage, (ix, iy), (x,y),
color_of_patch,-1)

              count += 1

              if count == 1:
                    f_x1, f_y1, f_x2, f_y2 = ix, iy, x, y
                    color_of_patch = (0, 0, 255)
                    print("Foreground patch done!\n")
                    print("Make background patch...")
              else:
                    b_x1, b_y1, b_x2, b_y2 = ix, iy, x, y
                    print("Background patch done!")

    cv2.namedWindow('image')
    cv2.setMouseCallback('image', draw_2_patches)

    print("Make forground patch...")

    while 1:
      cv2.imshow('image', originalImage)
```

```
      cv2.waitKey(1) & 0xFF
      if count == 2:
            time.sleep(0.5)
            break

    cv2.destroyAllWindows()


def main(image_path):
    originalImage = cv2.imread(image_path)
    take_patches_from_user(copy.deepcopy(originalImage))
    grayImage = cv2.cvtColor(originalImage, cv2.COLOR_BGR2GRAY)

    fg_map = get_likelihood_map(grayImage, f_x1, f_y1, f_x2, f_y2)
    bg_map = get_likelihood_map(grayImage, b_x1, b_y1, b_x2, b_y2)
    saliency_map = get_saliency_map(fg_map, bg_map)
    print("FG-map\n", fg_map)
    print("BG-map\n", bg_map)
    print("Salience map\n", saliency_map)
    saliency_map = [int(i * 255) for i in saliency_map]

    for i in range(grayImage.shape[0]):
       for j in range(grayImage.shape[1]):
            grayImage[i][j] = saliency_map[grayImage[i][j]]

    cv2.imshow('image', grayImage)
    cv2.waitKey()
    cv2.destroyAllWindows()



if __name__ == '__main__':
    if len(sys.argv) != 2:
       print("Usage: python3 code.py <path_of_image>")
       exit(1)

    image_path = sys.argv[1]
    main(image_path)
```