

Image registration and optimization technique

Name : Kevalya Gupta PRN : 1032210295 Batch : A1

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
from skimage.io import imread
from skimage.io import imshow

# Read the image
image1 = cv2.imread('samyakphone3.jpg')

# Convert the image to grayscale
gray_image = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)

# Display the grayscale image
plt.imshow(gray_image, cmap='gray')
plt.show()
```



image1

```
array([[[21, 19, 31],
        [23, 21, 33],
        [26, 24, 36],
        ...,
        [46, 53, 78],
        [44, 51, 76],
        [42, 49, 74]],

       [[21, 19, 31],
        [23, 21, 33],
        [26, 24, 36],
        ...,
        [48, 55, 80],
```

```

[47, 54, 79],
[45, 52, 77]],

[[22, 20, 32],
[24, 22, 34],
[26, 24, 36],
...,
[51, 58, 83],
[49, 56, 81],
[48, 55, 80]],

...,

[[45, 40, 37],
[43, 38, 35],
[40, 37, 33],
...,
[37, 36, 52],
[38, 37, 53],
[39, 38, 54]],

[[47, 42, 39],
[45, 40, 37],
[42, 39, 35],
...,
[37, 36, 52],
[39, 38, 54],
[40, 39, 55]],

[[48, 43, 40],
[46, 41, 38],
[43, 40, 36],
...,
[37, 36, 52],
[39, 38, 54],
[40, 39, 55]]], dtype=uint8)

```

```

# Read the image
image2 = cv2.imread('/content/samyakphone2.jpg')

# Convert the image to grayscale
gray_image = cv2.cvtColor(image2, cv2.COLOR_BGR2GRAY)

# Display the grayscale image
plt.imshow(gray_image, cmap='gray')
plt.show()

```

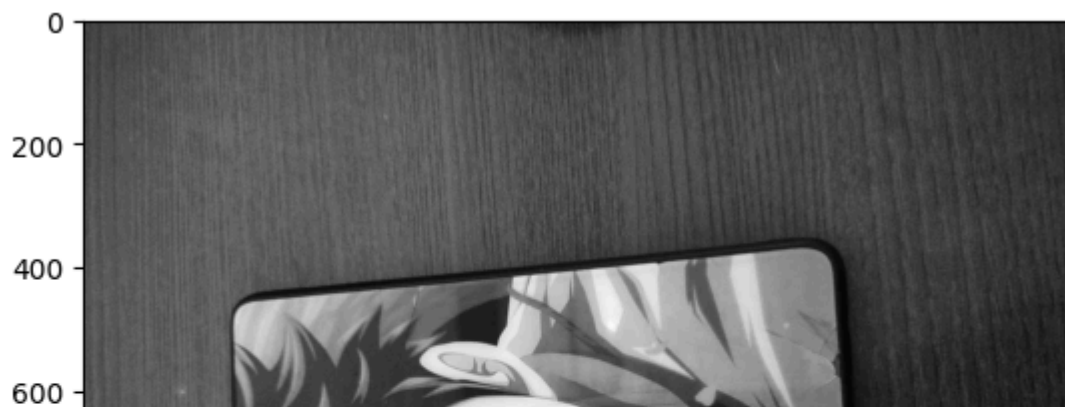


image2

```
array([[52, 58, 81],
       [50, 56, 79],
       [46, 52, 75],
       ...,
       [29, 34, 49],
       [29, 32, 47],
       [28, 31, 46]],

      [[53, 59, 82],
       [51, 57, 80],
       [48, 54, 77],
       ...,
       [35, 40, 55],
       [34, 37, 52],
       [32, 35, 50]],

      [[54, 60, 83],
       [53, 59, 82],
       [51, 57, 80],
       ...,
       [39, 44, 59],
       [37, 40, 55],
       [33, 36, 51]],

      ...,

      [[24, 26, 36],
       [23, 25, 35],
       [24, 25, 35],
       ...,
       [33, 37, 55],
       [34, 38, 56],
       [36, 40, 58]],

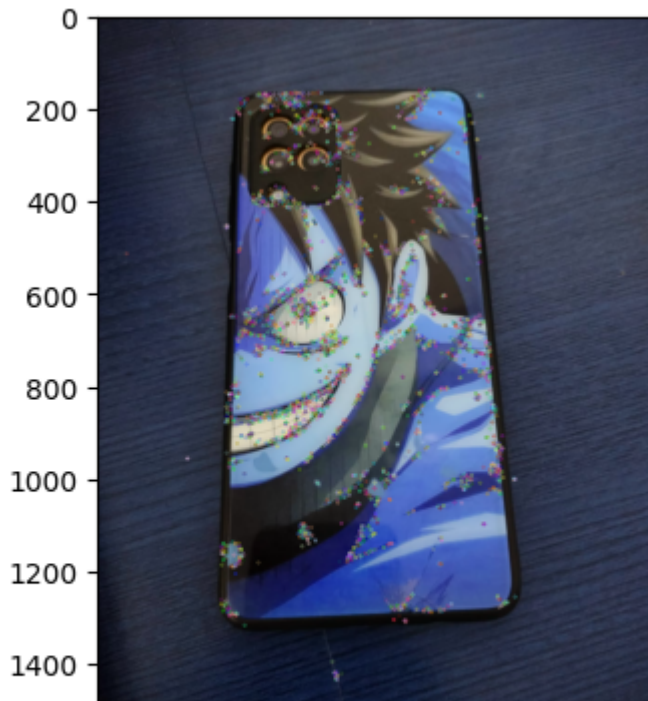
      [[23, 25, 35],
       [22, 24, 34],
       [22, 23, 33],
       ...,
       [37, 39, 57],
       [37, 41, 60],
       [39, 43, 62]],

      [[21, 23, 33],
       [21, 23, 33],
       [21, 22, 32],
```

```
...,  
[40, 42, 60],  
[41, 45, 64],  
[43, 47, 66]]], dtype=uint8)
```

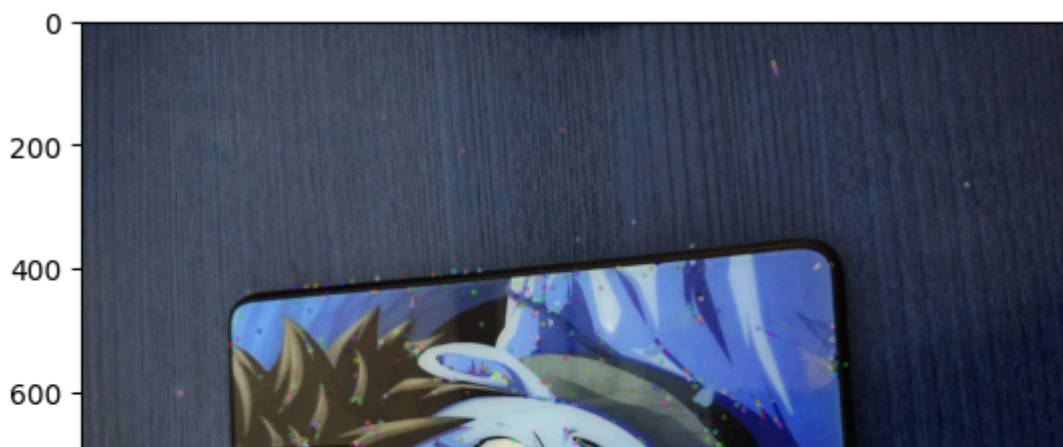
```
sift= cv2.xfeatures2d.SIFT_create()  
keypoints_1, descriptors_1 = sift.detectAndCompute(image1,None)  
img_1 = cv2.drawKeypoints(image1,keypoints_1,image1)  
plt.imshow(img_1)
```

<matplotlib.image.AxesImage at 0x78f087c3cd60>



```
keypoints_2, descriptors_2 = sift.detectAndCompute(image2,None)  
img_2 = cv2.drawKeypoints(image2,keypoints_2,image2)  
plt.imshow(img_2)
```

<matplotlib.image.AxesImage at 0x78f087e23fa0>



```
len(keypoints_1),len(keypoints_2)
```

```
(616, 836)
```

```
#Feature matching
```

```
bf=cv2.BFMatcher(cv2.NORM_L1,crossCheck=True)
```

```
matches=bf.match(descriptors_1,descriptors_2)
```

```
matches=sorted(matches, key= lambda x:x.distance)
```

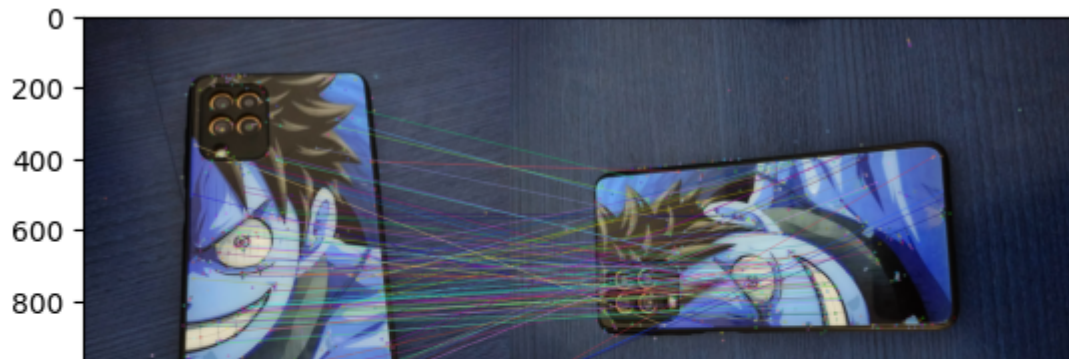
```
len(matches)
```

```
380
```

```
img3=cv2.drawMatches(img_1,keypoints_1,img_2,keypoints_2,matches[:100],img_2,flags=2)
```

```
plt.imshow(img3)
```

```
<matplotlib.image.AxesImage at 0x78f087e2d600>
```



```
#Extract matches keypoints
```

```
src_pts=np.float32([keypoints_1[m.queryIdx].pt for m in matches]).reshape(-1,1,2)
```

```
dst_pts=np.float32([keypoints_2[m.trainIdx].pt for m in matches]).reshape(-1,1,2)
```

```
#perspective transformation
```

```
M, _ = cv2.findHomography(src_pts, dst_pts, cv2.RANSAC,5.0)
```

```
M
```

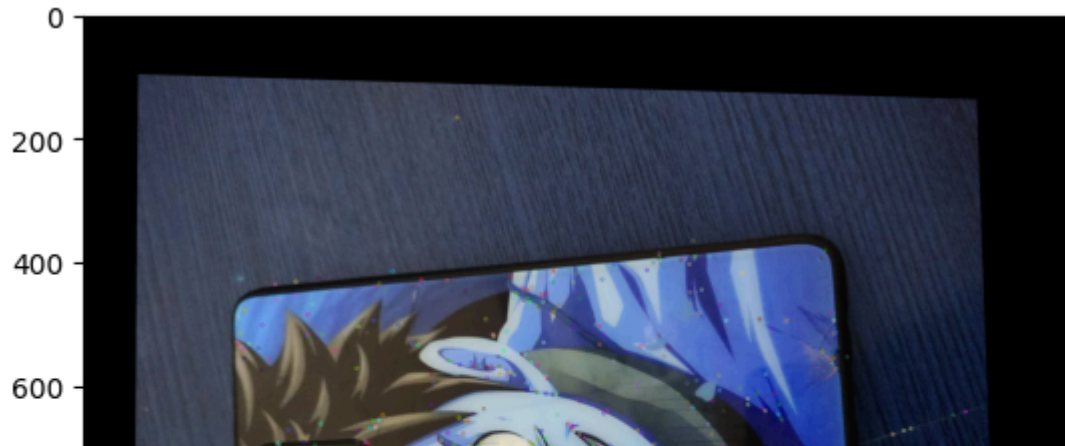
```
array([[ 8.95003181e-03,  1.01566355e+00,  8.14430169e+01],
       [-9.00698782e-01,  3.91399081e-02,  1.18070878e+03],
       [ 2.88499994e-05,  9.42306110e-05,  1.00000000e+00]])
```

```
#apply prespective transformation
```

```
newImage1 = cv2.warpPerspective(img_1, M, dsize=(img_2.shape[1], img_2.shape[0]))
```

```
plt.imshow(newImage1)
```

```
<matplotlib.image.AxesImage at 0x78f088d39a50>
```



```
plt.figure(figsize=(15, 5))
```

```
plt.subplot(1, 3, 1)
```

```
plt.imshow(image1, cmap='gray')
```

```
plt.title('Original Image')
```

```
plt.subplot(1, 3, 2)
```

```
plt.imshow(image2, cmap='gray')
```

```
plt.title('Ref')
```

```
plt.subplot(1, 3, 3)
```

```
plt.imshow(newImage1)
```

```
plt.title('Registered')
```

```
Text(0.5, 1.0, 'Registered')
```

