Name: Rishikesh Vadodaria

Roll no: C114

Batch : C2

Experiment 5

```
import cv2 as cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
import cv2 as cv2

image_path = '/content/TajMahal.jpg'
image = cv2.imread(image_path)

image1 = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image2 = image1.copy()
image3 = image1.copy()
image4 = image1.copy()
```

```
plt.imshow(image1)
```

```
<matplotlib.image.AxesImage at 0x7c23a449fc50>
```

```
sift = cv2.xfeatures2d.SIFT_create(contrastThreshold=0.09)
k1, d1 = sift.detectAndCompute(image1, None)
```

```
d1
```

```
array([[14., 25., 53., ...,  0.,  0.,  0.],
       [21., 42.,  9., ...,  1.,  6.,  3.],
       [ 2.,  0.,  0., ...,  5.,  7., 54.],
       ...,
       [45., 14.,  0., ...,  1.,  0.,  2.],
       [44., 20.,  3., ...,  3.,  0.,  2.],
       [ 5.,  0.,  0., ..., 86.,  4.,  2.]], dtype=float32)
```

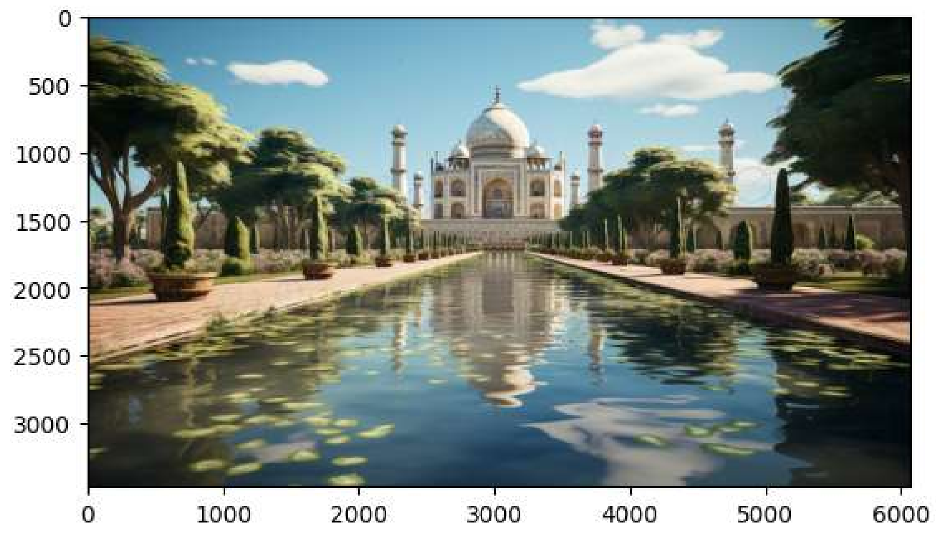
```
len(k1)
```

```
21923
```

```
keyimg = cv2.drawKeypoints(image1, k1 , image1, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

```
plt.imshow(keyimg)
```

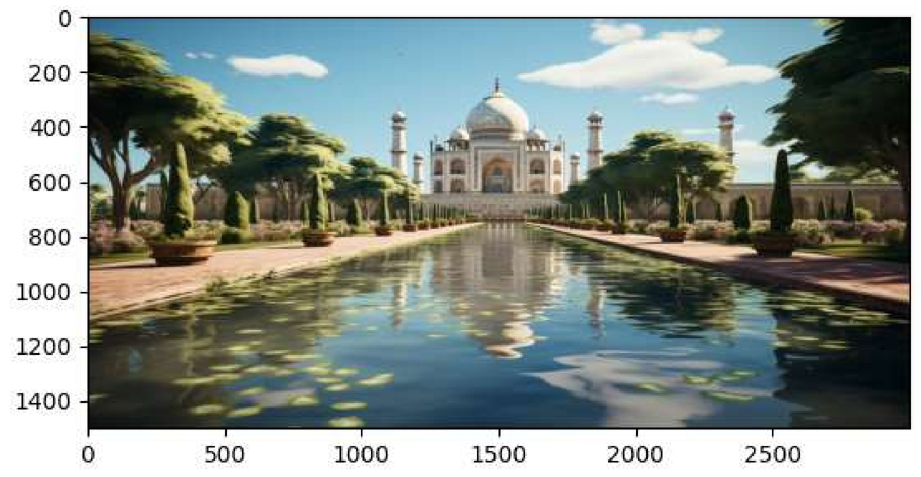
```
<matplotlib.image.AxesImage at 0x7c23a448d650>
```



Resize Image

```
image2 = cv2.resize(image2, (3000, 1500))
plt.imshow(image2)
```

```
<matplotlib.image.AxesImage at 0x7c239707dc90>
```



```
k2, d2 = sift.detectAndCompute(image2, None)
```

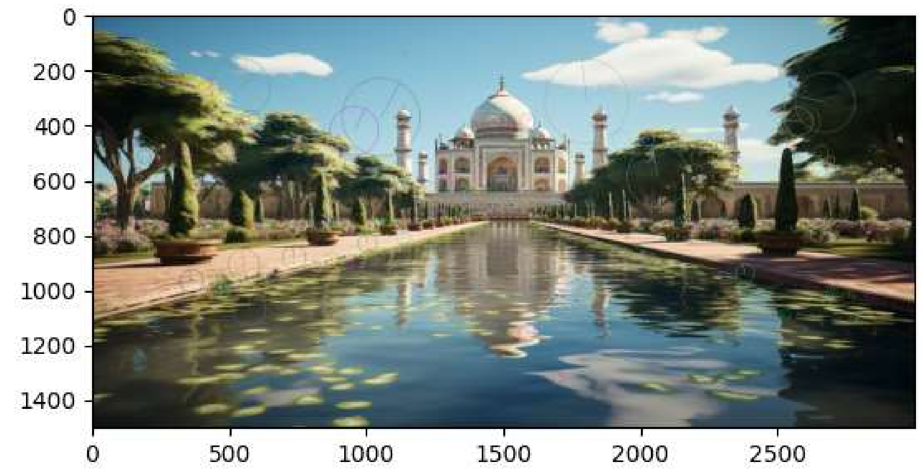
```
d2
len(k2)
```

```
9835
```

```
keyimg2 = cv2.drawKeypoints(image2, k2 , image2, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

```
plt.imshow(keyimg2)
```

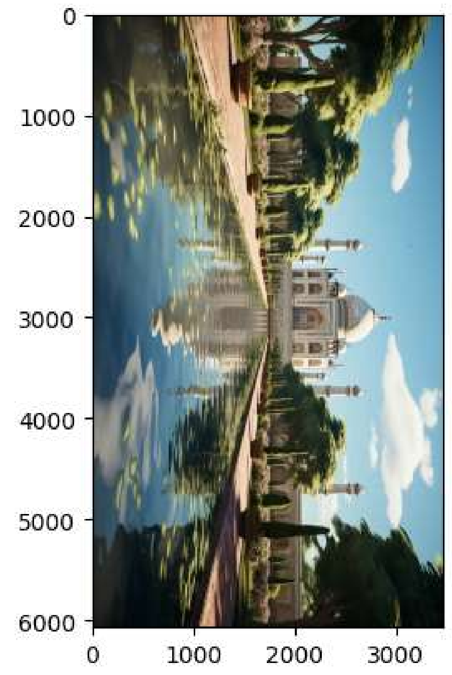
```
<matplotlib.image.AxesImage at 0x7c23973127d0>
```



Rotate Image

```
image3 = cv2.rotate(image3, cv2.ROTATE_90_CLOCKWISE)
plt.imshow(image3)
```

```
<matplotlib.image.AxesImage at 0x7c23885fbad0>
```

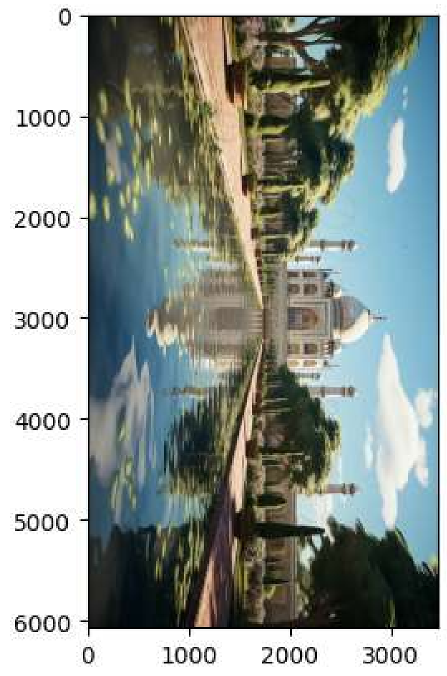


```
k3, d3 = sift.detectAndCompute(image3, None)
d3
len(k3)
```

```
21914
```

```
keyimg3 = cv2.drawKeypoints(image3, k3 , image3, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
plt.imshow(keyimg3)
```

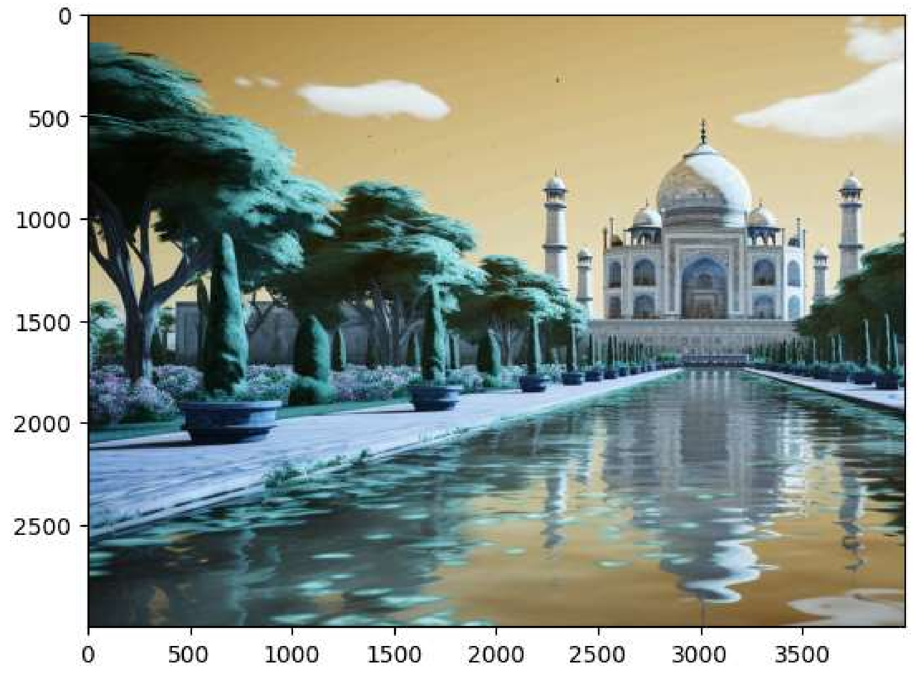
```
<matplotlib.image.AxesImage at 0x7c238857dc90>
```



Crop

```
image4 = image[0:3000, 0:4000]
plt.imshow(image4)
```

```
<matplotlib.image.AxesImage at 0x7c23885adc90>
```



```
k4, d4 = sift.detectAndCompute(image4, None)
d4
len(k4)
```

```
15736
```

```
keyimg4 = cv2.drawKeypoints(image4, k4 , image4, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
plt.imshow(keyimg4)
```

```
<matplotlib.image.AxesImage at 0x7c2388325510>
```

```
bf = cv2.BFMatcher()

matches1_2 = bf.knnMatch(d1, d2, k=2)
matches1_3 = bf.knnMatch(d1, d3, k=2)
matches1_4 = bf.knnMatch(d1, d4, k=2)

# Sorting matches based on distance
matches1_2 = sorted(matches1_2, key=lambda x: x[0].distance)
matches1_3 = sorted(matches1_3, key=lambda x: x[0].distance)
matches1_4 = sorted(matches1_4, key=lambda x: x[0].distance)

```

```
# Extract only the best match from each knnMatch result
best_matches1_2 = [m[0] for m in matches1_2[:10]]
best_matches1_3 = [m[0] for m in matches1_3[:10]]
best_matches1_4 = [m[0] for m in matches1_4[:10]]

# Draw the matches
image1_2_match = cv2.drawMatches(image1, k1, image2, k2, best_matches1_2, None)
image1_3_match = cv2.drawMatches(image1, k1, image3, k3, best_matches1_3, None)
image1_4_match = cv2.drawMatches(image1, k1, image4, k4, best_matches1_4, None)

```

```
image1_2_match, image1_3_match , image1_4_match
```

```
        [[ 50, 100, 135],
         [ 50, 100, 135],
         [ 47,  99, 136],
         ...,
         [205, 181, 123],
         [205, 181, 123],
         [205, 181, 123]],

        [[ 50, 100, 135],
         [ 47,  99, 136],
         [ 47,  99, 136],
         ...,
         [205, 181, 123],
         [205, 181, 123],
         [205, 181, 123]],

        ...,

        [[ 10,  19,  18],
         [ 11,  20,  19],
         [ 11,  20,  19],
         ...,
         [  0,   0,   0],
         [  0,   0,   0],
         [  0,   0,   0]],

        [[ 11,  20,  19],
         [ 11,  20,  19],
         [ 11,  20,  19],
         ...,
         [  0,   0,   0],
         [  0,   0,   0],
         [  0,   0,   0]],

        [[ 11,  20,  19],
         [ 11,  20,  19],
         [ 11,  20,  19],
         ...,
         [  0,   0,   0],
         [  0,   0,   0],
         [  0,   0,   0]]], dtype=uint8))
```

Conclusion

The given image **Tajmahal.jpg** was analyzed for the detection of keypoints and descriptors using the **SIFT (Scale-Invariant Feature Transform)** algorithm, identifying **21,923 keypoints** in total. These keypoints represent distinct and highly recognizable features in the image, making them useful for tasks such as image matching, recognition, and stitching. The extracted descriptors encode crucial information about the local structure around each keypoint, ensuring robustness to scale, rotation, and lighting variations.