Computer-Vision-TA2

Name : Utkarsh Shahare

Roll No: 58

Sem : 6th

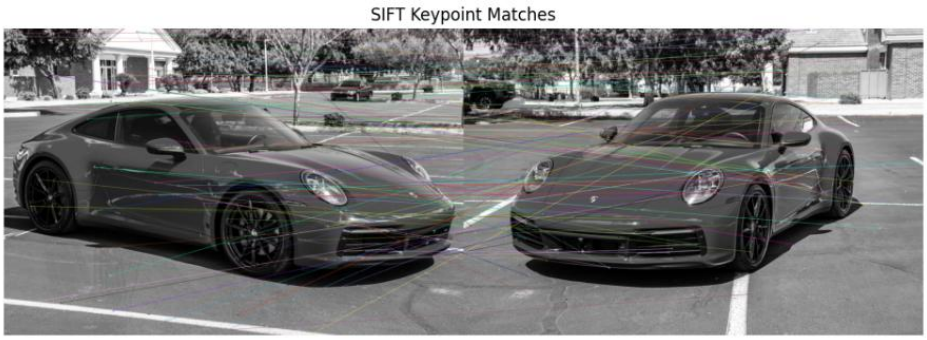
Branch : CSE – DS

Section : B4

ScreenShots :

1. Implement the SIFT algorithm to detect and match key points between two images.

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


# Load the two images (grayscale)
img1 = cv2.imread('image1.jpg', cv2.IMREAD_GRAYSCALE)
img2 = cv2.imread('image2.jpg', cv2.IMREAD_GRAYSCALE)

# Initialize SIFT detector
sift = cv2.SIFT_create()

# Detect keypoints and compute descriptors
kp1, des1 = sift.detectAndCompute(img1, None)
kp2, des2 = sift.detectAndCompute(img2, None)

# Use BFMatcher to match descriptors
bf = cv2.BFMatcher()
matches = bf.knnMatch(des1, des2, k=2)

# Apply Lowe's ratio test
good_matches = []
for m, n in matches:
    if m.distance < 0.75 * n.distance:
        good_matches.append(m)

# Draw matches
result_img = cv2.drawMatches(img1, kp1, img2, kp2, good_matches, None, flags=2)

# Show image using matplotlib
plt.figure(figsize=(12, 6))
plt.imshow(result_img)
plt.title("SIFT Keypoint Matches")
plt.axis("off")
plt.show()
```



4. Implement the Harris corner detector to find and visualize corners in a grayscale image.

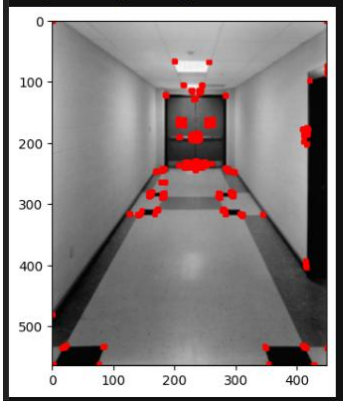
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

```
img = cv2.imread('corner.jpg')
img_rgb=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
gray_img = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
```

```
dst = cv2.cornerHarris(gray_img,2, 5, 0.04)
dst = cv2.dilate(dst, None,iterations=3)
```

```
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
thresh = 0.01 * dst.max()
corner_img = np.copy(img)
for i in range(0, dst.shape[0]):
    for j in range(0, dst.shape[1]):
        if dst[i, j] > thresh:
            cv2.circle(corner_img, (j, i), 1, (255, 0, 0), 1)
plt.imshow(corner_img)
```

```
<matplotlib.image.AxesImage at 0x258fdaef7d0>
```



5. Use the Shi-Tomasi corner detector to identify and mark corner points in an image.

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

```
# Load the image
img = cv2.imread('corner.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect corners
corners = cv2.goodFeaturesToTrack(gray, maxCorners=100, qualityLevel=0.01, minDistance=10)
corners = np.int0(corners)

# Draw the corners
for corner in corners:
    x, y = corner.ravel()
    cv2.circle(img, (x, y), 4, (0, 255, 0), -1)

# Convert BGR to RGB for matplotlib
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Display using matplotlib
plt.figure(figsize=(8, 6))
plt.imshow(img_rgb)
plt.title("Shi-Tomasi Corner Detection")
plt.axis("off")
plt.show()
```

```
C:\Users\utkar\AppData\Local\Temp\ipykernel_11136\938845870.py:7: DeprecationWarning: `np.i
1.24)
  corners = np.int0(corners)
```