CourseNa m Computer VisionLab **Course Code:** CSP-422

Experiment:1.3

Aim:

Write a program to analyze the impact of refining feature detection for image segmentation.

Software Required:

- PrPython
- OpenCV (cv2)
- NumPy (for array operations)

Description:

Image segmentation is the process of partitioning an image into multiple segments, where each segment represents a distinct object or region in the image. Feature detection is an essential step in this process as it helps identify the regions or objects of interest within an image.

Pseudo code/Algorithms/Flowchart/Steps:

Load the target image and reference images.

- Load the input image.
- Preprocess the image if needed (e.g., resize, denoise, or normalize).
- Apply a feature detection algorithm (e.g., edge detection, contour detection, or color-based segmentation) to identify relevant features.
- Refine the detected features if necessary (e.g., filter out noise or enhance feature boundaries).
- Perform image segmentation using the refined features.
- Post-process the segmentation results (e.g., fill holes, remove small segments, or apply morphological operations).
- Visualize and/or save the segmented image.

CourseName: Computer Vision Lab **Course Code:** CSP-422

Code:-

```
# EXPERIMENT 3
import cv2
import numpy as np


# Load target object image
target_image = cv2.imread('target_image.png', cv2.IMREAD_GRAYSCALE)


# Load dataset images
dataset_images = ['image1.png', 'image2.png', 'image3.png', 'image4.png']


# Initialize SIFT detector
sift = cv2.SIFT_create()


# Extract features from the target object
keypoints_target, descriptors_target = sift.detectAndCompute(target_image, None)


# Loop through dataset images
for image_path in dataset_images:
    # Load dataset image
    dataset_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)


    # Extract features from dataset image
    keypoints_dataset, descriptors_dataset = sift.detectAndCompute(dataset_image,
None)
```

CourseName: Computer Vision Lab **Course Code:** CSP-422

```
# Create BFMatcher object
bf = cv2.BFMatcher()

# Match features between target and dataset images
matches = bf.knnMatch(descriptors_target, descriptors_dataset, k=2)

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

# Calculate matching score based on the number of good matches
matching_score = len(good_matches) / len(keypoints_target)

# Print matching score for each image
print(f"{image_path} - Matching Score: {matching_score:.2f}")
```

Implementation:

Screenshot:

CourseName: Computer Vision Lab **Course Code:** CSP-422

Experiment_1.3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share R

Files

+ Code + Text

RAM Disk

Experiment: 1.3

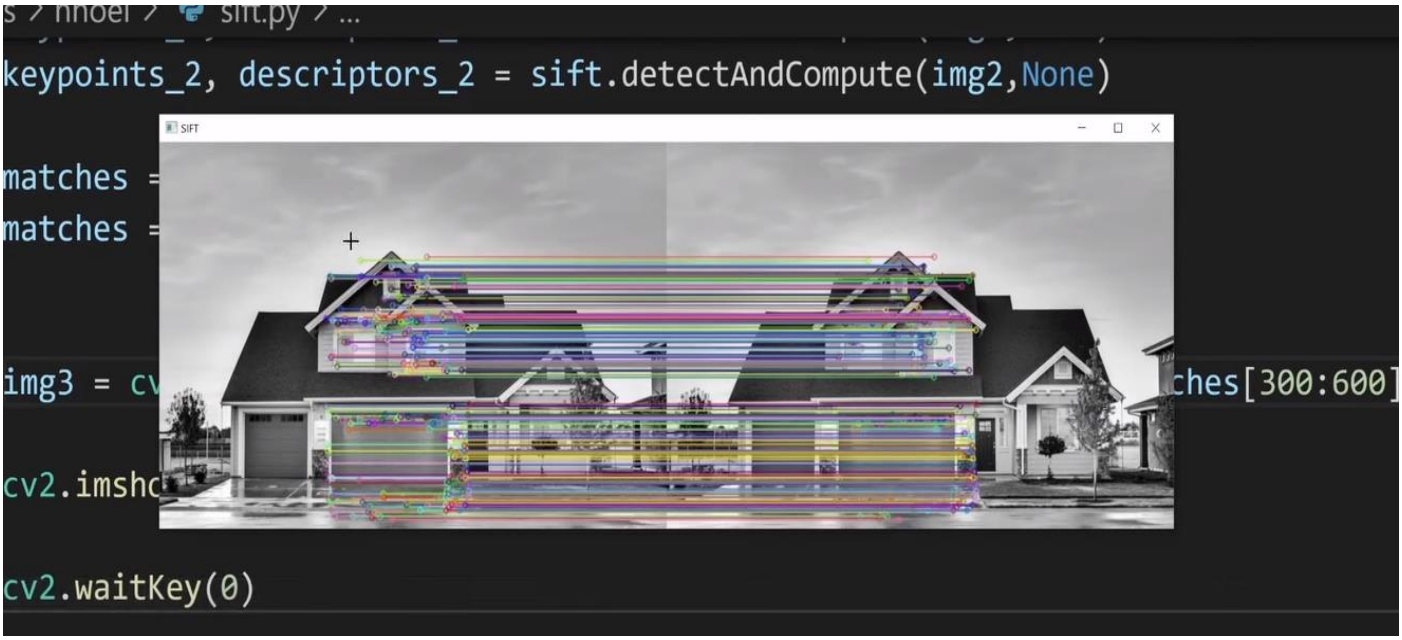
Write a program to analyze the impact of refining feature detection for image segmentation

```
[ ]

[1] import cv2
    import numpy as np

[12] target_image = cv2.imread('/content/Image1.jpeg', cv2.IMREAD_GRAYSCALE)
     dataset_images = ['/content/Image1.jpeg', '/content/image2.jpeg', '/content/image3.jpeg', '/content/image4.jpeg']

sift = cv2.SIFT_create()

[13] keypoints_target, descriptors_target = sift.detectAndCompute(target_image, None)

[8] for image_path in dataset_images:
        dataset_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
        keypoints_dataset, descriptors_dataset = sift.detectAndCompute(dataset_image, None)

        # Rest of the steps will go here...

    bf = cv2.BFMatcher()
    matches = bf.knnMatch(descriptors_target, descriptors_dataset, k=2)

    good_matches = []
    for m, n in matches:
        if m.distance < 0.75 * n.distance:
            good_matches.append(m)
```

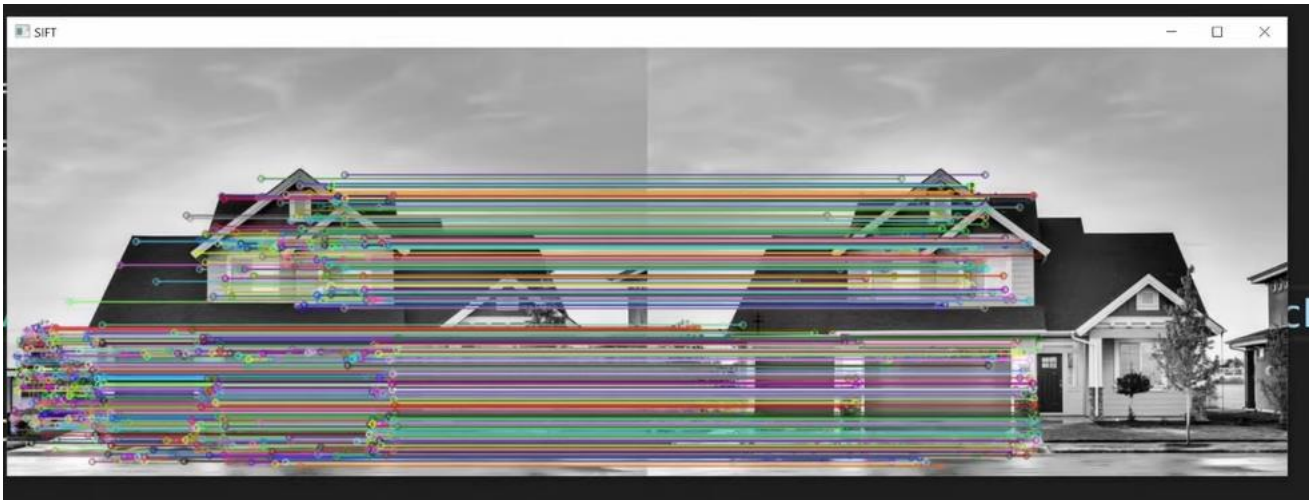
Files: .. .config sample_data Image1.jpeg image2.jpeg image3.jpeg image4.jpeg

Disk 81.47 GB available

Output:



CourseName: Computer Vision Lab **Course Code:** CSP-422



Output:

This program will display the segmented image with detected contours. You can further refine and customize the feature detection and segmentation steps based on your specific requirements..