Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Machine Problem No. 4			
Topic:	**Module 2.0: Feature Extraction and Object Detection**	**Week No.**	6-7
Course Code:	**CSST106**	**Term:**	1st Semester
Course Title:	**Perception and Computer Vision**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date		**Points**	

Machine Problem No. 4: Feature Extraction and Image Matching in Computer Vision

Overview:
In this machine problem, you will implement various feature extraction and matching algorithms to process, analyze, and compare different images. You will utilize techniques such as SIFT, SURF, ORB, HOG, and Harris Corner Detection to extract keypoints and descriptors. You will also perform feature matching between pairs of images using the FLANN and Brute-Force matchers. Finally, you will explore image segmentation using the Watershed algorithm.

Objectives:
1. To apply different feature extraction methods (SIFT, SURF, ORB, HOG, Harris Corner Detection).
2. To perform feature matching using Brute-Force and FLANN matchers.
3. To implement the Watershed algorithm for image segmentation.
4. To visualize and analyze keypoints and matches between images.
5. To evaluate the performance of different feature extraction methods on different images.

Problem Statement:
You are tasked with building a Python program using OpenCV and related libraries to accomplish the following tasks. Each task should be implemented in a separate function, with appropriate comments and visual outputs. You will work with images provided in your directory or chosen from any online source.

Tasks:
Task 1: Harris Corner Detection
- Load any grayscale image.
- Apply the Harris Corner Detection algorithm to find corners.
- Display the original image and the image with detected corners marked in red.

Function signature: def harris_corner_detection(image_path):

Task 2: HOG Feature Extraction

- Load an image of a person (or another object).
- Convert the image to grayscale.
- Extract HOG (Histogram of Oriented Gradients) features from the image.
- Display the original image and the visualized HOG features.

Function signature: def hog_feature_extraction(image_path):

Task 3: ORB Feature Extraction and Matching

- Load two different images.
- Apply ORB (Oriented FAST and Rotated BRIEF) to detect and compute keypoints and descriptors for both images.
- Use the FLANN-based matcher to match the ORB descriptors of the two images.
- Visualize the matching keypoints between the two images.

Function signature: def orb_feature_matching(image_path1, image_path2):

Task 4: SIFT and SURF Feature Extraction

- Load two images of your choice.
- Apply both SIFT and SURF algorithms to detect keypoints and compute descriptors.
- Visualize the keypoints detected by both methods in two separate images.

Function signature: def sift_and_surf_feature_extraction(image_path1, image_path2):

Task 5: Feature Matching using Brute-Force Matcher

- Load two images and extract ORB descriptors.
- Match the descriptors using the Brute-Force Matcher.
- Display the matched keypoints between the two images.

Function signature: def brute_force_feature_matching(image_path1, image_path2):

Task 6: Image Segmentation using Watershed Algorithm

- Load any image of your choice.
- Convert the image to grayscale and apply a threshold to separate foreground from the background.
- Apply the Watershed algorithm to segment the image into distinct regions.
- Display the segmented image.

Function signature: def watershed_segmentation(image_path):

Instructions:

1. **Input**:
   - You will be provided with two or more images for each task, or you can select images of your own.
   - Each image should be loaded, processed, and then displayed with appropriate markings (e.g., keypoints, segments, or matches).
2. **Output**:
   - Display the original images and results for each task (e.g., corner detection, feature extraction, matching, segmentation).
   - Ensure that each result is appropriately labeled (title and axis for the plot if needed).
3. **Implementation Guidelines**:
   - Implement each task as a separate function in Python.
   - Use OpenCV for image processing and feature extraction tasks.
   - Use Matplotlib to visualize and display the results.
   - Include proper comments to explain your code.

Submission:

- A PDF or markdown document (performance_analysis.pdf or performance_analysis.md).

Submission Guidelines:

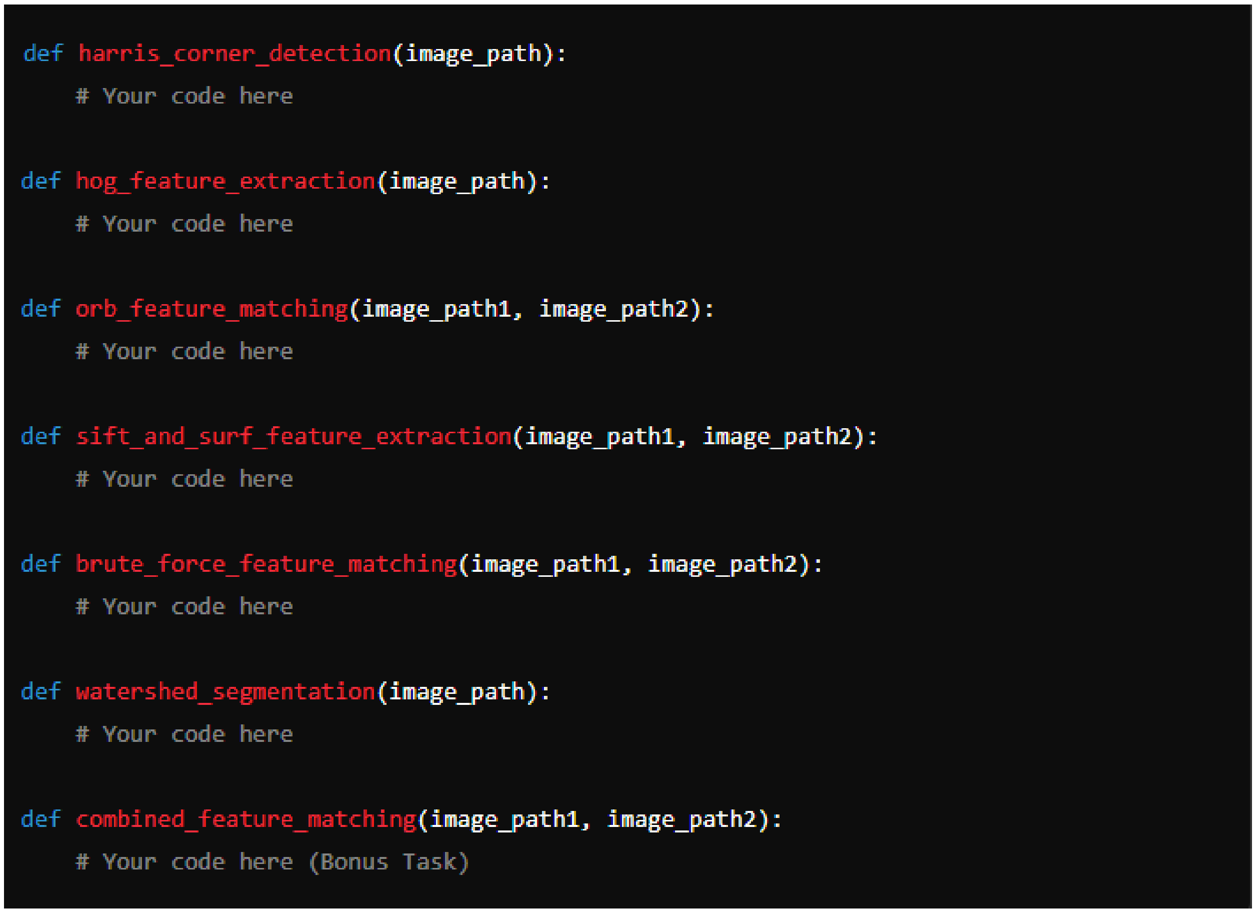
- **GitHub Repository**:
  - Create a folder in your CSST106-Perception and Computer Vision repository named Feature-Extraction-Machine-Problem.
  - Upload all code, images, and reports to this folder.

- **File Naming Format**: **[SECTION-LASTNAME-MP4]** 4D-LASTNAME-MP4
  - 4D-BERNARDINO-SIFT.py
  - 4D-BERNARDINO-Matching.jpg

Additional Penalties:

- **Incorrect Filename Format**: -5 points
- **Late Submission**: -5 points per day
- **Cheating/Plagiarism**: Zero points for the entire task

Sample Function Signatures



```python
def harris_corner_detection(image_path):
    # Your code here

def hog_feature_extraction(image_path):
    # Your code here

def orb_feature_matching(image_path1, image_path2):
    # Your code here

def sift_and_surf_feature_extraction(image_path1, image_path2):
    # Your code here

def brute_force_feature_matching(image_path1, image_path2):
    # Your code here

def watershed_segmentation(image_path):
    # Your code here

def combined_feature_matching(image_path1, image_path2):
    # Your code here (Bonus Task)
```

Rubric for Advanced Feature Extraction and Image Processing

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (60-74%)	Needs Improvement (0-59%)
Task 1: Harris Corner Detection	Clear detection and visualization	Minor issues in visualization	Basic implementation	Incorrect or no solution
Task 2: HOG Feature Extraction	Clear visualization and explanation	Minor issues in feature extraction	Basic explanation	Incorrect or no feature extraction
Task 3: ORB Feature Matching	Correct matching and visualization	Minor issues with matching	Basic attempt	Incorrect matching
Task 4: SIFT and SURF Extraction	Accurate feature detection	Minor issues with feature comparison	Basic implementation	Incorrect or no comparison
Task 5: Brute-Force Feature Matching	Clear visualization and comparison	Minor issues in visualization	Basic attempt	Incorrect matching
Task 6: Watershed Segmentation	Correct segmentation and result	Minor issues in segmentation	Basic segmentation	Incorrect or no segmentation