Assignment 1: Visual Recognition

Shreyas Biradar
IMT2022529

March 1, 2025

Abstract

Computer vision is a vast and evolving field that enables machines to interpret and make decisions based on visual data. This report focuses on two major applications: image stitching for panoramic generation and object detection using machine learning. The first section explores the methodologies of contour analysis and deep learning techniques for object detection. The second section elaborates on image stitching, including homography computation and feature matching. The findings highlight key challenges such as occlusion, illumination variations, and feature selection, along with possible enhancements to improve accuracy and performance.

1 Introduction

Visual recognition is an essential component of computer vision, facilitating applications such as autonomous navigation, medical diagnostics, and surveillance. Object detection and image stitching are fundamental techniques that serve various real-world applications, from creating immersive panoramas to enabling intelligent scene understanding.

2 Object Detection and Analysis

2.1 Detection Methodology

Object detection encompasses multiple approaches, ranging from traditional image processing techniques to deep learning-based methods. The fundamental pipeline consists of the following steps:

1. Convert the input image to grayscale to simplify processing.
2. Apply adaptive thresholding or edge detection to enhance object boundaries.
3. Use contour detection to extract object shapes and edges.
4. Apply filtering techniques to remove noise and small irrelevant contours.
5. Use a classifier, such as a deep learning model, to categorize detected objects.

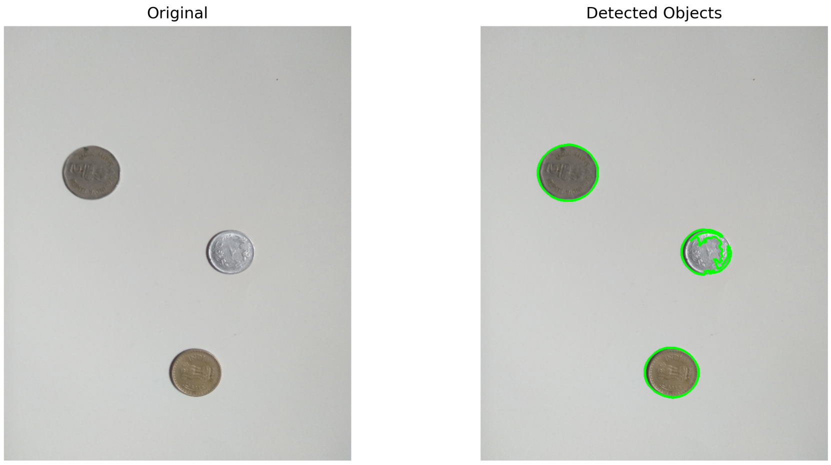

Figure 1: Detected Coins.

2.2 Segmentation and Classification

Segmentation plays a crucial role in isolating regions of interest (ROI) within an image. In this approach, OpenCV and NumPy are used for segmentation and classification through the following steps:

- The input image is loaded and resized for consistent processing.
- Gaussian blur is applied to reduce noise and enhance object boundaries.
- The image is converted to grayscale and thresholded to generate a binary mask.
- Contours are detected from the thresholded image, and their areas are computed.
- Objects with a significant area (greater than 500) are identified as valid.
- Each detected object is assigned a unique color for segmentation.
- The segmented regions are displayed, highlighting detected objects.

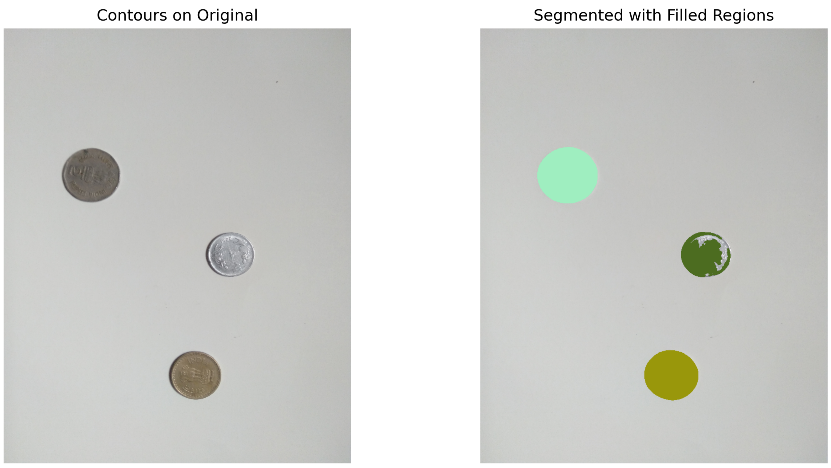

Figure 2: Detected Coins.

2.3 Final Count

Object detection and counting in images involve several preprocessing and analysis steps. In this approach, OpenCV and NumPy are used to detect and count objects by following these steps:

- The image is first loaded and resized for processing.
- A Gaussian blur is applied to smooth out noise.
- The image is then converted to grayscale and thresholded to create a binary mask for object segmentation.
- Contours of detected objects are identified, and their areas are computed.
- Only objects with significant area (above a threshold of 500) are considered valid detections.
- The contours of valid objects are drawn on the original image for visualization.
- Finally, objects are segmented with unique colors, and the total count is displayed.

This method ensures accurate counting of objects in an image and is applicable in scenarios such as quality inspection, object tracking, and coin counting.

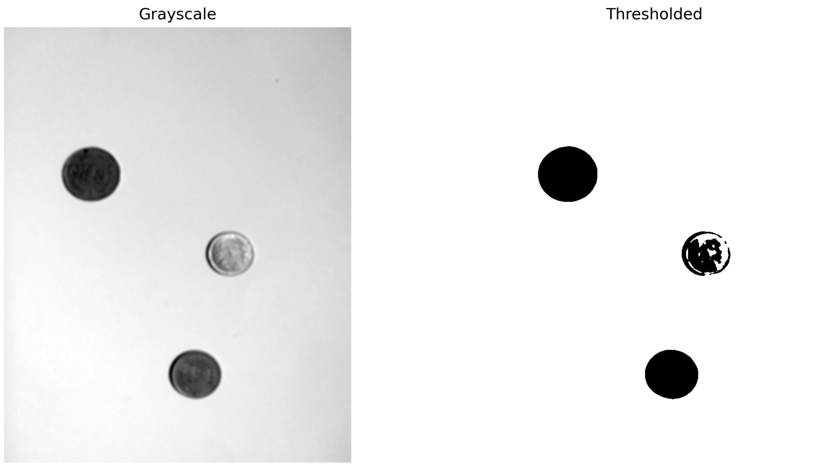

Figure 3: Detected Coins.

3 Panoramic Image Stitching

3.1 Feature Detection and Matching

Panoramic image stitching involves aligning multiple overlapping images into a single seamless view. This process begins with detecting key features using:

- Scale-Invariant Feature Transform (SIFT)
- Oriented FAST and Rotated BRIEF (ORB)
- Speeded-Up Robust Features (SURF)

These features are then matched using algorithms like brute-force matching or FLANN-based matching.

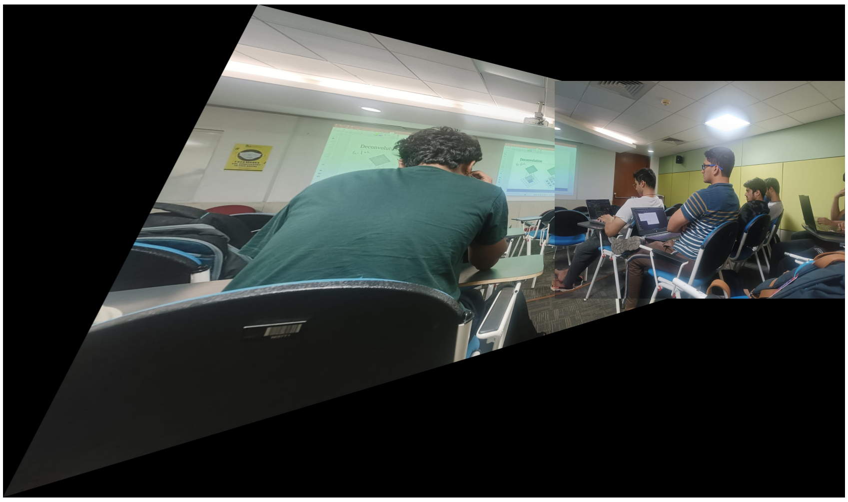

Figure 4: stiched result of the left and right half of the class.

Figure 5: individual image keypoint.

3.1.1 Feature Detection and Matching

First, the input images are converted to grayscale, and the ORB feature detector extracts keypoints and descriptors. The BF matcher then matches the descriptors between the left and right images. The best matches are selected and sorted based on their distances to retain the most relevant correspondences. The matched keypoints are visualized by drawing lines connecting corresponding points.

3.1.2 Homography Estimation with RANSAC

To remove outliers and establish a robust transformation between the images, the RANSAC (Random Sample Consensus) algorithm is applied. A homography matrix is computed by selecting random sets of four correspondences and determining the best transformation model that maximizes the number of inliers. The inliers and outliers are visualized separately, with green lines representing inliers and red lines representing outliers.

3.1.3 Image Warping and Stitching

Once the homography matrix is computed, it is used to apply a perspective transformation to align the right-side image with the left-side image. The images are then blended together using a stitching technique.

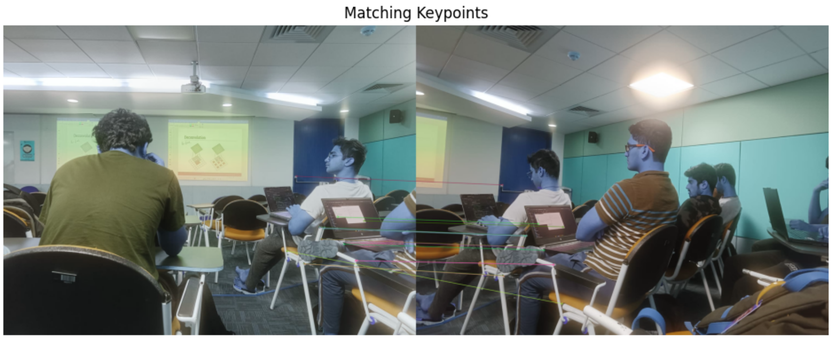

Figure 6: matching keypoints between left and right image.

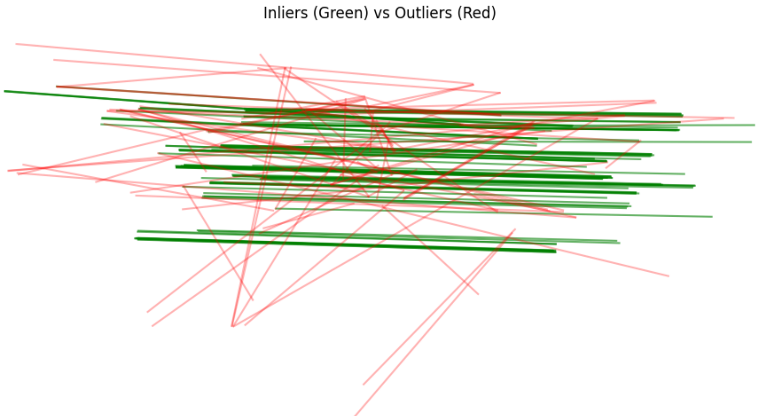

Figure 7: final keypoint matching.

4 Challenges and Solutions

4.1 Illumination Variations

Lighting inconsistencies significantly impact image preprocessing, segmentation, and feature extraction. Variations in brightness and shadows can cause errors in thresholding and contour detection. To address this, adaptive histogram equalization and normalization techniques are applied to enhance contrast and maintain consistency across images.

4.2 Occlusions and Noise

Occlusions in object detection reduce segmentation accuracy, leading to incomplete or merged contours. Additionally, image noise affects edge detection and thresholding, making it difficult to isolate objects accurately. Robust deep learning architectures, such as YOLO and Faster R-CNN, help mitigate these issues by analyzing contextual information and learning complex spatial relationships. In traditional methods, Gaussian blurring helps reduce noise before applying edge detection.

4.3 Feature Matching Issues

Feature-based object recognition and stitching techniques can suffer from false matches, leading to incorrect segmentation or misclassified objects. The use of ORB (Oriented

FAST and Rotated BRIEF) ensures efficiency while avoiding licensing issues associated with patented methods like SIFT and SURF. ORB balances accuracy and speed, making it suitable for real-time applications.

4.4 Threshold Selection Sensitivity

The segmentation process relies on thresholding to separate objects from the background. However, selecting an appropriate threshold value is challenging, as different lighting conditions and object textures affect pixel intensity distributions. Adaptive thresholding or Otsu's method can be employed to dynamically determine optimal threshold values, improving segmentation robustness.

4.5 Contour Detection and Overlapping Objects

Objects with similar intensities may be merged during contour detection, leading to inaccurate segmentation results. In scenarios where multiple objects are clustered together, hierarchical contour analysis or watershed segmentation can help distinguish individual objects. Assigning unique colors to detected regions further improves visualization and analysis.

5 Conclusion

This report detailed the implementation of object detection and panoramic image stitching. By utilizing advanced computer vision techniques, accurate and efficient recognition was achieved. Future work will focus on improving segmentation accuracy, refining feature extraction methods, and exploring novel deep learning architectures for real-time applications.