

Visual Recognition Assignment - 1

Abhik Kumar
IMT2022117

1 README

1.1 How to Run the Code

1. First, clone the GitHub repository containing the necessary scripts: Ensure that your current working directory (cwd) is set to the cloned repository before running any script.
2. Install the necessary dependencies using the following command:

```
pip install opencv-python numpy matplotlib
```

3. Then, execute the respective scripts as follows:

1.1.1 Part 1: Coin Detection and Segmentation

```
python coin_detection.py
```

This script will create a folder named `coin_detection_outputs`, which will contain all the required images.

1.1.2 Part 2: Panorama Stitching

```
python image_stitching.py
```

This script will create a folder named `image_stitching_outputs`, which will contain all the required images.

1.2 Directory Structure

```
assignment_1/  
  coin_detection.py  
  image_stitching.py  
  coins.png  
  img1.png  
  img2.png  
  img3.png  
  coin_detection_results/  
  stitched_outputss/
```

2 Introduction

The assignment contains two tasks: coin detection and segmentation, and panorama stitching. In the first part we detect and segment coins in an image, while in the second part we stitch multiple images into a single panorama.

3 Part 1: Coin Detection and Segmentation

3.1 Input Image

The input image is of scattered Indian coins:



Figure 1: Input Image

3.2 Explanation of Methods

- **CLAHE (Contrast Limited Adaptive Histogram Equalization):** CLAHE enhances the contrast of the grayscale image by redistributing pixel intensity. This improves the visibility of the coins.
- **Hough Circle Transform:** Hough circle transform detects circular shapes by transforming the image into a parameter space where circles can be identified based on the edge points.
- **Edge Detection:** Canny edge detection algorithm is used for edge detection. It applies gradient operator and filter weak edges based on a threshold.

- **Region-Based Segmentation:** Watershed algorithm is used to separate individual coins from the background.
- **Counting Coins:** The contours of segmented regions are detected and counted to calculate the number of coins.

3.3 Parameter Optimization

The parameters were selected through an iterative process by adjusting the sliders in a Streamlit interface.

1. **CLAHE:** To prevent excessive noise, the clip limit and tile grid size were fine tuned.
2. **Hough Circle Detection:** Several parameters were modified to ensure that circles representing coins were detected correctly.
3. **Canny Edge Detection:** The threshold values were tuned to balance detecting edges.

3.4 Final Results



Figure 2: CLAHE Enhanced Grayscale Image



Figure 3: Coins Detected by Hough Circles

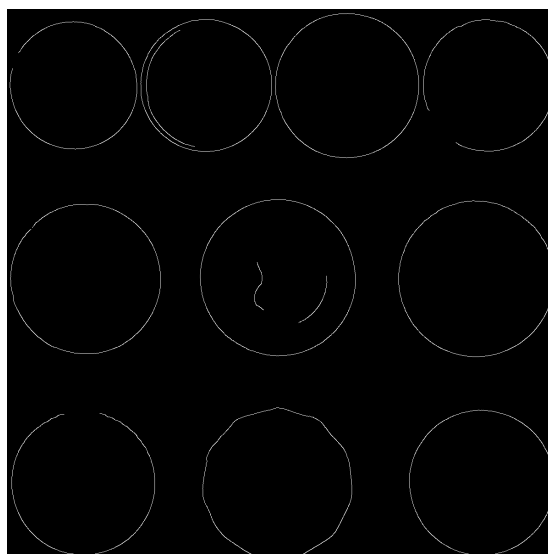


Figure 4: Canny Edge Detection

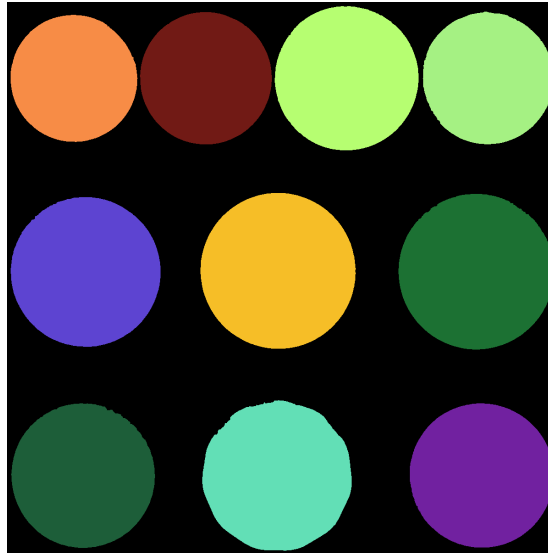


Figure 5: Color Based Segmentation



Figure 6: Individual Segmented Coins

4 Part 2: Panorama Stitching

4.1 Input Images

Three overlapping images are taken as input for panorama stitching.



Figure 7: Input Image 1



Figure 8: Input Image 2



Figure 9: Input Image 3

4.2 Explanation of Methods:

- **Feature Detection:** SIFT for keypoint extraction.
- **Homography Computation:** RANSAC-based homography estimation.
- **Blending:** Weighted gradient blending to smooth transitions.
- **Black Border Removal:** Post-processing step to refine the stitched output.

4.3 Implementation Details

The implementation includes:

- Extracting keypoints using SIFT.
- Matching features using FLANN-based matching.
- Computing homography to align images.
- Applying linear gradient technique and blending.

4.4 Final Results



Figure 10: Keypoints of Image 1



Figure 11: Keypoints of Image 2



Figure 12: Keypoints of Image 3



Figure 13: Final Stitched Panorama

5 Link to the Complete Project

The complete project can be accessed at:

https://github.com/Abhik-04/VR_Assignment1_AbhikKumar_IMT2022117