VR Assignment 1: Ayyan Pasha MT2024029

Title: Coin Detection, Segmentation and Panoramic Image Stitching

---

1. Overview

This project addresses two computer vision challenges:

1. **Coin Detection and Segmentation:** Identify and segment individual coins in an image while counting their total number.
2. **Panoramic Stitching:** Combine multiple overlapping images into a unified panorama using feature-based alignment.

---

2. Objectives

1. **Coin Recognition:**
    a. Detect circular regions corresponding to coins in an image.
    b. Segment each coin and compute the total count.
2. **Panorama Stitching:**
    a. Align overlapping images through feature matching.
    b. Seamlessly blend images to create a wide-angle panorama.

---

3. Technical Approach

3.1 Coin Detection & Segmentation

Algorithm Pipeline:

1. **Preprocessing:**
    a. Convert the image to grayscale for simplified processing.
    b. Apply Gaussian blur to reduce noise (kernel size: 11x11).
2. **Circle Detection:**
    a. **Hough Circle Transform:** Detect circular objects using tuned parameters:
        i. `minDist=70` (minimum distance between circles).
        ii. `param2=45` (threshold for false positive suppression).
        iii. Radius constraints (`minRadius=30`, `maxRadius=80`).
3. **Filtering Overlapping Detections:**
    a. Remove concentric circles by checking Euclidean distance between centers.

4. **Visualization & Output:**
    a. Draw bounding circles around detected coins.
    b. Save segmented regions and print the total count.

3.2 Panorama Stitching

Workflow:

1. **Feature Extraction:**
    a. **SIFT (Scale-Invariant Feature Transform):** Detect keypoints and compute descriptors.
2. **Feature Matching:**
    a. **Brute-Force Matcher with Cross-Check:** Ensure mutual consistency in matches.
    b. Select top 50 matches based on distance.
3. **Homography Estimation:**
    a. **RANSAC Algorithm:** Compute the homography matrix robustly.
4. **Image Warping & Blending:**
    a. Calculate panorama dimensions using transformed image corners.
    b. Apply a **translation matrix** to align images spatially.
    c. Overlay the second image onto the warped first image.

---

4. Implementation

4.1 Code Structure

- **coin_recognition.py:** Implements Hough Circle detection and segmentation.
- **panorama_stitching.py:** Handles feature extraction, matching, and stitching.

4.2 Dependencies

- Python 3.x, OpenCV (with contrib modules), NumPy, Matplotlib.
- Installation:

```
- pip3 install opencv-contrib-python numpy matplotlib
```

4.3 Execution Instructions

Coin Detection:

```
python3 1a_Detect_Coins.py
```

- Output: Displays detected coins, saves segmented regions, and prints the count.

Panorama Stitching:

```
python3 2a.py
```

- Output: Generates `stitched_panorama.jpg`.

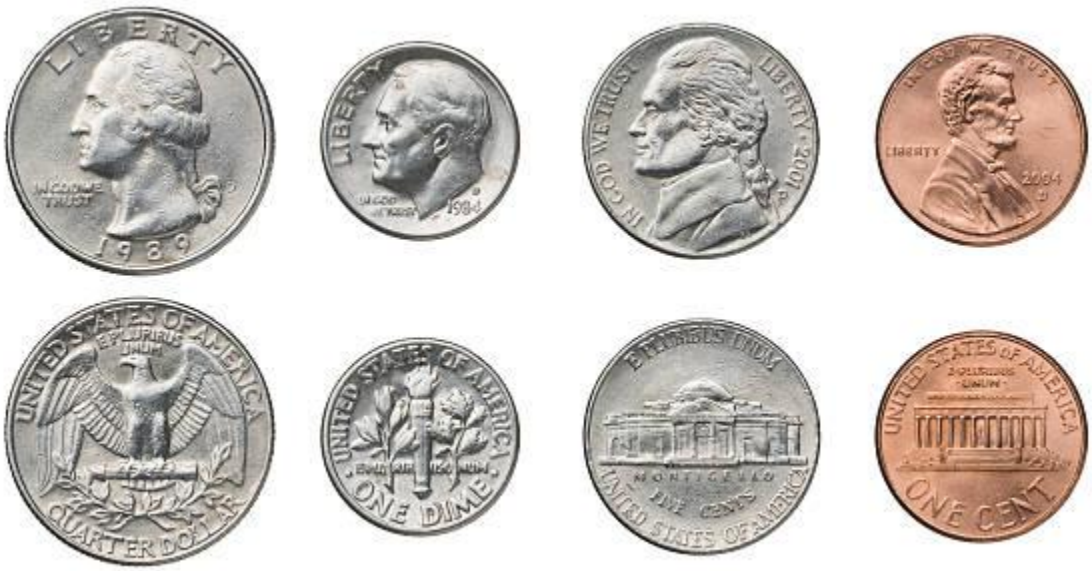
---

5. Results

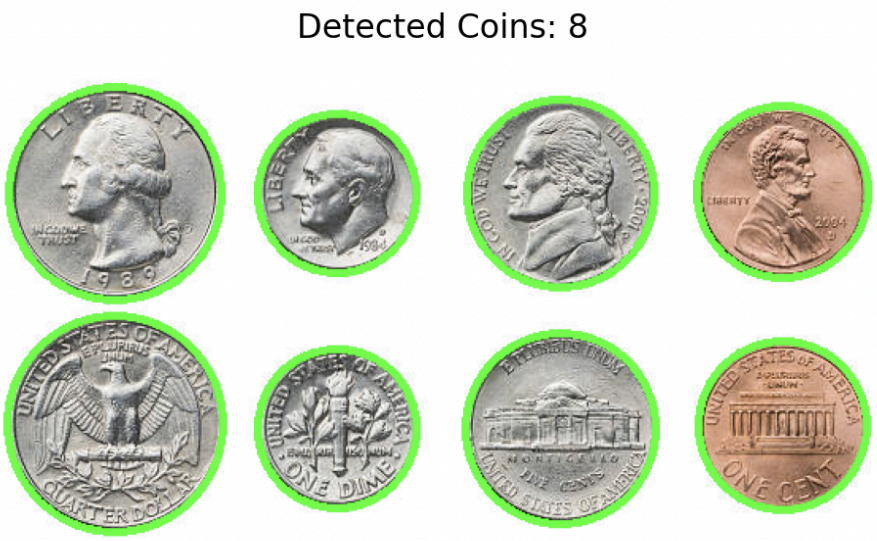
5.1 Coin Recognition

- **Detection Accuracy:** Tuned Hough parameters minimized false positives (e.g., overlapping coins).
- **Segmentation:** Each coin enclosed in a green circle (see `detected_coins.jpg`).
- **Count:** Script outputs: `Total coins detected: 8` (example).

Input:



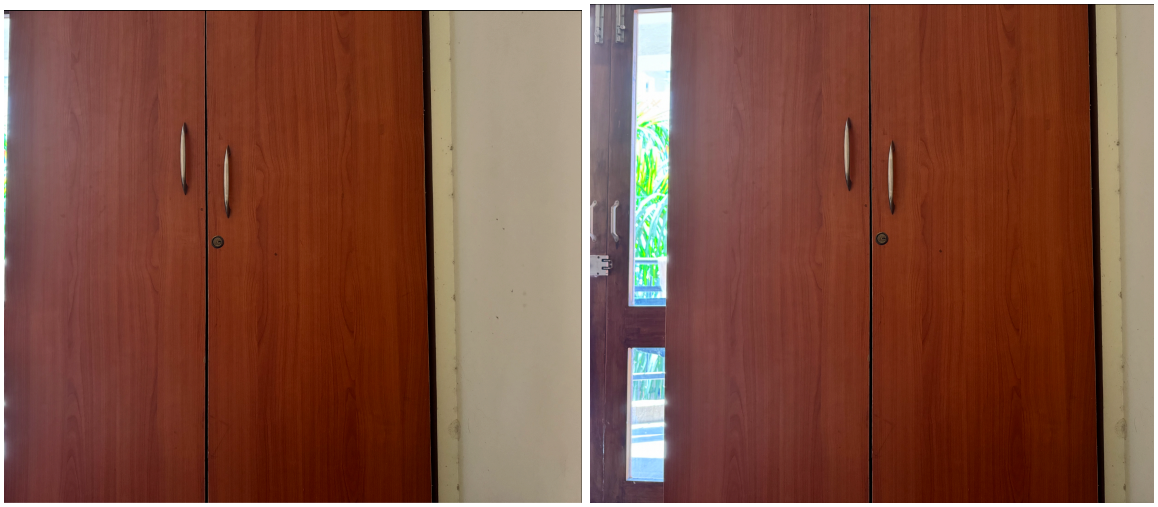
Output:



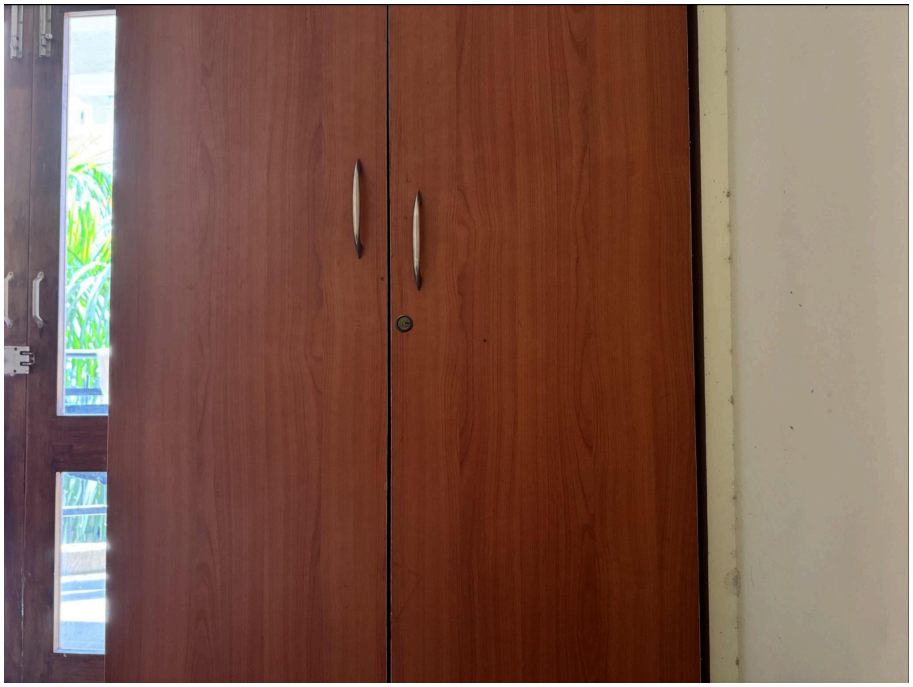
5.2 Panorama Stitching

- **Alignment:** Feature-based homography ensured precise overlap.
- **Output:** Seamless panorama with minimal visible seams (example below).

Input:



Output:



---

6. Challenges & Solutions

1. **Coin Detection:**
    a. **Challenge:** Parameter tuning for varying coin sizes.
    b. **Solution:** Dynamic radius constraints based on image resolution.
2. **Panorama Stitching:**
    a. **Challenge:** Exposure differences causing visible seams.
    b. **Solution:** Translation matrix adjustment for smoother blending.

---

7. Conclusion

- **Coin Recognition:** Demonstrated the effectiveness of Hough Transform for circular object detection.
- **Panorama Stitching:** Highlighted the role of homography and feature matching in image alignment.

GitHub Repository:
https://github.com/ayyanpasha/VR_Assignment

Name: Ayyan Pasha Roll Number: MT2024029