373 Coin Detection Assignment Extension Report - hshi270

Introduction
This report describes the modifications made to the original coin detection pipeline to address the extension requirements for the coin detection assignment. The main goal of this extension was to improve the edge detection process, and successfully detect coins in more challenging images.

Modification 1: Edge Detection using Laplacian Filter

Original: In step 2 of the assignment (Edge detection), we used the 3x3 scharr filter in the horizontal and vertical directions to get the edge maps and took the absolute value.

Extension: The Scharr filter was replaced with the Laplacian filter to improve edge detection, especially in images with less distinct edges. The Laplacian filter is more sensitive to changes in intensity and helps in detecting edges with better accuracy.

- The Laplacian filter was applied by using a custom kernel to detect changes in intensity in the image.
- A 3x3 Laplacian kernel given in the handout was used to compute the second derivative of the image, which highlights regions of rapid intensity change, such as edges.
- The filter was applied to the image by sliding the kernel over each pixel in the grayscale image and calculating the weighted sum of the neighboring pixels, resulting in an image that emphasizes the edges of the coins.

Modification 2: Change of Threshold value

Original: In step 4 of the assignment when we performed a threshholding operation to segment the coin from the background, the threshold value was set to 22 where any pixel value below this threshold was set to 0 and 255 otherwise.

Extension: We increased the threshold value from 22 to 150. This change helps to clearly separate the coins from the background by focusing on the brighter areas of the image, which are usually the coins. The higher threshold reduces unwanted noise and false detections, making the detection more accurate, especially for the hard images with complex backgrounds or uneven lighting. This step ensures that the coins stand out more clearly, leading to a better overall detection.

We did this by setting the threshold value to 150 in the computeThreshold method in which The higher threshold value effectively reduced noise and false detections and also resulted in a more accurate segmentation of coins, especially in images with complex backgrounds or uneven lighting.

Modification 3: Increased Iterations for Dilation and Erosion

Original: In step 5 of the assignment, we performed several dilation steps followed by several erosion steps with 7 iterations each to refine the detected coin regions.

Extension: The number of iterations for both dilation and erosion was increased from 7 to 12. This change allows for more thorough processing of the coin shapes, enhancing the separation of closely placed coins and improving the accuracy of the detected coin boundaries. The additional iterations help in better smoothing and defining the edges of the coins.

We did this by running the dilation and erosion methods 12 times respectively which ensured that the coin shapes were more distinctly refined and any remaining noise or small artifacts were removed more effectively.

Modification 4: Addition to step 7 (drawing bounding box) for counting number of coins and detecting coin type

Original: In step 7 of the assignment, we only extract the bounding boxes around all detected regions (coins) and draw these bounding boxes on the output image. However, the original implementation did not include functionality for counting the number of coins or identifying the type of each coin.

Extension: This part was modified to not only draw the bounding boxes but also to count the total number of detected coins and identify the type of each coin based on its size. This was achieved by calculating the area of each bounding box and comparing it against predefined size ranges for different coin types. Additionally, objects that do not fall within these predefined size ranges or do not have the correct aspect ratio can be ignored, reducing false positives.

Here is the following method:

1) The bounding boxes are extracted around each detected region.
2) The area of each bounding box is calculated.
3) The aspect ratio of the bounding box is checked to ensure it is close to 1, indicating a rough square which outlines a roughly circular shape for coins.
4) If it passes the aspect ratio, the coin type is then determined as follows by its area:
    - $2 coin: Area between 70000 and 76000 pixels
    - $1 coin: Area between 51000 and 57000 pixels
    - 50 cents: Area between 60000 and 68000 pixels
    - 20 cents: Area between 46000 and 49500 pixels
    - 10 cents: Area between 37000 and 45000 pixels
5) Print out the type of coin which is determined by the area range of the bounding box it falls under.
6) Print out the number of coins detected by counting how many bounding boxes fall successfully under the ranges.

Conclusion:
The modifications significantly enhanced the coin detection pipeline, particularly for the hard images. Replacing the Scharr filter with the Laplacian filter improved edge detection accuracy. Increasing the threshold from 22 to 150 effectively separated coins from the background, reducing noise and false detections. Increasing the iterations for dilation and erosion from 7 to 12 improved the accuracy of coin boundary detection. Additionally, counting the detected coins and identifying their types based on bounding box area provided a more comprehensive analysis. These changes made the pipeline more robust and reliable, meeting the extension requirements and improving overall performance.