
Computer Vision

Assignment 1

Sami Sellami

March 3, 2019



Problem3:

The task is to count the total value of coins present in a paper, for that we first need to eliminate the coins touching the boundaries of the paper and then count the value of coins.

For this we followed the steps shown in the following algorithm:

Procedure: Counting the total value of coins in a paper

Input: input image

Output: the total value of coins

1. blur the image
 2. apply canny edge detector
 3. threshold the image and apply morphological operation
 4. **if** the paper is detected as foreground
 5. apply contours and approximation to obtain the corners of the paper
 6. apply perspective transformation to extract the paper from the image
 7. threshold the image and apply morphological operation
 8. **if** the coins are detected as foreground
 9. find contours of the coins
 10. find the bounding circles of each coins and remove the coins touching the boundaries of the paper
 11. use the radius of the circles to extract the value of each coin
 12. draw the values on the final image and count the total value of coins
 13. **else** adjust the thresholding parameters and redo steps 7-10
 14. **else** adjust the thresholding parameters and redo steps 3-10
 15. **return** total value of coins
-

- **step1-3: Preprocessing**

These steps consist of applying blurring to get rid of noise and the high frequencies present in the table, and then, apply canny edge detector to detect the irregularities of the table and remove them by means of thresholding and morphological operations (closing or opening) to extract the paper as foreground, for this an interface was created to adjust the necessary parameters (threshold value, type of morphological operation and the number of times of its application, block size used, use_otsu ...etc) until the best binary image is found

When the code runs it propose to the user to continue if the result is good or to abort the compilation and adjust the parameters once again before relaunching the code, for most of the images, the optimal parameters have already been set, and one should only use the check box *use_defaul_parameters*

In figure one can see clearly the contour of the paper which is gonna be useful to apply perspective transformations

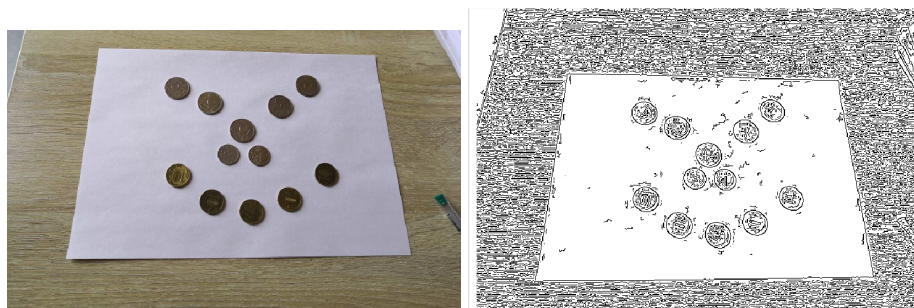


Figure 1: Original image and canny edge detector

- **step 3-6: Paper extraction**

These steps aims to obtain the image of the paper containing the coins, for this we find contours, approximate and filter them to obtain a contour with four points corresponding to the corners of the paper, after that, by using perspective transformation we obtain an image containing the paper and the coins to be

evaluated, it should be noted that the order and position of the corners cannot be known in advance, this is why we implemented a small code for matching the points from source image to the destination; the smallest values of x and y should correspond to the upper left corneretc

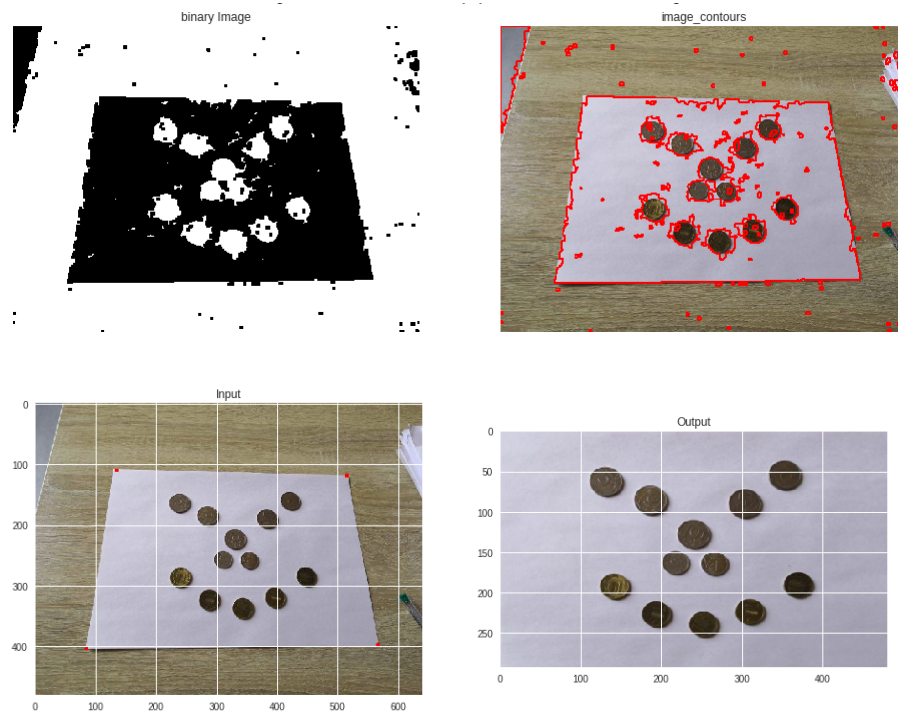


Figure 2: Finding contours and perspective transformation application

- **step 7-12: Coins detection and value estimation**

In these steps, we first apply thresholding using *otsu* method to isolate the coins in the image as foreground, then we apply contours detections to obtain the coins edges, after that we use the function *cv2.minEnclosingCircle* to obtain a bounding circle that will give us an estimation of the radius of the coins.

NB: in order to remove the contours that are inside other contours, we used the *heirarchy* variable returned with *cv2.findContours* to keep only the contours that have no parents

The next step is to extract the value of the coin using the radius informations, for this one should only notice that the coin "1" for example has the small radius and "5" has the biggestetc, we simply used the difference in diameters to get the values required. Finally, we draw the values informations on the final image

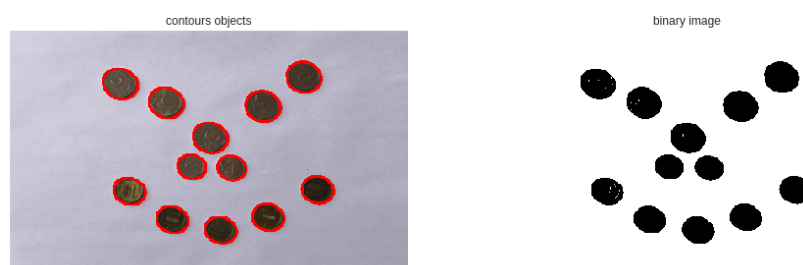


Figure 3: Coins contours detection

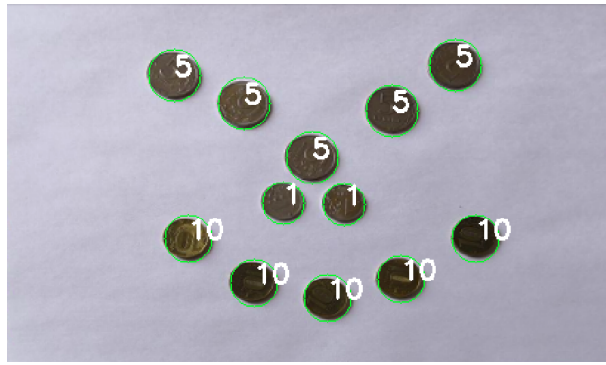


Figure 4: Final image with coins values