Computer vision project: Counting Coins

Raesetje Bonjo Sefala, [redacted]

November 12, 2017

1 Abstract

Object recognition is an important task in Computer Vision. Although, recognising objects in images can be a complex task especially if the objects in the images have a more camouflaged background or there is lighting variation or even different perspectives of the images. This can be a difficult task
This project aims to recognise images of coins on a table by identifying individual segments of the coins including their respective values and then summing them up to get their total value.

2 Introduction

Apart from the fact that humans have been counting coins themselves for a very long time however as the world progresses, the thought of achieving high accuracy results on low quality equipment becomes interesting. These days we can also use low resolution cameras and computing machines to recognise and count objects such as coins.

The data provided consists of 207 unlabeled training RGB images shot from an office table with a bright light source. The images had varying numbers of modern South African coins: 20 cent coins, 50 cent coins, 1 rand coins, 2 rand coins and 2 different 5 rand coins. The images also consisted of arbitrary objects such as pens, rulers and USB hard drives. Figure 1 and figure 2 show examples of the dataset.

Figure 1: Coin image sample 1

3 Methodology

3.1 Detecting the circles

The coin dataset is an unlabeled dataset and as a result we have to start by segmenting the coins from the image setting. Due to the light reflecting onto the table, this task is nontrivial. To try solve the lighting issue, this project will explore background masking techniques.

Figure 2: Coin image sample 2

3.1.1 Background subtraction

Using background subtraction(where image 1 is the image with coins on and image two is the plain image with just the table) resulted in a darker result, hiding the more camouflaged coins. Figure 3 shows an example. Since the table does not look precisely the same in both cases, if you increase the pixel intensity value(with a gamma function for instance),some traces of the difference pops up on the resulting image(figure 4).

Figure 3: Background subtraction result

3.1.2 Thresholding

Since the table is not evenly coloured, Gaussian Blurring and Adaptive Threshold techniques also failed to completely mask the background. Figure 5 demonstrates the result.

3.1.3 Hough Circles and Masking

If we only use a circle detection algorithm such as the Hough circle transform which is is defined by 3 parameters:(x center, y center and the radius) all of which allow us to define a circle but using this

Figure 4: Background subtraction + gamma result

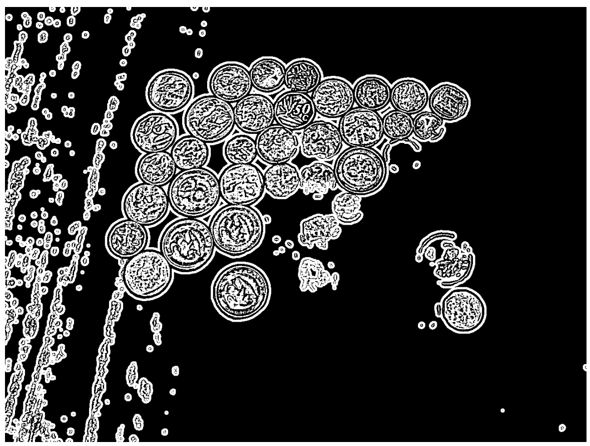

Figure 5: Blur + Thresholding

function alone results in outputs such as the one in figure 6.

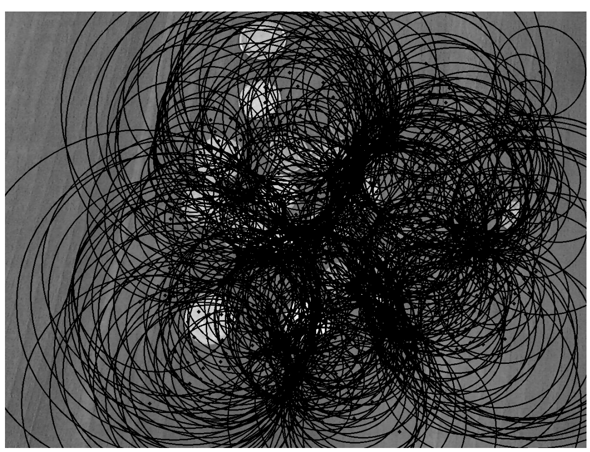

Figure 6: Hough Circle transform on figure 1 data

Instead what gave better results was fine tuning the Hough Circles transform function and then masking the result(that is, anything else outside the circles). Figure 7 shows the resulting masked image.

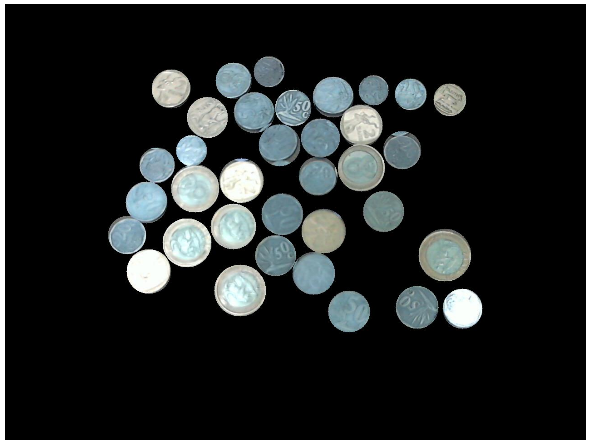

Figure 7: Tuned Hough circle transform + mask

3.1.4 The Watershed algorithm

After trying all these methods, we noticed that the final method worked best to mask the background in order to allow the watershed algorithm to perform much better with better fitting of the circles to show positive coin detection. The Watershed algorithm works by drawing circles from the center going out to try find the edges and thus enclosing the circle for positive detection of a circle.
It has more advantage over the Hough Circles algorithm when it comes to touching or overlapping circles as it is able to better estimate the enclosed circles. We can compare the results with figure 8 and figure 9 below.

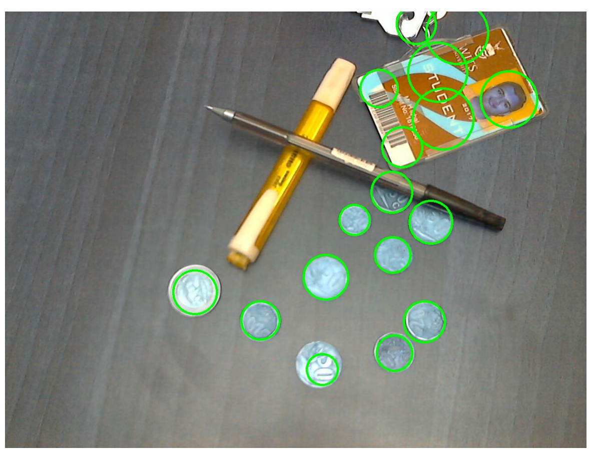

Figure 8: Hough circle transform image output

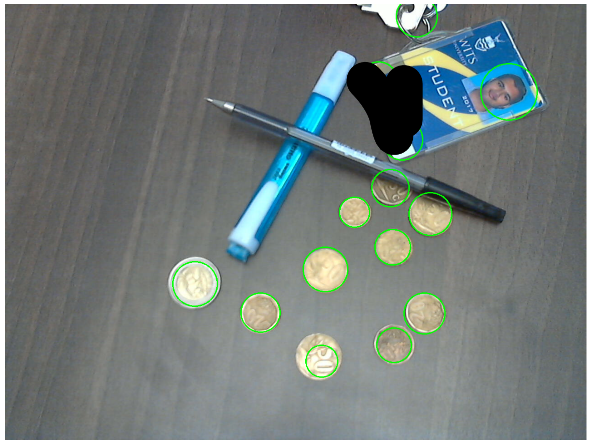

Figure 9: Watershed algorithm image output

3.2 Classification

From the Watershed algorithm, we obtained the center pixel value and the radius of each circle and from these we can use a lookup table to classify the coins according to the color of the center pixel and the radii as demonstrated in figure10.

Figure 10: Final output

4 Observations

The masking technique worked as expected. The main aim was to remove the lighting factor in order to achieve better segmentation. Although, the coin classification step required a better algorithm for more accuracy, a convolutional neural network would perhaps do a better job with classifying the individual coins in the image. This would however require manual labeling of the images.

- Good lighting positions and more homogeneous backgrounds are important properties for more accurate recognition.
- More tightly bound circles from the Hough Circle transform step result in tight circles in the watershed algorithm and the more lose circles allow for more accurate circles in the watershed algorithm if the background is more homogeneous.
- The perspective at which the image shots were taken at makes for inconsistent coin segmentation sizes which makes using the radius as a classifier a not good enough choice. Although the results are good enough(72%)

5 Conclusion

Although we get some accurate results when using the above mentioned methodology, the results would be better if we used a better classification algorithm since the segmentation is quite decent. Although it is generally difficult to segment some coins as they seem to be well camouflaged.
With this project, we can use the above-mentioned algorithms to detect individual coins in an image, classify them according to their value and get an estimate of the total sum of the values on the images.