CSCI 631 - Foundations of Computer Vision

Coin Identification

Kowsic Jayachandiran

Introduction

Vending machines all around the world accept both bills and coins. Vending machines use the physical diameter of the coins to sort the coins to their respective places. The project aims to do the same with images of coins.

Given an image containing coins, the objective of the project is to find the number of coins of each type. The types of coins used for this project are coins used in the United States of America - pennies (1 cents), nickels (5 cents), dimes (10 cents), and quarters (25 cents). The coins are identified using the fact that they differ in their diameters. This employs the use of Hough Transform to find the circles in the image.

Source of Test Images

The images of the coins were taken using a regular cellular phone camera under controlled conditions. The camera was kept stationary while the positions of the coins were changed and new coins were added to create a diverse variety of possibilities of coins. The images were then used uncompressed for testing.

Pre-processing of Images

The images were all captured in black background. This was done to give a good contrast between the coins and the background. The image is converted to black and white using a threshold value so that it will be easier to distinguish between the coins and the background. After doing this, there were noise in the background that were also visible in the binary image. The noise were removed using the fact that they don’t occupy more than 500 pixels in the image. Since the coins are much larger than that, this type of cleaning would not affect the coins. The objects that are only partially visible and touching the border are removed.

Hough Transform

[2] The Hough transform is used to detect lines, circles and other curves. It works even when there is noise in the image. We know that the general linear equation of a line is of the form $y = mx + c$. This line equation can also be represented in polar coordinates: $x\cos\theta + y\sin\theta = r$ where r is the perpendicular distance from the origin to the line and $\theta$is the angle subtended by the perpendicular line with x-axis [2].

The first step is to convert the image to an image with just the edges; canny edge detector works the best for this step. The edge pixels that have the same values of r and $\theta$are believed to belong to the same line. Using this fact, the lines are identified. From this, we understand that Hough transform tries to identify a line or a curve that best fits the edge points. Varying the parameters, we can find different curves in an image using Hough transform [2].

imfindcircles Function in MATLAB

There exists an in-built function in the Image Processing Toolbox of MATLAB called imfindcircles [1] that finds circles in the given image using Hough Transform. Provided with the right parameters, it can even find overlapping circles. The method returns the center coordinates and the radii of each circle found. The Sensitivity parameter helps in identifying the kind of circles that we desire. The values of Sensitivity parameter are in the range of 0 to 1 - the closer the value is to 0, the function will be very sensitive to identifying even the smallest of circles which includes even the noise. Since the pixels occupied by coins are larger than those occupied by noise, the Sensitivity factor needs to be high. This parameter will vary depending on the circles in the image. In case the function doesn't return circles, then we should tweak the value of Sensitivity until it can identify the circles.

Classifying Coins

After the circles have been identified, it becomes a matter of classifying the circles. For testing purposes, it is assumed that there is always at least one dime in the image. This is because the reference size here is a dime. The ratio of the diameters of other coins to this dime is calculated and is then classified accordingly. The reason we choose diameter for classification instead of thickness is that the coin's thickness will be very limited when the picture of the coins is taken from the top.

After a few analyses on the test images, it was found that if the radius of a coin to the radius of the smallest dime is

- greater than 1.4, the coin is a quarter.
- greater than 1.18 and lesser than 1.4, the coin is a nickel.
- greater than 1.1 and lesser than 1.18, the coin is a penny.

It is a dime, otherwise.

A work like this, but using Artificial Neural Network, was done by Velu C.M, Vivekandan P, Kashwan K R in Indian Coin Recognition and Sum Counting System of Image Data Mining Using Artificial Neural Network [3].

Working

The following is the walkthrough of how the script works.

Environmental Constraints

1. The coins are to be kept on a black background.
2. The foreground objects can be only coins.
3. For testing purposes, the camera was held at a fixed distance from the plane of the coins.
4. The lighting must be such that it should be visible on the image taken by the camera, that is the diameter of the coins should be visible.

Image Acquisition

The images were captured through the camera of a cellular phone, in this case Samsung Galaxy S6 Edge Plus. The background was kept black to show contrast between the coins and the background. The images were then used uncompressed for testing. A sample image is shown in Figure 1.

Figure 1. Sample test image captured using camera

Pre-processing of Image

The image acquired through the camera is in RGB color space. This must be changed to grayscale image for differentiating the background from the foreground coins as shown in Figure 2. After conversion to grayscale, histogram of the grayscale image is used to identify the threshold value within which all the information of the coins exist which is shown in Figure 3.

Figure 2. Grayscale image

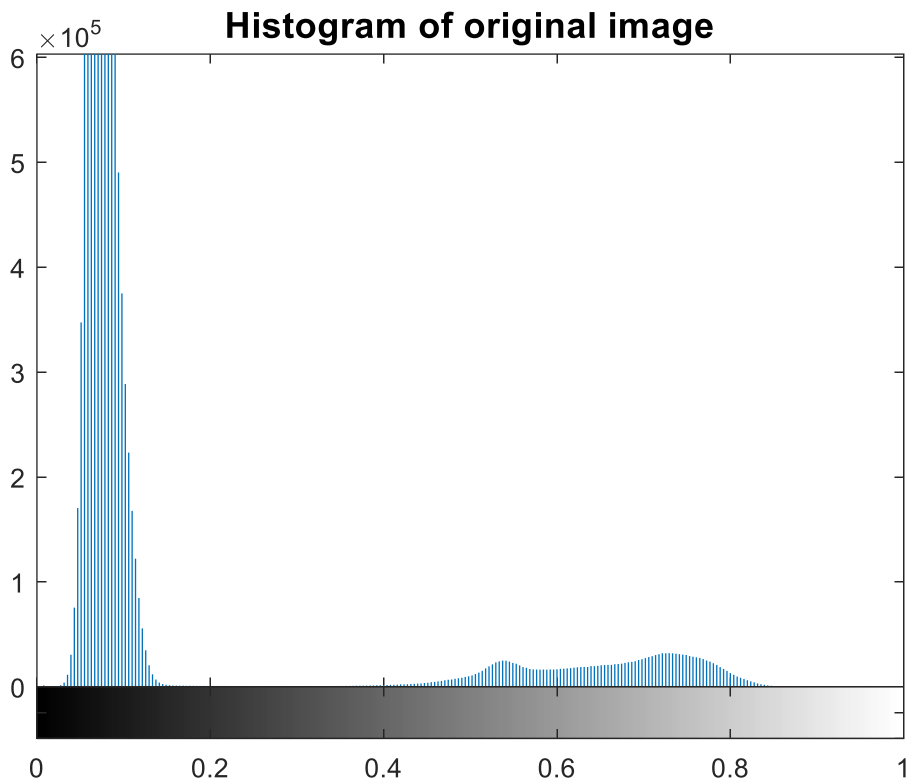

Figure 3. Histogram of original image

From the histogram, we find that the threshold value is 0.2, which means that the foreground coins are within the 0.2 threshold value and the remaining noise constituting the portion beyond 0.2. Using this threshold value, the image is converted to a binary image. The binary image is shown in Figure 4.

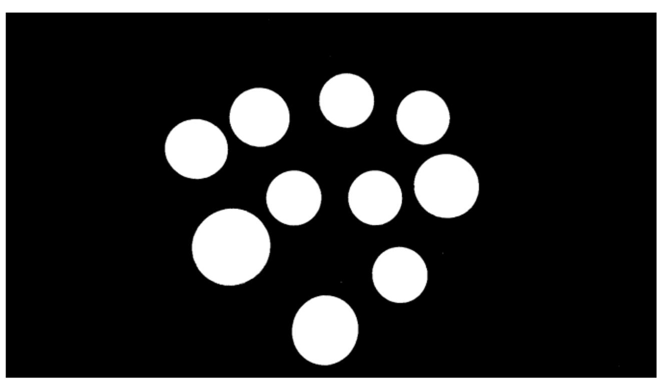

Figure 4. Binary image

Using the fact that the coins occupy greater than 500 pixels in an image, we remove all noise from the image that occupies fewer than 500 pixels. Also, fill the holes, if any, in the coins.

Using Hough Transform to Find Circles

We use the imfindcircles function of MATLAB's Image Processing Toolbox to find the circles. The function employs the use of Hough transform to find the circles in the image. The circles are identified depending on how the parameters of the imfindcircles function are initialized.

We use the binary image to identify circles. In the binary image, we can observe that the circles are brighter in color and the background is dark. We can use this information to set the ObjectPolarity option to *bright* to identify all bright circles in the image [1]. This option alone won't find all the circles in the image; imfindcircles function has another option called Sensitivity which when set to a low value (decimal close to 0) will identify even the smallest noise in the image and when set to a high value (decimal close to 1) will identify large circles [1]. For the test images, it was found that a value of 0.97 will identify all the large coins in the image.

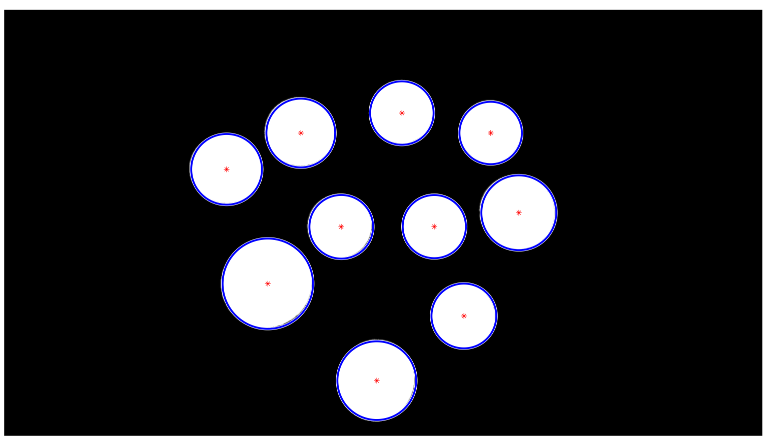

Figure 5. Identified circles using Hough Transform

The imfindcircles function will return the centers and radii of each circle thus identified. These circles have been plotted using viscircles function and shown in Figure 5.

Classifying Coins

After the radii of the circles are obtained, the smallest radius is chosen as the reference, which will be a dime. The ratio of the radius of each circle to this reference dime radius will be used to classify the coins. This classification process has already been explained in an earlier section.

Results

A count is maintained for each denomination. The corresponding count is incremented when there is a match. These counts are then displayed in the console to let the user know the number of coins of each denomination. The total value is also calculated based on the denomination counts to tell the user the total value of all the coins in the image. The results as displayed to the user are shown in Figure 6.

```
Command Window
Quarters : 1
Nickels  : 2
Dimes    : 5
Pennies  : 2

Total value of coins in the image : $0.87
fx >>
```

Figure 6. Results

Challenges Faced

- The Sensitivity option in the imfindcircles function was initially hard to determine as it was in decimals and I had to use brute-force method to check for what value of Sensitivity would all the circles be visible. Since this involved running imfindcircles repeatedly, it took a lot of time to find the Sensitivity value.
- From the distance the camera was held, sometimes the pennies appeared larger and thus were incorrectly classified as nickels.
- The ratio value to classify a coin was hard, particularly in cases where there was no dime in the image. So, I had to add a condition that at least one dime should always be in the picture.

What did I learn?

This project made me understand

- how Hough transforms on circles work,
- how easy it is to identify circles without much coding work,
- how MATLAB made it easy to use just function calls for almost everything instead of having to write the code for each algorithm,
- also, how a simple project such as this could use something as complex as Hough transform in an easy way.

Additionally, this project would work well with Indian coins because Indian coins appear in different sizes and shapes.

Future Works / Improvements

- One straight forward method of identifying coins would be to figure out the engravings on the coin to identify what the coin says – this would mean reading the characters on the coins.
- If the camera was mounted on a tripod stand and fixed, then the diameter of the coins would have been a constant and thus it would have been able to classify even better.
- The project should be able to work with any kind of good contrasting background and lighting even in the presence of some foreign objects in the picture.

References

[1] "Find Circles Using Circular Hough Transform - MATLAB Imfindcircles". *Mathworks.com*. N.p., 2016. Web. 29 Oct. 2016. 07:00 PM.

[2] R. Fisher, S. Perkins, A. Walker and E. Wolfart, "Image Transforms - Hough Transform", Homepages.inf.ed.ac.uk, 2003. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm. [Accessed: 30- Oct- 2016]. 05:00 PM.

[3] C M, Velu, Vivekanandan P, and Kashwan K R. "Indian Coin Recognition and Sum Counting System of Image Data Mining Using Artificial Neural Networks." International Journal of Advanced Science and Technology 31 (2011): n. pag. Web. 28 Nov. 2016. 11:00 PM.

[4] ImageAnalyst's tutorial on Image Segmentation. Image Segmentation Tutorial. http://www.mathworks.com/matlabcentral/fileexchange/25157-image-segmentation-tutorial. Web. 28 Oct. 2016. 11:00 PM.