Lappeenranta University of Technology

Digital Imaging and Image Preprocessing

BM40A1201

Computer recognition of coins

Technical report

2020-01-06

Authors

Noah Hellman	0560307
Akif Nawaz	0555451

Contents

1 Introduction

Image Analysis is the process of using various digital image processing methods to extract useful information from digital images. It is used for simple task such as finding shapes, detecting edges, noise removal as well as complex tasks such as the one presented in this report, counting of Euro coins from an image. These coins are found in smaller denominations like 5, 10, 20 and 50 cents, as well as in larger denominations like 1 and 2 Euros. These coins vary in both radii and their color (hue) which, are quite good properties for distinguishing between the various coins. This report makes use of these two properties of the coins for devising an image based classification system.

For the classification task, we were provided with multiple images of coins placed randomly around a checkerboard. Along with these images we were provided with the corresponding dark, bias and flat field images. Additionally a few images were taken to evaluate the performance of the system under different conditions such as lighting, number of coins, zoom, focal length etc.

The basic algorithm for the developed system involves

- preprocessing the image to achieve intensity as well as spatial calibration,
- segmenting the coins and
- extracting the coins' features and classifying each coin.

2 Method

Given a measurement image of a set of coins on a table with a checkerboard visible, as well as a set of bias, dark and flat field images, the program will determine the amount of coins for each currency that are visible in the measurement. The program first calibrates the intensity of the image using the bias, dark and flat field images as well as real-life scale using the checkerboard with a known width. The coins are then segmented from the images using a thresholded variant of the measurement. Features are then extracted for each coin and a classification is performed using the values of the features.

2.1 Image Preprocessing

The given initial set of images consists of varying number of coins against a checkerboard background. Often these images are bereft with noises due to cross-talk,pixel sensitivities and bias. Intensity calibration is useful technique that can be applied to these images to eliminate these errors:

$$R_{\text{cal}} = \frac{R - \overline{B} - \overline{D}}{\hat{F}}$$

The R_{cal} is the resulting calibrated image. R is the raw image whereas $\overline{B}$ and $\overline{D}$ are the mean of the provided bias and dark images. $\hat{F}$ is the sum of all the flat field images normalized to have a mean of 1 which is calculated using the mean $\overline{F}$ as $\frac{\overline{F}}{|\overline{F}|}$. However, the presence of checkerboard adds some difficulty to the calibration specifically in those cases when the coins are covering the checkerboard area. To counteract the presence of the checkerboards on such coins we eliminated the checkerboard from the flat field image while calibrating the image.

An additional calibration step is also needed to convert the measurements obtained by the camera to real world measurements. This is essential because it may be that not all the different types of coins will be present in all the images. Implying, using relative comparison between the largest and smallest coin in an image would not be an ideal scheme and lead to errors in this case. Hence we proceeded to derive the scaling factor for conversion from camera coordinates to real-world coordinates and hence obtain the diameters of the coins that are quite close in comparison to their real world counterparts. To achieve spatial calibration we make use of the checkerboard as its boxes are known to be of an already measured size. To account for errors in checkerboard points estimation by the in-built MATLAB function we take the mean of the normalized length along each of the inner edge of the board points divided by the number of detected squares along that edge.

2.2 Coin Segmentation

For segregating the coin from the background we adopt a similar approach of checkerboard removal as the checkerboard has significant effect on determining the threshold for binarizing the image. Otsu's method is used to obtain a threshold when the checkerboard is removed. The binarization is then performed with that threshold on the original image. The binary image is then opened and closed morphologically to remove spots within the coins. Finally, the circles of the coins are found using the imfindcircles MATLAB function. The function uses circular Hough transform with Atherton and Kerbyson's phase coding method to detect the circles [1].

2.3 Feature Extraction

In total, three features are used to classify the different coins;

- the diameter of the coin,
- the average hue of the coin, weighted by the saturation, and
- difference of the average saturation of the inner and outer part of the coin.

The diameter is calculated by taking the number of pixels and scaling it according to the above mentioned geometric calibration in order to get a length in millimeters.

The weigthed averaged $\overline{h}$ of the hue is calculated by taking the sum of all the hues multiplied by all the saturations, and dividing it by the total saturation as

$$\overline{h} = \frac{\sum_i^N h_i s_i}{\sum_i^N s_i},$$

where h_i, s_i is the hue and saturation of the i:th pixel within the coin.

The saturation difference is calculated by taking the average saturation of the inner pixels subtracted by the average saturation of the outer pixels. Pixels within 80% of the radius are considered inner and the rest of the pixels within the coin are considered outer pixels.

2.4 Classification

The method used to classify the coins is inspired by fuzzy sets and fuzzy logics. Each currency of coin has three fuzzy sets with each a membership function, one for each feature. For example, the 5 cent coin should have a small diameter, reddish hue and very low saturation difference. The fuzzy set for each currency is the standard intersection (minimum) of its three fuzzy sets.

The coin is then classified as the currency with the highest membership. If the coin doesn't have a membership above a certain threshold for any of the currencies, the object is not considered a coin.

The fuzzy sets or membership functions are created and altered manually to fit the available currencies. The functions are either trapezoid or gaussian sets.

nr	5c	10c	20c	50c	1EUR	2EUR
1772	0	0	0	0	0	0
1773	0	0	0	0	0	0
1774	0	0	0	0	0	0
1775	0	0	0	0	0	0
1776	0	0	0	0	0	0
1777	0	0	0	0	0	0
1778	0	0	0	0	0	0
1779	0	0	-1	0	0	0
1780	0	0	0	0	0	0
1781	0	0	0	0	0	0
1782	0	0	0	0	0	0
1783	0	0	0	0	0	0

Table 1: All errors for each image of the given measurement set.

2.5 Additional images

For the experimentally derived images, a camera was mounted on a pedestal, set at a particular height above the checkerboard with the coins scattered randomly around the camera's field of view.

Firstly a flat-field image was obtained from the set-up and later multiple images were captured with slightly varying camera parameters like zoom and illumination. Illumination was varied with the help of 2 additional light sources while varying the number and denominations of coins. These were used to train the classifier by hand along with the given test measurements.

Another set of images were taken in order to evaluate the final implementation. These was then given labels that were compared against the classifications of the implementation. In order to get an unbiased evaluation, these images were not used at all during the training of the classifier, only for testing the final implementation.

3 Result

An implementation of a detection and classification has been implemented with MatLab using the algorithms described in the method section. The effectiveness of the classifier has been measured using the given measurement set as well as an unbiased evaluation set. The total number of errors are displayed in table 1 for the measurement set and in table 2 for the evaluation set. Positive numbers mean the classifier thought there were more coins of the specified currency than labeled, and vice versa for negative numbers.

For the measurement set, there was one misclassified coin out of a total of 76 coins in 12 images. The implementation did not detect a 20c coin on the checkerboard. For the

nr	5c	10c	20c	50c	1EUR	2EUR
1	-3	-1	0	0	0	0
2	-1	-1	0	0	0	0
3	1	0	0	0	0	-1
4	0	0	0	0	0	0

Table 2: All errors for each image of the evaluation set.

evaluation set, 6 coins were undetected and 1 coin was misclassified for a total of 40 coins in four images.

The accuracy was thus roughly 99% for the measurement set and 82% for the evaluation set. However, after inspection it turns out that all undetected coins were on the checkerboard and the detection ratio was therefore exactly the same as the ratio of coins laying outside the checkerboard. The one coin that was misclassified was a 2 Euro coin mistaken for a 10c coin because only the inner part was segmented.

4 Discussion

Our implementation of an image based coin estimation system is quite adept in determining various coin denominations accurately under ideal conditions. However, special conditions can sometimes lead to errors as the it has detrimental effect on the image segmentation especially in cases where the coin is placed completely on the checkerboard.

The developed methodology was first tested against a provided set of test images and later on an additional set of experimentally obtained images to evaluate the generalizing capability of the system. The system was not only able to detect the coins in the provided image with absolute accuracy but was also able to determine the denominations of the coins with a very good accuracy.

For the evaluation set, the system was again accurate to a very good level. The error mostly arose due to low illumination compounded with the fact that the coins were placed on the checkerboard completely making it even more difficult to segment the coins from the checkerboard background.

As can be seen by the results, the primary issue with the implementation lies in the segmentation. The correctly segmented coins was correctly classified in 100% of the cases. However, coins that lied on the checkerboard were never detected and one coin was incorrectly segmented which caused the total accuracy to go down.

References

[1] MathWorks. imfindcircles - find circles using circular hough transform. https://www.mathworks.com/help/images/ref/imfindcircles.html.