Image Processing Project

Object recognition for coins calculation

DU Pingjie / YANG Di

TPA_09

Tutor : Mohamed Elawady

télécom saint-étienne
école d'ingénieurs / nouvelles technologies

UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

CONTENTS

Introduction

In this image project, we have achieved a coin calculation system which can identify coins inside images captured using a phone camera, differentiate each coin and calculate the total value of money. This system consists of: some new coins, white background paper, phone camera, phone/PC connection cable, PC and the software MATLAB.

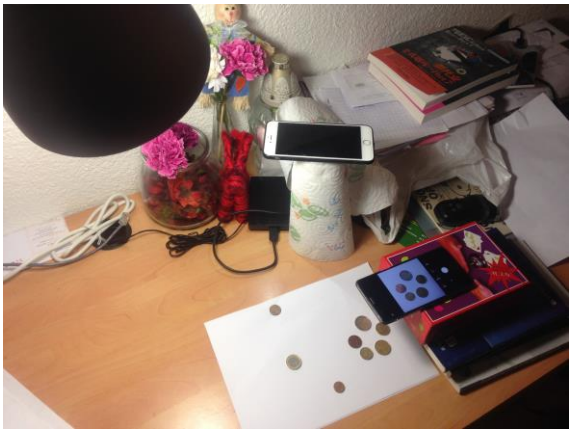


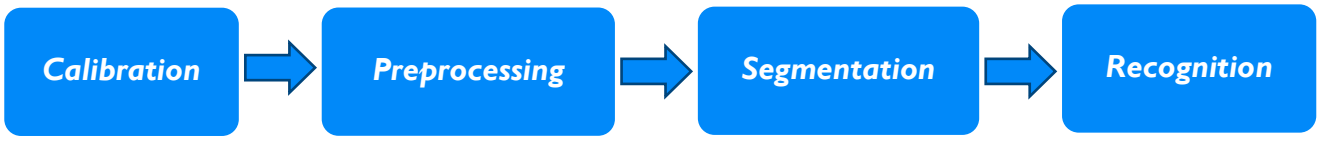
Figure 1:condition of taking photos

We made a dataset including 52 photos (32 Single-object cases, 20 Multi-object cases) to test the functionalities of the application. The photos were taken from a phone called ONEPLUS, which was set at the same height for each image and we could get the same "zoom" on each image. The coins were placed on a piece of white paper.

Figure 2：image of group14

Methodology

To realize this application, we have followed the 4 steps:



➢ Calibration

The calibration allows to define the colors (R, G, B) of different kinds of coins in our dataset as well as the scaling factor (SF) which is the multiplicative factor for making it possible to transform the diameter in pixels to the diameter in millimeters. These settings will be used for recognition. They were calculated by the provided MATLAB code.

Figure 3: measure of diameter

Figure 4: measure of color

➢ Preprocessing

For realizing a better segmentation, it was necessary to make image pretreatments. At first, a Gaussian filter was used for reducing noise (local maxima are smoothed). Then we transformed the RGB domain to the HSV domain with a dynamic expansion. This area is the most powerful to differentiate the background and coins.

Figure 5: image after filter

Figure 6: image HSV

➢ Segmentation

We detected and deleted the shadows at first and we combined 2 methods to detect the coins:
1. threshold segmentation by calculating gradient of the 3rd dimension of the image hsv for detecting 2euro. (sometimes and 1euro)

2. Methodology

2. threshold segmentation directly by 2nd dimension of the image hsv for detecting other coins.

We compared several methods to find the threshold, finally we used the *multithreshold* to segment the image into 3 classes and then we segmented the image again into 2 classes by the second threshold (the bigger one). After that we could detect all coins from the background.

Post-segmentation treatments were carried out to eliminate some defects by the morphological operations. (opening, closing...)

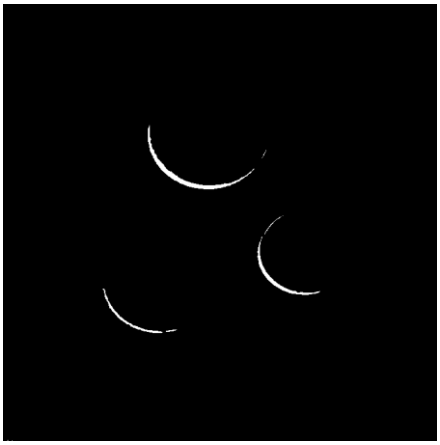


Figure 7: shadows detection

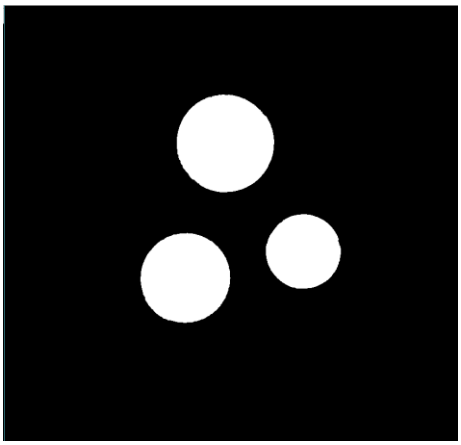


Figure 8: segmentation

- **Recognition**

We assigned a label to each disk and for each label we saved the diameter (equivalent diameter), the coordinates of the central pixel and the coordinates of all the pixels of the label. Moreover, for each label we measured the average color (R, G, B) of the pixels. Then we identified the different coins firstly according to their diameters and then according to the colors for the coins which have almost the same size. (using calibration)

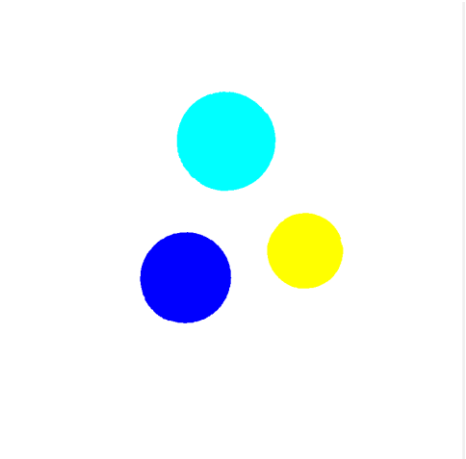


Figure 9: result of label

Figure 10: result of recognition

Results Analysis

- **Analysis I: Quantitative**

Nom en lettres	N° Success	N° Fail	Accuracy
Our dataset	**50**	**2**	**96.15 %**
Provideddataset	**43**	**11**	**79.63 %**

Error analysis:

1. We got some errors due to the copper's and gold's colors. We used the R-G for distinguish the copper and gold, but in the provided dataset, the differences between these two values are very closed, there isn't a clear boundary between them, so it often made mistakes when recognizing coins containing 2/5 cent or 10/20 cent. We should have found another way to recognize the two colors instead of R-G
2. The scale factor given in the dataset is incorrect. We measured that it should be 0.019. (We corrected it when testing)

- **Analysis II: Noise study**

We added Gaussian noise in our dataset. We didn't get a expected result because our photos are so bright that our application couldn't calculate correctly no matter what the value of r was. So, we used the provided dataset for this testing.

3. Results Analysis

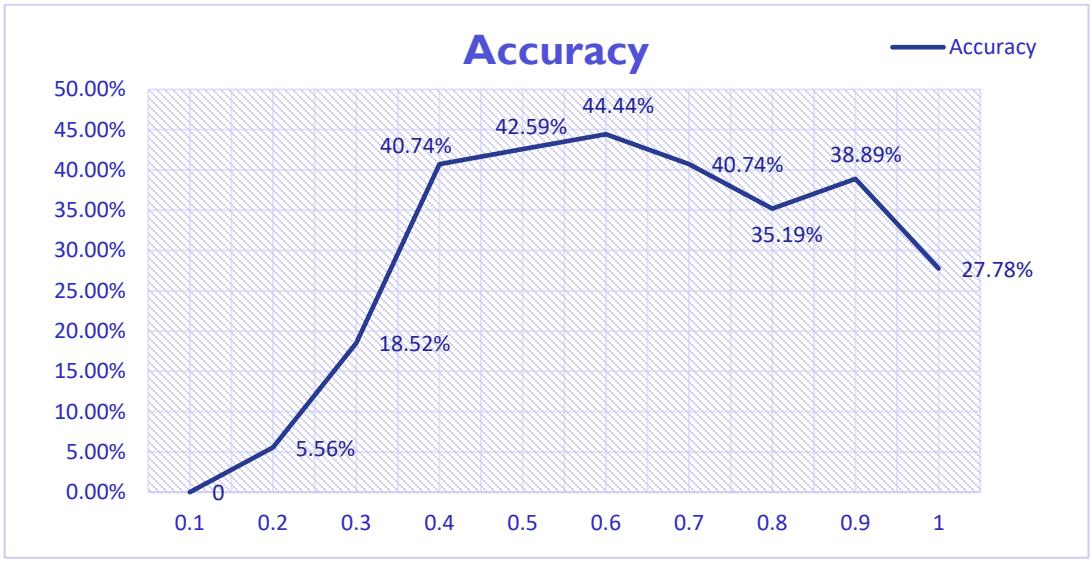


Figure 11: graph of noise study

From the graph, we find that it is impossible to detect the coins when r is very small (0.1) because under this circumstance the image is too bright. The maximum of accuracy happens while r is 0.6 or 0.7. And 'r=1' means that image is just influenced by Gaussian noise, the image will be also relatively difficult to recognize.

Figure 12: image with bruit(r=0.6)

➢ Improvement

1. We should have taken the photos with a little darker light and it would be easier for segmentation and against noise.
2. We should have taken more images because several images in the dataset aren't in good quality
3. It would be better if we also use the method 'contour color' to recognize coins
4. It would be better if we found another way (for example R-G-B) to recognize the two colors of copper and gold instead of R-G.

- **interface (GUI)**

After the analysis we made a interface for our application:

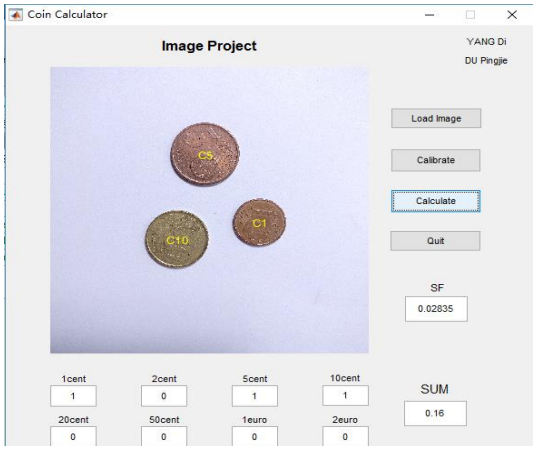


Figure 13: GUI

Steps of using:

Load image → Calibrate → Set → OK → Calculate

Code Explanation

- ***coins_detect.m*:** function for detecting all the coins, calculating the total value of the coins.
- ***tse_imdetectmaxgrad.m, tse_imgrad.m, tse_imhysthreshold.m:*** auxiliary functions used for coins segmentation by the method gradient.
- ***Calibration.m:*** calibration window to input SF
- ***DemoGui.m:*** the main interface of users

Please see comments in the Matlab codes for more details.

Reference

- Aide Matlab
- *https://fr.mathworks.com/matlabcentral/fileexchange/54647-shadow-detection?s_tid=srchtitle*

  Shadow detection in matlab.com
- Documents of presentations of the 4 sessions.