Coin Identification

January 2022

Steps:

1. Calibration
2. Preprocessing
3. Geometric radius estimation
4. Color estimation:
5. Model performance
6. Other approaches (k-means clustering)
7. Documentation

Abstract

Aim - correctly detect and classify coins in image as 5, 10, 20, 50 euro cents and 1,2 euro; count the occurance of each possible type.

Results - authors used geometric and illumination correction to stabilize the images. Authors measured geometry (radius of the coins) with Circle Hough Transform and color (LAB color space intensity probability and histograms similarities using chi square distance) to classify coins.

Conclusions - Overall model achieved 84 % accuracy on the training set. Geometric estimation is prone to error caused by the blurring and morphological operations needed for correct coins detection. Color estimation is biased because of the illumination. Using various approaches resulted in higher accuracy then by using approaches separately.

keywords: Geometric calibration, Illumination calibration, White-color balance, Circle Hough Transform, Multivariate normal probability density function, histogram, chi square distance, LAB, majority voting, Bayesian classification, Gradient, edge detection, morphological operations, image regions.

Contents

1 Calibration

1.1 Illumination Calibration

Having a mean image of dark, bias and flat images first subtract from all images bias image (B_mean), then repeat the process with subtracting dark image (D_mean). In final step we average our flat image around 1 and divide our raw image (R) by it. (List. 1)

Listing 1: Illumination calibration

```
D_mean = D_mean - B_mean;
F_mean = F_mean - B_mean;
R = R - B_mean;
R = R - D_mean;
F_mean = F_mean - D_mean;
F_prim = F_mean - mean2(F_mean) + 1;
c_img = R./F_prim;
```

1.2 Geometric Calibration

For geometric calibration binarized, grayscale flat image containing only checkerboard was used. After morphological closing image regions were obtained by using **regionprops** function. Regions which were detected as squares by comparing region width and height were used to estimate mapping from milimeters to pixel values in regard to horizontal and vertical alignment. Figure 1 depicts green squares which were detected and used in the process.

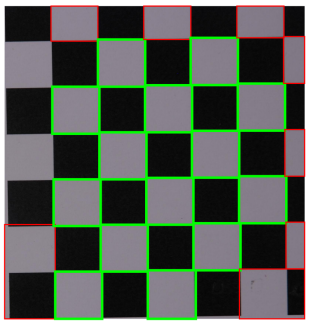


Figure 1: Geometric calibration

2 Preprocessing

2.1 Image cropping

Because of the large size of the images in the dataset it was needed to crop them to the region of interest to increase the speed of the processing. By inverting a binarized

grayscale image of coins it was doable to detect the region which contained all the coins and checkerboard. By using image region information, region with the biggest area's bounding box was used to crop image accordingly. (Fig. 2)

Figure 2: Image cropping

2.2 White-color balance

Because color measurements are very sensitive in regard to the illumination, normalizing illumination in the images approach is used. To do it Matlab built-in **illumpca** function was used to estimate the illuminant and then use **chromadapt** to adjust color balance of RGB image. (List. 2, Fig. 3)

Listing 2: Illumination normalization

```
1       A_lin = rgb2lin(image);
2       illuminant = illumpca(A_lin);
3       B_lin = chromadapt(A_lin, illuminant, ...
            'ColorSpace','linear-rgb');
4       normalized_image = lin2rgb(B_lin);
```

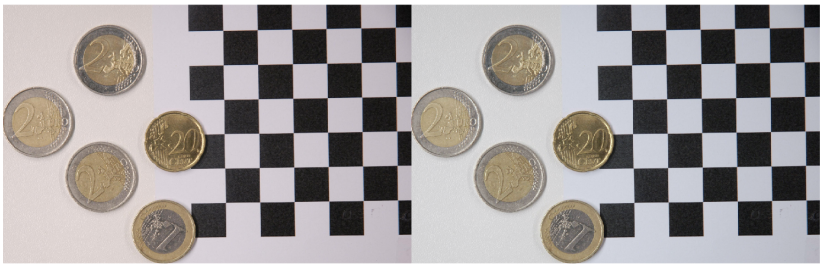


Figure 3: White-color balance

3 Estimation

3.1 Geometric radius estimation

To estimate the radius of the coins using Circle Hough Transform we must obtain a black and white image with distinctive coins regions. Normal approaches using binary thresholding didn't succeeded in this problem, because colors intensity of coins in parts is similiar to the intensity of the background (Fig. 5). Changing the color space to LAB and using B channel (containg information about yellow and blue) improved solution, but still much was to desire. The best solution was obtained by using image gradient and applying morphological operations onto that. For Circle Hough Transform better performance was achieved for blured images, which contained less noise. (List. 3, Fig. 4-6). Because of the huge sensitivity (0.965 in **imfindcircles**) it was necessary to detect if there are some additional circles which need to be omitted in further process. That's why we skipped the circles which repeated itself and were withing the radius of other circle.

Listing 3: Image gradient - ROI

```
[Gmag, ¬] = imgradient(I,'intermediate');
se = strel('disk',30);
BW = imclose(Gmag,se);
h = fspecial('disk',22);
BW = imfilter(BW,h,'replicate');
BW = imgaussfilt(BW,5);
BW = (BW>0.05);
```

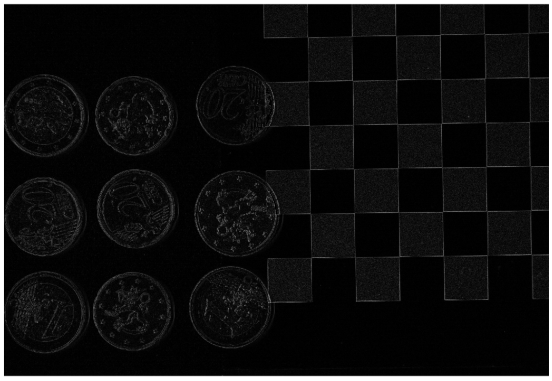
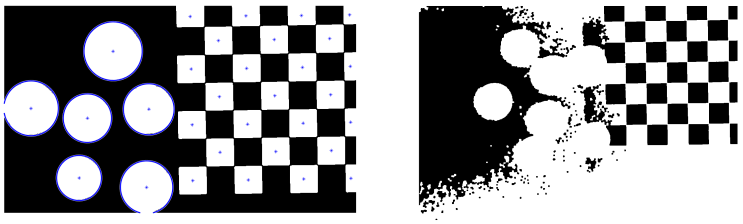


Figure 4: Image gradient



(a) Well illuminated image (b) Badly illuminated image

Figure 5: Coin region of interest using just binary thresholding

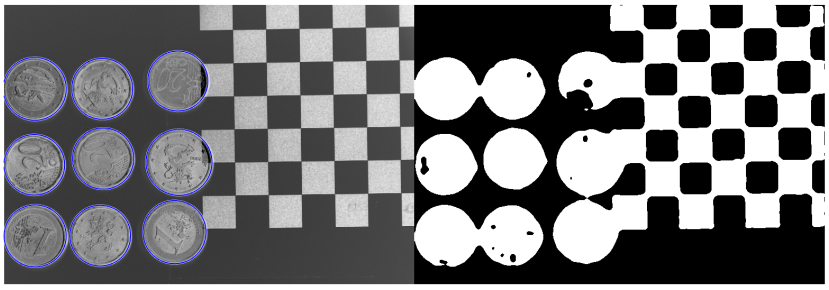


Figure 6: blue circles obtained with Circle Hough Transform and coin region of interest using gradient

Knowing the radii and centers of each coin it is possible to divide one image into cells of images containing one coin only. Then, to improve the estimation, one approach was to remove a shadow which sometimes appeared surrounding a coin as a result of blur and imperfections in image segmentation. I estimated possible RGB bounds of gray shadow and based on that mask was created to eliminate the shadow. Finally obtained radii were compared to the known radii of model coins, their similarity was calculated using normal probability density function.

3.2 Color estimation

3.3 Estimation using LAB color space

Because LAB color space seemed to be working quite well in image segmentation author wanted to see if information about B channel might be useful in color measurement-based estimation. Author used two approaches. One is based on computing histograms of an image and model coins and comparing their similarities using chi square distance method written by Piotr Dollar [1]. Second approach compares scaled and normalized images pixel-wise by using multivariate normal probability density function.

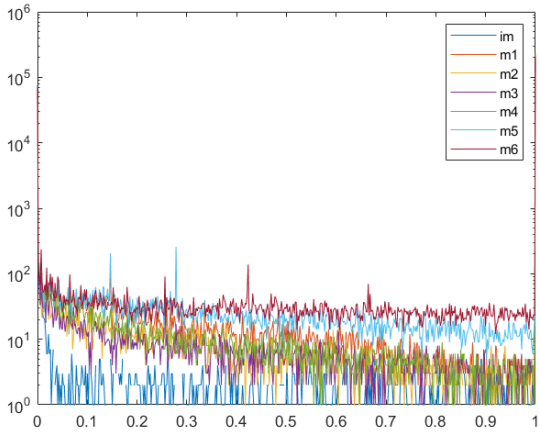


Figure 7: LAB color space histogram comparison

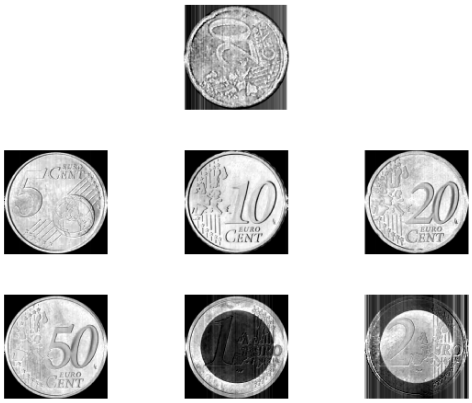


Figure 8: LAB color space comparison

As it can be easily observed on figure 8 1 euro and 2 euro coins are distinctive from others in this color space, however this solution strugles in differentiating 5, 10, 20 and 50 euro cents coins.

3.4 Estimation using histograms

To support previous model histogram is used for comparison in RGB space. As it can be seen on figure 9 there is much more visible difference between histograms in RGB space than in LAB space (Fig. 7)

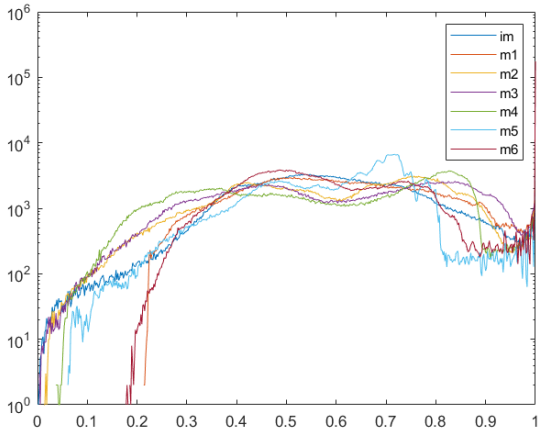


Figure 9: RGB color space histogram comparison

Because smaller histograms distances means bigger similarity after min-max scaling it we need to subtract our result from 1 so now the coin model with smallest distance gives us the biggest "probability". After obtaining both geometric and color estimation combine the votes using majority voting system with weights to certain approaches which were estimated by trail and error.

4 Model performance

I managed to achieve a decent total average accuracy of 84 %. Table 1 presents accuracy of each training image.

Table 1: Model performance

Photo name	Coins	5	10	20	50	1	2	Missing	Error	Accuracy(%)
DSC1772	estimation	1	1	1	1	1	1	0	0	100
	correct	1	1	1	1	1	1			
DSC1773	estimation	0	0	1	0	1	3	0	0	100
	correct	0	0	1	0	1	3			
DSC1774	estimation	3	1	3	0	0	1	0	4	50
	correct	1	1	5	0	0	1			
DSC1775	estimation	3	1	2	0	0	0	1	0	85.7
	correct	3	1	3	0	0	0			
DSC1776	estimation	5	1	2	0	0	0	1	4	44.4
	correct	3	1	4	0	1	0			
DSC1777	estimation	2	0	1	0	2	0	1	0	83.3
	correct	2	0	1	0	3	0			
DSC1778	estimation	0	0	3	0	1	0	0	0	100
	correct	0	0	3	0	1	0			
DSC1779	estimation	3	0	5	0	0	0	0	1	87.5
	correct	3	0	4	1	0	0			
DSC1780	estimation	3	1	0	0	0	0	0	0	100
	correct	3	1	0	0	0	0			
DSC1781	estimation	0	0	4	1	0	0	0	0	100
	correct	0	0	4	1	0	0			
DSC1782	estimation	2	0	4	0	3	0	0	2	77.7
	correct	0	0	5	1	3	0			
DSC1783	estimation	0	0	2	0	3	0	0	1	80
	correct	0	0	1	1	3	0			
TOTAL										**84**

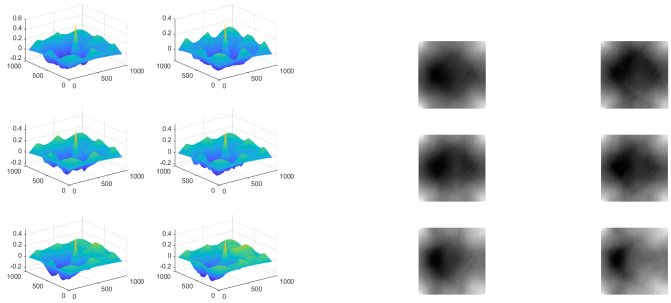
[H]

5 Other approaches tried

Although approaches didn't reach the final model they have potential to be used in the future to improve the model. First approach for color estimation was supposed to compute the distance of RGB vectors of images which colors were clustered into k values. This solution was very promising, but was very time consuming. Because it underperform in comparison to the histogram approac that is why it didn't appear in final solution.(Fig. 10)

Figure 10: RGB k-means clustering

Other tried solution was trying to find some pattern onto which the model could recognize the solution. The biggest problem in this approach would be to recognize tails because in euro coins they differ. Author tried to use normx2corr and Fourier transform correlation to estimate patterns. Author sees a potential in using CNN as well in this problem, but because of the time limit he didn't manage to try this approach.



(a) Normxcorr2 (b) Fourier transform

Figure 11: Coin pattern recognitions

Listings

References

[1] Chi square distance function. `http://www.cs.columbia.edu/~mmerler/project/code/pdist2.m`. Last accessed: 18.01.2022.