BM40A1201 DIGITAL IMAGING AND IMAGE PREPROCESSING

ABSTRACT

Digital imaging techniques based system for precise coin identification and counting, vital for real-world applications like automated cashier systems and self-checkout.

STUDENTS

Durbar Hore Partha 001473021
Shruti Varpe|002049771

Contents:

1. Abstract:

Using the proposed system, we aim to delve into the realm of coins identification and calculation by deploying the digital imaging techniques. The ability to identify and count coins in digital images is essential in real-world applications, such as automated cashier systems. The intricacy stems from the various features of coins, such as differences in color and radius, which call for advanced image processing.

The suggested methodology consists of coin segmentation and feature extraction for precise identification and counting, after preprocessing stages to calibrate intensity and spatial aspects. Automated vending machines and self-checkout systems at supermarkets are practical instances of this technology in operation, as they require quick and accurate coin identification to ensure smooth transactions.

The system's capacity to adjust to various environmental factors, including as changes in illumination and focus, is essential to its effectiveness in a variety of environments. The usefulness of using digital image processing to expedite and improve the precision of coin recognition and counting procedures in practical applications is highlighted in this abstract.

2. Introduction:

The proposed system puts to use the concepts of Image Processing and Analysis in order to extract vital information from the digital images like shapes identification, edge detection, noise removal, etc. for the complex task such as counting Euro coins of values 5, 10, 20 and 50 cents, as well as 1 and 2 Euros from the provided images. For this, the digital images are firstly preprocessed, several image processing techniques and segmentation are applied, features of the images are extracted, classification is performed and Image Analysis is employed.

The emphasis is given on the different diameters and colors of coins as distinctive features, with a focus on extracting useful information from photographs. These characteristics are the foundation of the presented classification system. In order to perform the work, it is necessary to analyze several photographs of coins organized on a checkerboard along with images of flat field, dark field, and bias. Preprocessing stages for coin segmentation, intensity and spatial calibration, and feature extraction for classification are all included in the created approach. The system's performance is examined under varied settings such lighting, coin number, magnification, and focal length adjustments.

The generic algorithm for the system involves:

- Preprocessing techniques for optimal intensity and spatial calibration
- Segmentation of coins
- Feature extraction and classifying each coin.

3. Method:

The system will calculate how many coins of each currency are visible in the measurement given a measurement image of a set of coins on a table with a checkerboard visible and a set of bias, dark, and flat field photos. Using bias, dark, and flat field photos as well as real-life scale using a checkerboard with a defined width, the application first calibrates the image's intensity. Next, a thresholded version of the measurement is used to separate the coins from the pictures. Following the extraction of features for each coin, a classification is carried out using the feature values.

a. Image Preprocessing:

i. Preprocessing digital images:

Preprocessing digital images concentrated in the function of the script “DIIP_Project” The main objectives are to normalize flat fields, eliminate bias and dark fluctuations, and then calibrate the input image. For improved image clarity, the script also includes a checkerboard removal method.

ii. Image Loading and Processing:

For measurement, bias, dark, and flat fields, the script reads images from designated directories. The flat field is normalized and mean bias and dark

images are computed to remove variances. The primary step in the calibration procedure is to take the raw measurement image, subtract the mean bias and dark, then divide the result by the normalized flat field. A calibrated image prepared for additional analysis is the end result.

iii. Checkerboard detection:

To locate a checkerboard pattern in the measuring image, the script makes use of the **detectCheckerboardPoints** function. The script calculates the coordinates of a checkerboard's four corners and makes a mask to eliminate the checkerboard from the original image if one is found. This procedure removes undesirable patterns, improving the quality of the image.

Let us explain the steps:

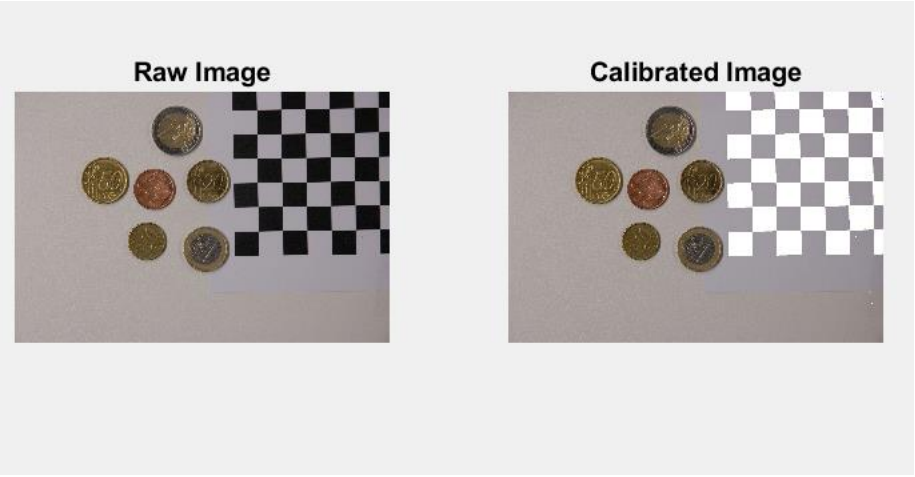
- Calibration

For calibrating the image Bias images, dark images, flat images were given. For calibrating the images mean of the Bias, dark and Normalized Norm_F values of flat field images were calculated and then the below formula is used:

$$R_{cal} = (R - B\hat{} - D\hat{})/F\hat{}$$

Using this formula the given raw image is calibrated.

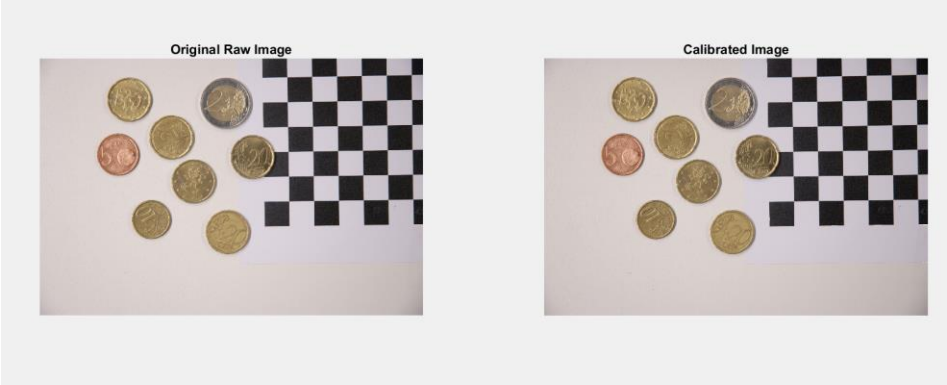
- Checkerboard poin detection:
  Then the checkerboard points are detected using matlab built in function
- Mask the checker board using polygon: We do 2D masking of polygon then present it in 3D



- Remove checker board: Removing checkerboard was tried but the image scale is difficult to be processed for further analysis.

p.s.: Used for trial

- As the previous attempt was not successful the image is further calibrated using calibration_measurement function
- Checkerboard_updated fuction was built to finally process and get calibrated image for calibration.
- replace_Dhp function is used to mask the checkerboard into binary and get the calibrated image.



Rescaling Image:

Addition to this a method is used for calibration to rescale the calibrated image to real life image scaling using scale_DHP function. It is necessary to scale the image to real life image for scaling the imaging so that it can be classifies based on the real life attributes of the coin.

Coin Classification:

Coin segmentation is a multi-step technique that is implemented into the coin estimate function **estim_coins().** To guarantee a consistent size, it starts by resizing all of the input images, including measurement, bias, dark, and flat images. Next, using the mean bias, mean dark, and normalizing factor, the **calibration_measurement()** function is used to calibrate the measurement image's intensity. In the calibrated image, checkerboard locations are identified, providing

reference points for geometric calibration. The goal of the intensity calibration is to account for differences in the system by combining bias, dark, and flat field images.

The **circle_detection()** function is then used to identify circular items (coins) in the calibrated image, most likely by employing characteristics unique to circular forms. This step is essential to separate prospective coins from the remainder of the image.

b. Coin Estimation Function:

The coin estimating procedure is managed by the **estim_coins()** function, which integrates estimation of various coins effectively. To enable geometric calibration, it begins by scaling input photos and finding checkerboard points. Next comes intensity calibration, which makes use of each image's normalization factor, mean bias, and mean dark.

Next, we use the **circle_detection()** function to find circular objects in the calibrated image that could be coins. The computation of a scaling factor facilitates the determination of the physical dimensions of the observed coins by connecting pixel measurements to real-world dimensions.

Using the **Image_features()** function, the program iterates through the detected circular objects, extracting features for each one. These attributes include things like pixel intensity distribution, size, and form. These characteristics are used to classify coins and differentiate between various coin types. This procedure makes it possible to estimate how many and what kinds of coins are included in the input images.

c. Feature Extraction:

A crucial step in the coin estimate process is feature extraction. In order to translate pixel measurements into actual sizes, the algorithm computes a scaling factor, which establishes a vital connection between image data and physical dimensions. Next, for every circular object found in the calibrated image, features are retrieved. These characteristics, which were calculated with the help of the **Image_features** function, contain details on each coin's dimensions, form, and distribution of pixel intensity.

Coins can be effectively categorized into many classes by removing these distinguishing features. Feature extraction is necessary in order to obtain qualitative and quantitative data that facilitates precise coin recognition and quantification inside the images.

d. Evaluation:

A model's performance is assessed using the **evaluate.m** script, which compares a set of initial measures to the output. The findings are estimated using concepts of

the Euclidean distance metric, mean error and a predetermined threshold. Our proposed model produces a performance metric of 91%, which shows how well the model matches the initial measurements with its predictions.

4. Evaluation:

While evaluating our experiment we had different result sets other than the reference values given in the moodle.
For the very first iteration our reference values were:

First ateempt:

DSC1772: 1,2,0,1,1,1
DSC1773: 3,1,0,2,0,0
DSC1774: 1,0,0,6,1,0
DSC1775: 0,0,0,3,1,3
DSC1776: 0,1,0,4,2,0
DSC1777: 0,3,0,1,0,2
DSC1778: 0,1,0,3,0,0
DSC1779: 0,0,2,4,0,2
DSC1780: 0,0,0,0,1,3
DSC1781: 0,0,1,4,0,0
DSC1782: 0,3,1,5,0,0
DSC1783: 0,3,1,1,0,0

mean error:0.40+
Accuracy: 60% (approximate)

With improved scaling:

DSC1772: 1,2,0,2,1,0
DSC1773: 3,1,0,2,0,0
DSC1774: 1,0,0,5,1,1
DSC1775: 0,0,0,3,1,3
DSC1776: 0,1,0,4,1,3
DSC1777: 0,3,0,1,0,2
DSC1778: 0,1,0,3,0,0
DSC1779: 0,0,2,4,0,2
DSC1780: 0,0,0,0,1,3
DSC1781: 0,0,1,4,0,0
DSC1782: 0,3,1,5,0,0
DSC1783: 0,3,1,1,0,0

Which gone mean error of 0.23
And the accuracy we got was 83%

After iterating for several times and changing various classification criterions in functions we achieved

With improved image_features

DSC1772: 1,1,1,2,1,0
DSC1773: 3,1,0,1,0,0
DSC1774: 1,0,0,5,1,1
DSC1775: 0,0,0,3,1,3
DSC1776: 0,1,0,4,1,3
DSC1777: 0,3,0,1,0,2
DSC1778: 0,1,0,3,0,0
DSC1779: 0,0,1,4,0,3
DSC1780: 0,0,0,0,1,3
DSC1781: 0,0,1,4,0,0
DSC1782: 0,3,1,5,0,0
DSC1783: 0,3,1,1,0,0

Mean Error: 0.11785
Accuracy: 91.6667%

Where we got accuracy over 90%

Discussion on evaluation:
Higher accuracy can be achieved by improving the experiment:

- Camera calibration improvement: Image calibration can be improved with improvised camera calibration techniques.
- Image scaling after calibration: If the image is rescaled alike a real-life image after calibration it can be more accurate.
- Image features improvement: Image features can be improved using certain features more accurately.
- Coin classification: Based on image features the coin dimensions, width, saturation, hue calculation can be improved using better calculation technique

5. Future Scopes:

The project of coin detection and estimation of Digital Imaging and Image processing inherits scopes for the future of this field. This project is implemented based on modern evaluation and camera calibration techniques. This project can be improved in the future for better detection and more accurate image analysis, detection or estimation. This project relies on mathematical approaches and inbuilt matlab functions allowed in LUT university, which may help in the future for building better camera calibration and image detection or estimation model building.

In the future, the technology might be used to estimate currencies from around the world, meeting the various needs of various locations. Estimating historical coins could provide insightful information about numismatics and cultural heritage. Simplifying financial transactions can be achieved by investigating real-time cash counting applications at self-checkout counters and cashier systems.

Applications for the technology might also be found in places like museum exhibits, where it's crucial to identify and classify antique coins. Furthermore, real-time cash counting can be deployed at cashier systems and self-checkout counters in supermarkets.

6. Conclusion:

The suggested coin identification and counting system that makes use of digital imaging techniques effectively resolves the issues of color and radius variations. The system provides accurate coin identification and counting with strong picture preprocessing, calibration, coin segmentation, and feature extraction. It is appropriate for applications such as self-checkout in supermarkets and automated cashier systems due to its flexibility in a variety of situations. Subsequent improvements might concentrate on streamlining the calibration procedures, enhancing feature extraction and segmentation, and expanding compatibility to accommodate a wider range of currencies for worldwide use.

7. References:

- https://se.mathworks.com/help/vision/camera-calibration.html
- https://github.com/rohitrango/automatic-watermark-detection
- https://se.mathworks.com/matlabcentral/fileexchange/14042-coin-recognition
- https://github.com/himanshusharma89/Coin-Detection-using-MATLAB
- https://se.mathworks.com/matlabcentral/answers/786741-help-with-coin-counting-from-an-image
- https://stackoverflow.com/questions/26855264/identifying-different-coin-values-from-an-image-using-matlab