IVR Coursework 1
Report

Weiting Goh (S1450710) and Tomas Markevicius(S1452595)

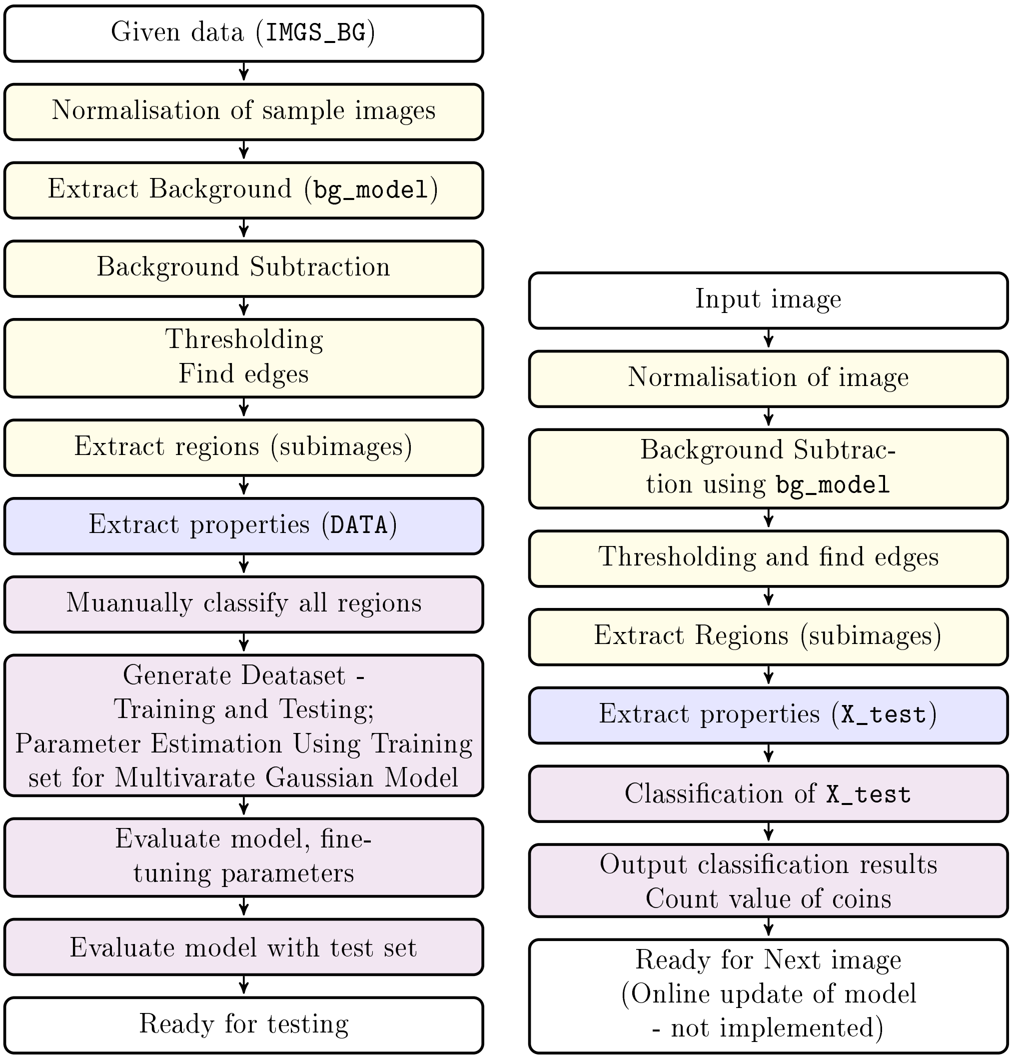
October 26, 2016

1 Introduction

To recognise and count the coins in an image, Coinsy, have three subtasks: 1) Image processing (includes image segmentation), 2) Features extraction, and 3) Classifiation (and then counting the coins).

For Coinsy to be a proficient detector, classifier and counter, we have to first train it. The training processes differs from evaluation as described in figure Figure 1 - the operation pipeline for training and evaluation. We will describe the methods that we use for each of these subtasks in the next section - methodology, and her evaluation results after. Lastly, we conclude with a discussion on Coinsy performance. The code for the project is listed in the appendix.

In the following subsections, we give an brief description of the operation pipeline for each subtasks from training to testing. The approach here is an abstract idea of what we did for our base model. Additional techniques were explored and will be discussed in later section. We start by understanding the data.



(a) Operation pipeline to train Coinsy

(b) Pipeline for (trained) Coinsy to count coins

Figure 1: Differences in Pipeline; Yellow boxes indicates the image processing procedures; Blue indicates the feature extraction procedures; Violet indicates the classification procedures.

Data

The data we were given are 14 images (consisting of 5 `harder` and 6 `simpler images`). Each consists of objects in the foreground for us to classify and calculate the values. The class labels and its repecitve oobject and values are as follow:

Class	Object	Value
1	1 Pound coin	1 pound
2	2 Pound coin	2 pound
3	50 Pence coin	50 pence
4	20 Pence coin	20 pence
5	5 Pence coin	5 pence
6	Washer with small hole	75 pence
7	Washer with large hole	25 pence
8	Angle bracket	2 pence
9	AAA Battery	no value
10	Nut	no value
11	unknown	-

The objects differ in colors, size and shapes; some are very similar to the background - such as 1 pound coins (see Figure 2). Hence, the classifier must be invariant to rotation of objects, and importantly to detect the objects will the images to be processed, such that the objects are salient to the computer vision. The real challenge lies in segmentation of the image.

Figure 2: Sample images from given data.

1.1 Image Processing

The image processing step aims to 1) make all images (for training the model or for evaluation) comparable, 2) extract objects in the foreground (also

Figure 3: Sample images after their background is subtracted.

known as image segmentation). The outcome of this stage is an array of subimages ready for feature extraction.

The central idea is to, first, model the background before subtracting it from all images; second - make the edges of the objects 'obvious' by thresholding; third - crop out the objects to obtain the subimages. This is one of the many image segmentation technique available, and further exploration of other techniques is discussed later.

Background Subtraction

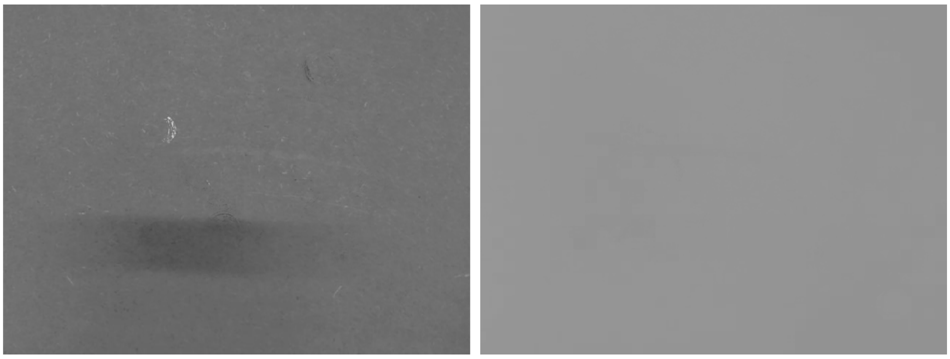
In this coursework, since a background image is not readily available, we have to model it. Noting that images varies in illumination, we have to make the images comparable by normalising it first, using the following formula.

$$P_{r,c}(R', G', B') = (\frac{R}{\sqrt(R^2 + G^2 + B^2)}, \frac{G}{\sqrt(R^2 + G^2 + B^2)}, \frac{B}{\sqrt(R^2 + G^2 + B^2)})$$

With objects scattered around randomly in the images, we find the median of all image pixels for each channel separately in order to reconstruct the background.

The outcome of the background with and without normalisation is shown in Figure 4.

The sample images with their background removed is shown in Figure 3. It is evident that the background removal process removed the background



(a) Background model without normalisation

(b) Background model after normalisation

Figure 4: Background model generated from all 14 images

- making the images appearing black. However, it also inevitably reduce the intensity for the bottom half of each images, such that the objects are no longer salient to our eyes. This is because the background we modelled have a lower intensity at the bottom, possibly due to presence of shadow in all 14 images.

Nevertheless, the historgram is still bimodal, which is essential for thresholding to be effective.

Segmentation

1.2 Classification

2 Methodology

THIS IS THE TARGET

3 Result

A For training

```
%% SCRIPT FOR IMAGE SEGMENTATION
%% PRE-PROCESSING%%
fprintf('\t\tPREPROCESSING IMAGES\n');
%% NORMALISATION
%   Here, we normalise the images before extracting the background
disp(bar);

fprintf('\n>> Normalising the images\n');
for i = 1:num_img_bg
    fprintf('.');
    IMGS_BG{i} = normalise_RGB(IMGS_BG{i},0);
%     figure; % DEBUG
end
fprintf('\n%d images Normalised\n\n', i);

disp(bar);

%% IMAGE SEGMENTATION %%
fprintf('\t\tIMAGE SEGMENTATION\n');
fprintf('Approach:\n1) Generate background model\n2) Background subtraction\n3) Thresholding\n');
%% GENERATE BACKGROUND MODEL
disp(bar);

WINDOW_SIZE = 1;
fprintf('\n>> Generating Background Model With WINDOW_SIZE = %d\n', WINDOW_SIZE);
bg_model = bg_extract(IMGS_BG, WINDOW_SIZE);

fprintf('\n'); disp(bar);

```

```
%% BACKGROUND SUBTRACTION
disp(bar);

fprintf('\n>> Subtracting bg_model from all IMGS\n');
IMGS_NORM   = IMGS_BG(1:9); % get all the normalised images,
[IMGS_BGREMOVE, ~] = bg_subtraction(IMGS_NORM, bg_model);

fprintf('\n'); disp(bar);

%% THRESHOLDING
% do the normal thresholding here,
%   i.e. thresh(I_r - B_r) | ...| thresh(I_b - B_r)
disp(bar);

fprintf('\n>> Doing thresholding...');
sizeparam = 16;
[IMGS_THRESH, thresh_vals] = dothresh(IMGS_BGREMOVE, sizeparam)
    ;

fprintf('\nIMGS_THRESH and thresh_vals added\n\n'); disp(bar);

%% NOTE:
% AT this stage, we have threshold all our training set.
% call another script for feature extraction

%% NEXT>> EXTRACT FEATURES
```

References

[1] E.R. Davies. *Computer and Machine Vision (Fourth Edition)*. Academic Press, Boston, fourth edition edition, 2012. ISBN 978-0-12-386908-1. doi: http://dx.doi.org/10.1016/B978-0-12-386908-1.

00001-X. URL http://www.sciencedirect.com/science/article/pii/B978012386908100001X.

[2] Noah Snavely. Lecture 2: Image filtering. URL http://www.cs.cornell.edu/courses/cs6670/2011sp/lectures/lec02_filter.pdf.

[3] A. Walker R. Fisher, S. Perkins and E. Wolfart. Hipr2 - image processing learning resources. URL http://homepages.inf.ed.ac.uk/rbf/HIPR2/index.htm.

[4] Andrew Ng. Stanford machine learning class notes. URL http://www.holehouse.org/mlclass/index.html.