

1 Introduction

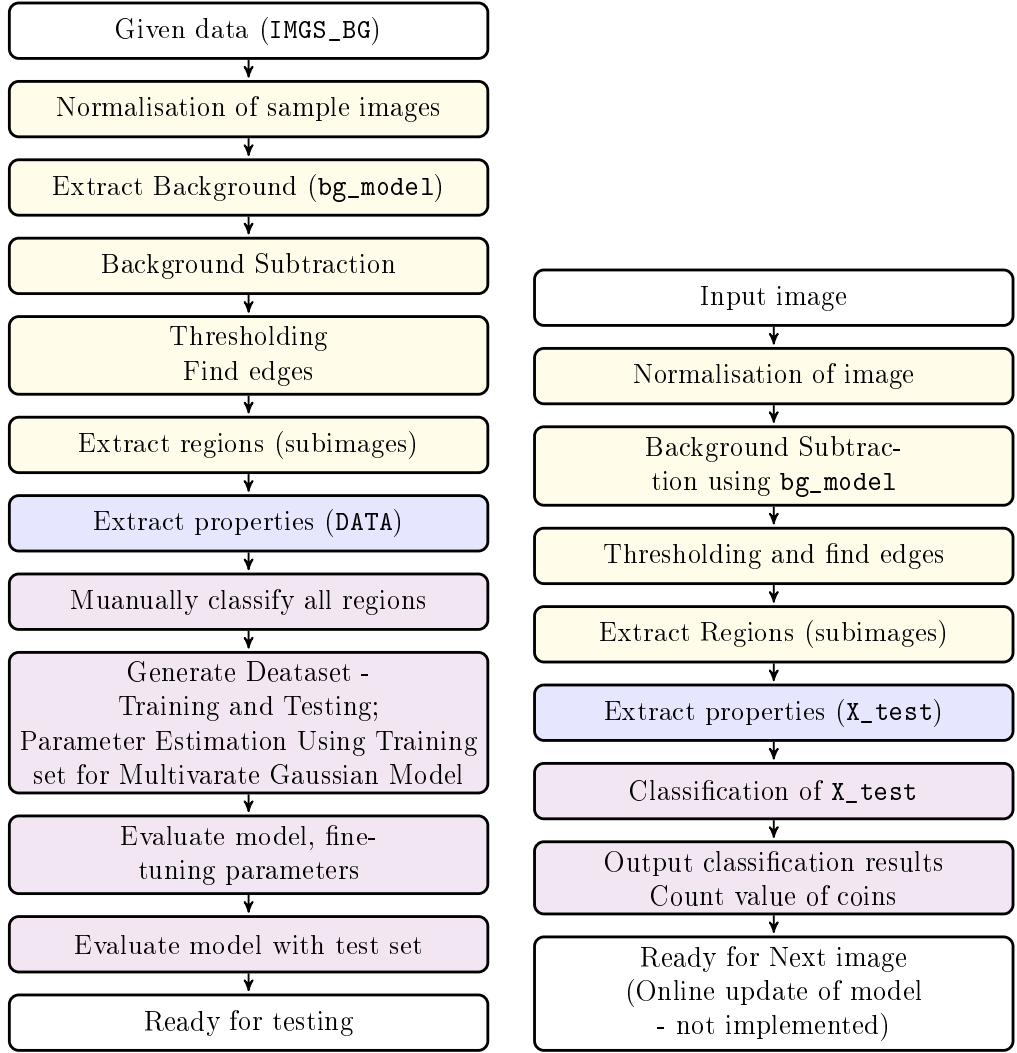
To recognise and count the coins in an image, Coinsy, have three subtasks: 1) Image processing (includes image segmentation), 2) Features extraction, and 3) Classification (and then counting the coins).

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.

Data

The data we were given are 14 images (consisting of 5 **harder** and 9 **simpler images**). Each consists of objects in the foreground for us to classify and calculate the values. The class labels and its respective object and value are as follow:



(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.

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	-
10	Nut	-
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 features extracted must be invariant to rotation, and importantly to detect the objects will the images to be processed, such that the objects are salient to the computer vision. On top of the images itself output from `imshow` or `imagesc`, we can understand an image from the distribution of the pixel intensities (such as a histogram) and its gradient magnitude in each channel.



Figure 2: Sample images from given data.



Figure 3: Sample images after their background is subtracted.

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 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 finding a suitable threshold to binarise the image; third - crop out the objects to obtain the subimages, which will be our data points.

Feature Extraction

Given the subimages, this stage aims to represent each subimage with a feature vector that contains properties to adequately describe the class it belongs to (or shape). Notably, we have many circular objects varying in size. This renders global descriptors such as convexity and elongation less useful for these classes.

Classification

The task here is to label an unknown (sub)image given its set of feature vector. The model we consider is a multivariate gaussian classifier that classifies an image based on the parameters of each class (class mean and standard deviation (or variance)). This multivariate gaussian classifier outputs the posterior probability of a given feature vector x ($P(C_k|X)$) for each class k , with the class giving the highest probability being the label. That is $x.class = \operatorname{argmax}_k P(C_k|x)$.

A slight tweak for Coinsy is her ability to reject the class label output by the classifier if the highest probability falls out of her confidence interval. In such cases, Coinsy will intervene and change the class to unknown (class 11).

The last step of Coinsy is to sum the value of all the objects she managed to identify from a given image.

Code

The following directory trees will provide an overview of the code utilised for the project. Codes presented in the appendix are hyperlinked, although some may depend on the code repository given in <http://www.inf.ed.ac.uk/teaching/courses/ivr/matlab/flatpartrecog/>.

`src`

- imgs Store images from function
- dataset Datasets from experiments
- setup
- training
- main
- image_processing; .2 extract_features
- manual_classification
- trainclf_loglikelihood
- filters
 - median_filter
 - median_filter_iter
 - gaussian_filter_1d
 - gaussian_filter_2d
 - .2 image_processing
 - normalise_RGB
 - bg_extract
 - bg_subtraction
 - dothresh
 - edg_morph_disk
- features
 - getFeatures
 - rawmmoment
 - centralmoment
 - SI_moment
 - humomentinvariants
 - complexmoment
- classification
 - split_data
 - user_classify
 - findConfusion
 - gaussianDistr
 - multivariate