Fruit and vegetable classification from live video

Lukas Danev

MInf Project Report
Master of Informatics
School of Informatics
University of Edinburgh

2017

Abstract

Fresh fruit and vegetable classification from the live video streams is motivated by a practical application for self-checkout terminals, which have become widely used in supermarkets in recent years. The look-up process of every item sold by weight is inconvenient for shoppers who are requesting a simplification. The supermarkets in the United Kingdom lose £1.6 billion every year due to stealing through the self-checkouts, of which majority of the stolen items are fresh fruits and vegetables. The whole computer vision pipeline is designed for fruit and vegetable classification in the self-checkout environment. The main stages of the pipeline are segmentation of items from background, feature extraction mainly based on colour, and classification with Gaussian Bayes classifier. The classification of items placed in a plastic see-through bag was the biggest challenge addressed, which was successfully resolved by colour clustering segmentation and feature design. With un-bagged items, the correct category was selected in the top 3 choices 100% of the time, 98% of the time with bagged items, and for the combined dataset of un-bagged and bagged items 99% of the time. The overall accuracy of the experiments was 96%, 79%, 85% respectively. The execution time of the whole classification pipeline is under 1 second, satisfying the requirement of the application at the self-checkout terminal.

Acknowledgements

I wish to express my sincere gratitude to Prof. Robert Fisher for the help and assistance during the whole duration of my project. At all stages of my project he was very patient and willing to explain concepts of computer vision, its evaluation and hyper-parameters optimisation techniques. His constructive feedback helped me progress the right direction. Towards the end he gave me very useful feedback on my report even very close to the deadline.

I thank Francisco Vargas for the suggestion on using colour clustering for segmentation task, which turn out to be very effective. I would like to thank Matus Falis, and Samuel Sucik for the supply of hot beverages and Marek Strelec for the supply of cold beverages during the endless nights of working on the project. Lastly I am grateful to Stefanie Speichert for express proofreading of the final draft.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Lukas Danev*)

Table of Contents

Chapter 1

Introduction

One of the biggest changes in shopping experience in supermarkets in the recent five years is the addition of self-checkout. While this reduced the waiting time for available cashiers and the number of staff needed, bringing benefits to both customers and the supermarket, it introduced several new issues. Items sold by weight have to be looked up by the customer in a list of all the products which do not have barcode attached. This can be very time-consuming, tedious and frustrating for the customer. Products sold by weight are mainly fresh fruits and vegetables. With the aim to ease and speed-up the self-checkout experience a solution for automated recognition and classification of fresh produce is proposed in this report.

A simple addition of a camera to the self-checkout counter, which would capture what is placed on its scale, can provide an image of the produce. Using the computer vision and machine learning techniques described below an algorithm can classify the produce into its correct category (eg.: banana, pepper, cucumber...) without further user interaction or rank the categories and offer the user the three most likely categories.

The automatisation of the recognition results in a number of benefits for both supermarkets and their customers, some of which can be related to increase of revenues. Directly measurable time each customer spends at the self-checkout would decrease. This leads to shorter waiting time in lines and bigger throughput. It is known that the current checkout process is not perfect, according to NCR study 37% of customers (3rd most frequent response out of 11) would appreciate simplification of purchasing items sold by weight [27].

The other major issue is stealing through self-checkout. It is estimated that more than 1.6 billion worth of items have been stolen last year just in the UK [3]. Of these 1.6 billion 67% were fresh fruits and vegetables. For shopper it is very tempting to buy everything as the cheapest potatoes or carrots if nobody checks. However, 57% of the cases the thefts were not intentional, the shoppers just gave up on finding the correct item. The majority of these 1.6 billion can be saved by the proposed classification algorithm which cannot be fooled and it will actually help people to not become thieves.

Indirectly, the solution leads to food waste reduction and packaging reduction. A large proportion of fruits and vegetables are being prepacked, so they do not need to be looked up in the lists and have a barcode attached. If one or two pieces in prepackaged bag gets bad, the whole bag is thrown away, including the healthy pieces. This is unnecessary waste. If all fresh produce would be sold by weight the bad pieces can be taken out and the healthy ones used. This is more sustainable, in the interest of supermarket and it offers shoppers a chance to choose the pieces he or she wants.

To achieve the goal of fruit and vegetable classification from live video a processing pipeline will be first developed on static images. Once this is in place and it is reasonably fast it can be inserted in a loop at the beginning of which a new picture is taken and fed to the processing pipeline. A high level processing pipeline is shown in figure 1.1. The first stage of static image processing decides whether the scene has changed from the default and continues only when an object is present in order not to waste computing resources. The second stage is the segmentation of the foreground representing a new object, from the background. The steps of this segmentation consist of background subtraction, morphological operations on binary images, specularities removal, colour clustering and are further discussed in section 4. It is assumed that the camera is stationary. This is a reasonable assumption to make as it would be attached to the self-checkout terminal. To identify the segmented object various characteristics called features are computed and stored in a feature vector. In the case of fresh produce the discriminative features are mainly related to the colour, but shape and texture can be determining factors as well. The feature extraction process is discussed in section 5. The final stage is the classification into correct categories using machine learning models trained on a manually labelled dataset. Gaussian Bayes classifier is the main model used for the classification as discussed in section 6. The output of the classification is an ordered ranking of possible classes from which the corresponding label is inferred. If even the highest probability is still lower than the defined threshold then the output is nothing which means that the product was not recognised as fruit or vegetable from the trained model.

In supermarkets, fruit and vegetables are often put into semi-see-through plastic bags in order to carry multiple pieces. The blurriness caused by the plastic bag presents a great challenge for the classification algorithm which, as far as known, was not addressed yet. Techniques for dealing with the plastic bag are discussed throughout the report and especially in the chapter 4 about segmentation.

Performance will be evaluated based on accuracy (top hit), average rank of the correct label and the top three hit. A standard accuracy is calculated as proportion of correctly classified images over the total number of images. The rank refers to the position of the correct label in the sorted list of predictions according to their likelihood. The top three hit considers an image to be correctly classified if its label appears in the first three most probable classes. This would be considered a success because offering a customer three options to choose from instead

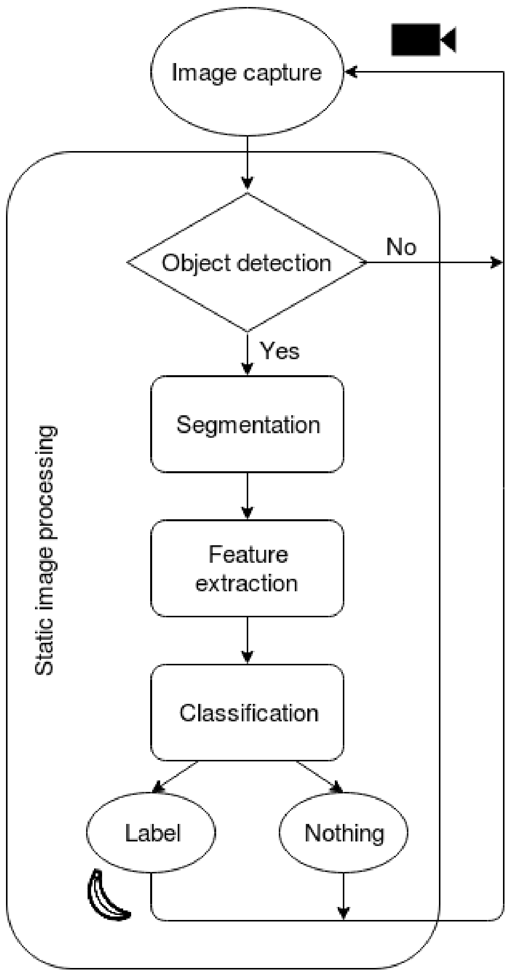


Figure 1.1: The high level pipeline of the image processing for object classification

of full list would be a huge improvement. The performance evaluation is further discussed in section 6.2.

The idea of fruit and vegetable classification from images in supermarket environment was first proposed by [20]. His work developed a classification of fruit and vegetable from static images with a white background. Most of computer vision research in agriculture was focused on quality assessing or detection for harvesting purposes. More details are discussed in section 2.1. A brief history of cash register, groceries checkout and how they evolved to a modern self-checkout is discussed in section 2.2.

1.1 Contributions

- The classification of the fresh produce placed into a plastic semi-see-through bag. More specifically, the demonstration that clustering of the colours is an effective segmentation technique to distinguish the item in the plastic bag and the plastic bag itself.
- The design and evaluation of the segmentation pipeline, colour based feature extraction pipeline and the classification with a Gaussian Bayes classifier.
- The assignment of prior contribution weights to pixels, which reflect the expected location of an item and reduce the effect of segmentation imperfections.

1.2 Previous work carried out in MInf Project Part 1

In the MInf4 project, or MInf Project Part 1, the exploratory analysis of the brains resting state functional connectivity in fMRI data [5] was conducted . This topic is unrelated to the MInf5 project and a concession was obtained for MInf Project (Part 2) (INFR11093), to vary the project topic from that undertaken in Part 1, as approved by the course organiser and supervisor. The MInf5 project report is self-contained, and ,therefore, there is no need to put the MInf5 work in the context of the MInf4 work.

Chapter 2

Background

2.1 Computer vision in fresh produce industry

In the context of the fresh produce industry, computer vision techniques are mainly used for fruit detection for harvesting purposes or for quality evaluation [26]. An automatic spherical fruits recognition system in the natural conditions is proposed by [15]. A combination of a laser range-finder model and a dual colour and shape model is used, targeting mainly oranges, apples and other citruses.

Quality evaluation based on visual appearance is discussed in [12], who used the Hue, Saturation and Intensity colour model to grade the quality of apples. [6] examined how the reflectance properties of carrots and celeriac changes over a period of 14 days. The aim was to create an alternative quality control in the food industry to the usual analysis of chemical compounds extracted from the produce items. [18] developed automatic detection of skin defects in citrus fruits based mainly on their texture features. All of these studies have in common that they examine particular properties of one class of fruit or vegetable.

Fruit and vegetable recognition by fusing colour and texture features was proposed by [7]. Their design relies on the background subtraction and classification with a multi-class support vector machine. [17] used fusion of multiple colour channels for multi-class fruit detection from pictures with highly complex backgrounds. However, the bottleneck is the detection speed which takes 28 seconds to process one image. Colour and texture features were derived for fruit recognition using the minimum distance classifier [2]. The texture features are based on the sum and difference histogram of the neighbouring pixels and are presented by [8].

With an exception of [17], all the other papers used images of fruits and vegetables on a white background. They report that the main discriminative features between the different classes of fruit and vegetable are based on colour and texture.

2.2 Self-checkout

The shopping experience has evolved rapidly over the past century and is being heavily transformed by new technologies and increasing customer expectations. A necessary part of each shopping trip is the payment for the products at the checkout where all the items are reviewed and the final price is calculated. Shops are growing larger offering a wider range of products and, as the number of purchased items was gradually increasing, there was a need for automation of the checkout process. One of the first advances of technology is the mechanical cash register invented in the late 19th century [4]. In 1980s the machine readable barcode system was developed for labelling products with universal identification code which speed up the checkout in food chain stores [28]. This was a big step forward as instead of entering each item code manually it was scanned in an instant. However, not all products can be labelled with a barcode - the main examples are fresh produce, sold by weight, and bread. For these products the identification code has to be entered manually even today.

The idea of an automated self-service checkout system was first patented in 1988 and its main characteristic are very similar to the current self-checkouts [16]. In the proposal from 1988 the issue of unlabelled items remained untouched. This was addressed in a patent from 1990 which describes self-checkout of produce items [13]. The solution consisted of a touch screen interface through which a customer chooses which kind of fruit or vegetable he or she is buying. A diagram taken from the original patent description is shown in figure 2.1. The system is very comparable to what is used in practice today. Five years later the same inventor patented an upgrade of his original idea which involves a video camera recording the scale of the self-checkout counter, transmitting the video stream to the screen of a human operator, who sees video streams from several counters, and manually types in the identification codes of produce [14]. The approach in this paper is building on top of the video camera idea, but, instead of the human operator, a classification algorithm is designed.

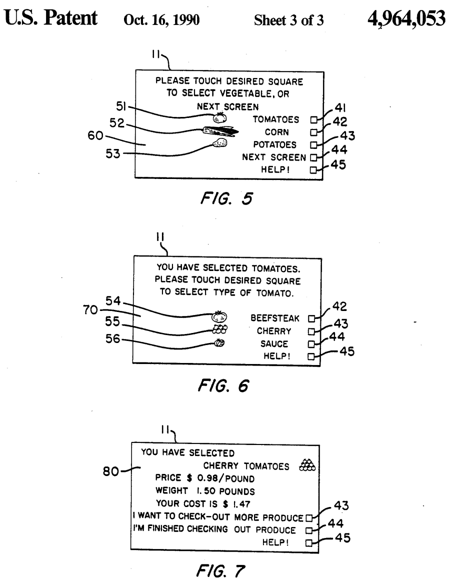

U.S. Patent Oct. 16, 1990 Sheet 3 of 3 4,964,053
PLEASE TOUCH DESIRED SQUARE TO SELECT VEGETABLE, OR NEXT SCREEN
TOMATOES
CORN
POTATOES
NEXT SCREEN
HELP!
FIG. 5
YOU HAVE SELECTED TOMATOES. PLEASE TOUCH DESIRED SQUARE TO SELECT TYPE OF TOMATO.
BEEFSTEAK
CHERRY
SAUCE
HELP!
FIG. 6
YOU HAVE SELECTED CHERRY TOMATOES
PRICE $ 0.98/POUND
WEIGHT 1.50 POUNDS
YOUR COST IS $ 1.47
I WANT TO CHECK-OUT MORE PRODUCE
I'M FINISHED CHECKING OUT PRODUCE
HELP!
FIG. 7

Figure 2.1: Diagram explaining the self-checkout process of produce items patented in 1990 [13] using touch screen interface

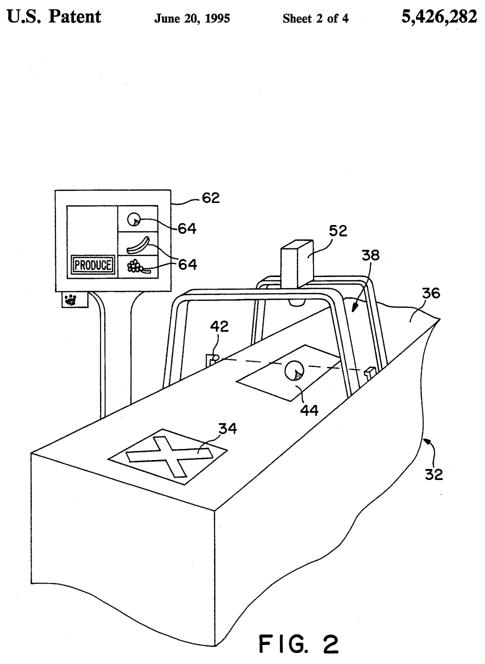


Figure 2.2: Diagram explaining the self-checkout process of produce items patented in 1995 [14] using a video camera coupled to a service terminal where a human operator types the identification codes of produce

Chapter 3

Dataset acquisition

The dataset has been acquired in two iterations. The three main differences between the first and the second iteration are the number of fruit and vegetable classes, the background of the images, and the type of the camera. In both iterations fruits and vegetables were captured un-bagged and bagged in see through plastic bags. The summary of the number of classes and images for each iteration is shown in table 3.1.

3.1 Iteration 1

The first iteration contains six classes, namely banana, cucumber, grapefruit, kiwi, pepper, and tangerine. Each class was made from ten pictures of un-bagged items and ten pictures of bagged items with the exception of cucumber. This yields 60 un-bagged and 50 bagged images of fruit and vegetables. The example pictures of un-bagged items are shown in figure 3.1 and of the bagged items in figure 3.2.

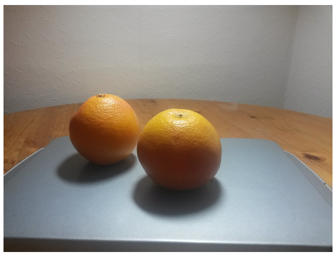
The background is composed of three main regions - a chrome tray which imitates the scale at a self-checkout terminal, a wooden table, and a white wall with a light structure. Two pictures were captured containing only the background, without any items present. These are later used for background subtraction. This process is further described in section 4.2. The images were taken from a static view point, so ,ideally, the background should be identical in all images. The fruit and vegetable items are always placed on the tray, but the position, orientation, and the number of items are varying across the images. The images has been captured with a rear camera of the Samsung Galaxy S4 mini and with a resolution set to 2048x1536 pixels.

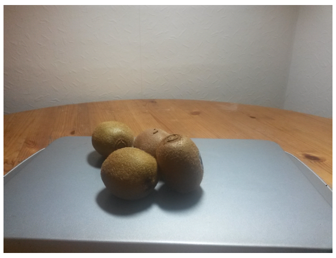
	Iteration 1		**Iteration 2**	
	un-bagged	**bagged**	**un-bagged**	**bagged**
banana	10	10	25	25
cucumber	10	-	25	25
grapefruit	10	10	25	25
kiwi	10	10	25	25
pepper	10	10	25	25
tangerine	10	10	25	25
grapes	-	-	25	25
lemon	-	-	25	25
melon	-	-	25	25
onion	-	-	25	25
carrot	-	-	25	25
potato	-	-	25	25
garlic	-	-	25	25
tomato	-	-	25	25
Total:	**60**	**50**	**350**	**350**

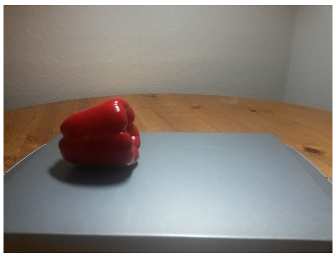
Table 3.1: The distribution of 810 images acquired in both iterations for each class.

(a) banana

(b) cucmber


(c) grapefruit


(d) kiwi


(e) pepper

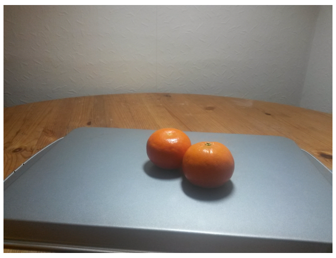
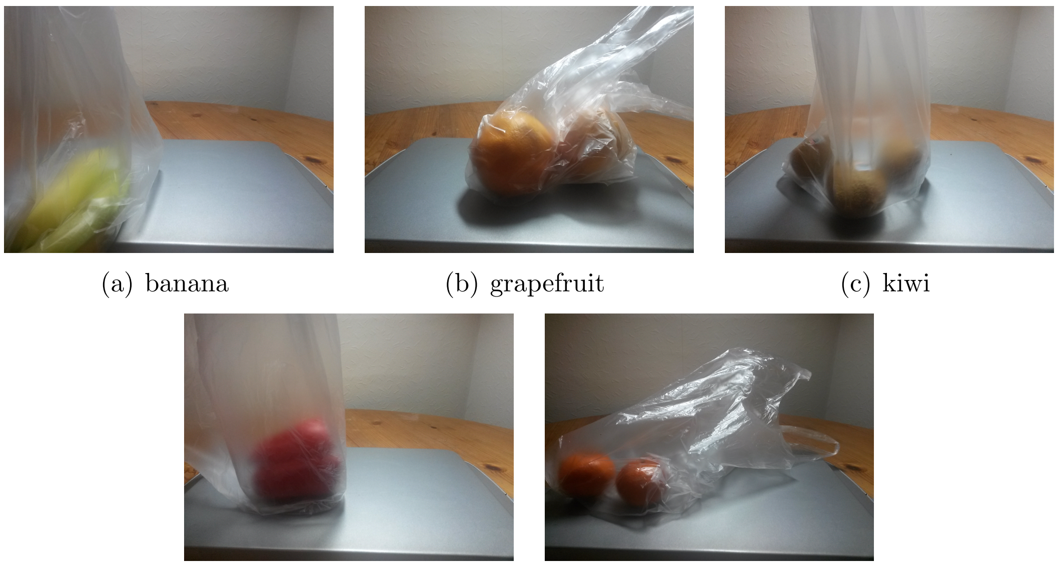

(f) tangerine

Figure 3.1: Examples of un-bagged fruit and vegetables from the first iteration.



(a) banana (b) grapefruit (c) kiwi

(d) peper (e) tangerine

Figure 3.2: Examples of bagged fruit and vegetables from the first iteration.

3.2 Iteration 2

In the second iteration the 6 classes from the iteration 1 were expanded to 14 classes. The newly introduced classes are carrot, garlic, grapes, lemon, melon, onion, potato, and tomato. The examples of un-bagged and bagged items are shown in figures 3.5 and 3.6 respectively.

The background is actually a bit simpler compared to the first iteration. It contains the same chrome tray, a white wall which does not have the structures in this case, and a colour sticker at the bottom. The colour sticker is intended to be used for colour correction under varying lighting conditions. The wooden table which was a part of the background in the first iteration is not included in the second, because it is not representative of the conditions of self-checkout in a supermarket. This time, the images have been captured by a web camera of the model AUSDOM AF225 connected to a computer with a resolution of 640x480 pixels. The detailed setup including a description of a small application developed for image capturing is in section 3.2.1 below.

3.2.1 Micro-app for dataset acquisition

A systematic approach was used to create the dataset in the second iteration. To avoid confusion, each image contains in its name the type of the item, whether it is a bagged item, and a reference to the background image. The reference to the background image which is an image without any item, allows for different backgrounds for different images. This is useful if the lighting condition changes or something in the background moves. For 700 images 26 backgrounds were created.

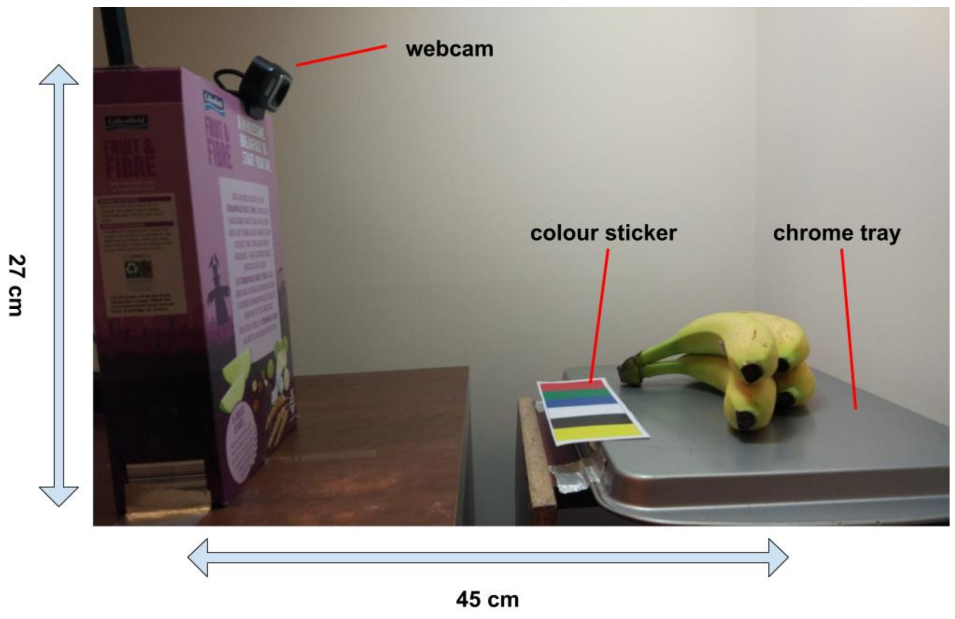


Figure 3.3: The apparatus for capturing images for the second iteration of the dataset acquisition. The webcam is sensing the tray at an angle approximately 37 degrees.

A small Python application was implemented to automatically satisfy the requirements described above. The image capturing capabilities were adapted from [1]. The graphical user interface is shown in figure 3.4. It contains a live stream from the webcam, two standard buttons for capturing either background or an item image, and 14 radio buttons for selecting the type of the item. It automatically generates the descriptive title for the image, with an exception whether a bagged/un-bagged item which was hard-coded, and keeps track of all the images, their types and backgrounds created.

In order to achieve stable conditions the webcam was taped to the top of the box (taped to the table), from which it was sensing the tray at a 37 degrees angle. The apparatus with dimensions and labels is shown in figure 3.3.

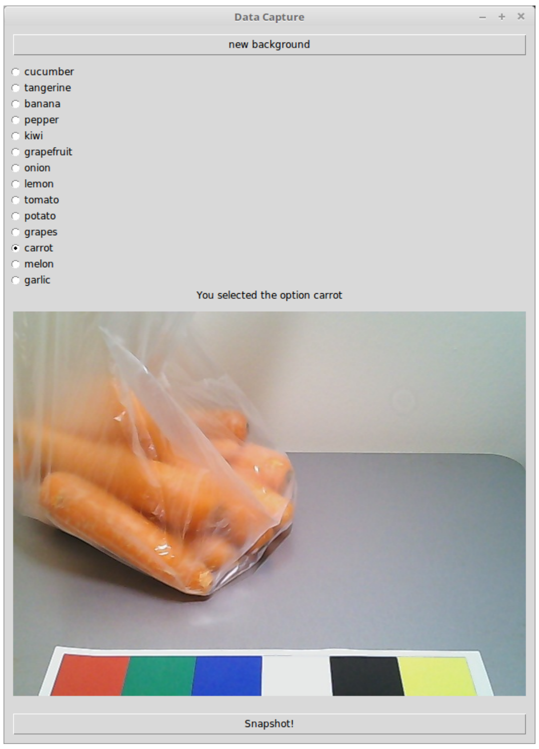


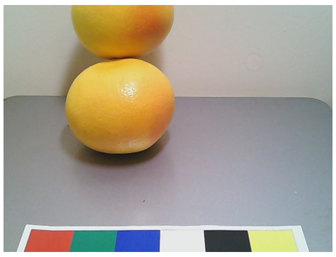
Figure 3.4: Interface for dataset acquisition.

(a) banana

(b) carrot

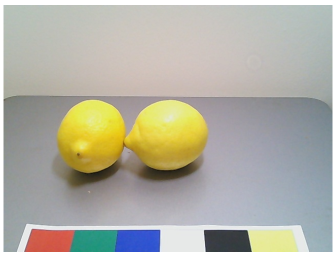
(c) cucumber

(d) garlic


(e) grapefruit

(f) grapes

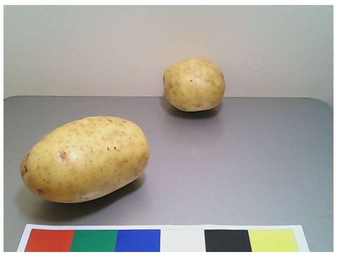
(g) kiwi

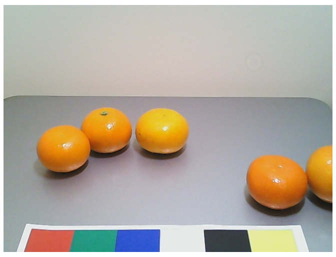

(h) lemon

(i) melon

(j) onion

(k) pepper


(l) potato


(m) tangerine

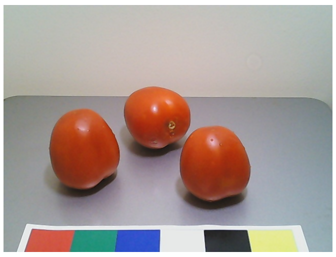

(n) tomato

Figure 3.5: Examples of un-bagged fruit and vegetables from the second iteration.

(a) banana

(b) carrot

(c) cucumber

(d) garlic

(e) grapefruit

(f) grapes

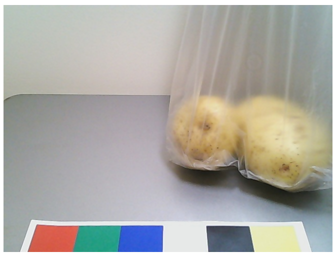
(g) kiwi

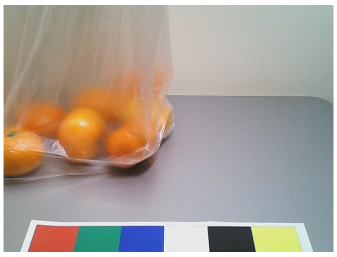
(h) lemon

(i) melon

(j) onion

(k) pepper


(l) potato


(m) tangerine

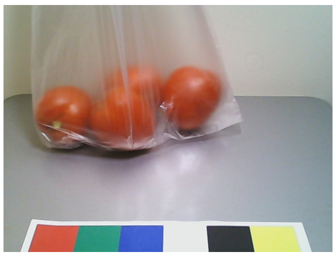

(n) tomato

Figure 3.6: Examples of bagged fruit and vegetables from the second iteration.

Chapter 4

Segmentation

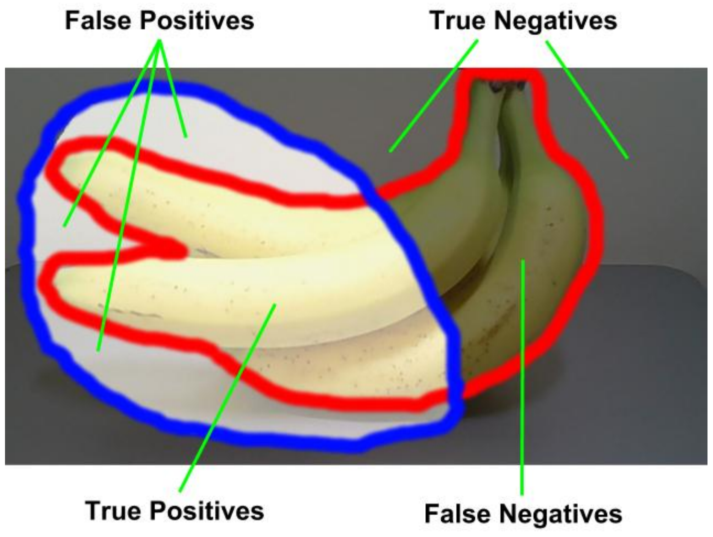
By definition, segmentation is a process of grouping the image pixels into meaningful structures [10]. It is the first stage of static image processing which leads to the fruit and vegetable classification. Its main goal is to identify regions of an image where a fruit or vegetable item appears, more precisely, to identify pixels belonging to the item. These pixels are used in later stages to calculate characteristic features for a classifier. A set of segmentation techniques used to segment items is described in the following subsections.

4.1 Segmentation evaluation

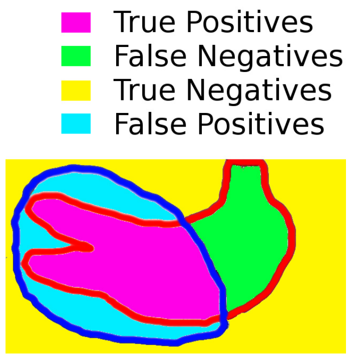
The goal of segmentation in this task is to decide for each pixel whether it belongs to an item or background. The quality of the segmentation can be evaluated by looking at the proportion of correctly classified pixels.

Segmented pixels can be divided into 4 categories, which are demonstrated in figure 4.1. In the figure, the red contour encloses the ground truth segmentation of bananas (fruit pixels) and the blue contour encloses the region, which is detected by the segmentation algorithm. The first category of pixels are true positives (TP), which are pixels detected that actually belong to the fruit or vegetable. True negatives (TN) are pixels, which are correctly undetected because they belong to the background. False positives (FP) are pixels, which are detected by mistake and actually belong to the background. The last category is false negatives (FN), which are pixels, which are not detected but belong to the fruit or vegetable. Abbreviations TP, FN, TN, and FP denote the number of pixels in each category.

TP, FN, TN, and FP represent the absolute numbers of pixels. A more comparable metric is rate which is relative and on range between 0 and 1. Formulas for calculating the ranges are shown in figure 4.2 [9]. The True Positive Rate (TPR) is a proportion between the correctly detected pixels and all the foreground (fruit and vegetable) pixels, showing how much of an item is detected.



(a)



(b)

Figure 4.1: A model demonstration of TP, TN, FP, and FN regions. The blue contour shows detected region, the red contour shows the ground truth segmentation.

TPR is also referred to as recall and sensitivity. True Negative Rate (TNR), or specificity, is a proportion between the correctly undetected background pixels and all the background pixels. False Positive Rate (FPR) is a proportion between the incorrectly detected background pixels and all the background pixels. False Negative Rate (FNR) is a proportion between the undetected foreground pixels and all foreground pixels. TNR and FPR are complements of each other and they always sum to one. Similarly, TPR and FNR are complements, as well. A segmentation algorithm is aiming for high TPR and TNR which implies low FPR and FNR.

The segmentation analysis described above requires a ground truth segmentation to be known. For this reason 90 images, 60 un-bagged and 30 bagged, from the first iteration were manually annotated with the ground truth mask. The masks were created using GIMP, an image manipulation program, by colouring the fruits and vegetables regions white and the background black. Example masks with reference pictures are shown in figure 4.3. It is to be noted that the ground truth segmentation of bagged items contains only the items inside the plastic bag, not the plastic bag around them.

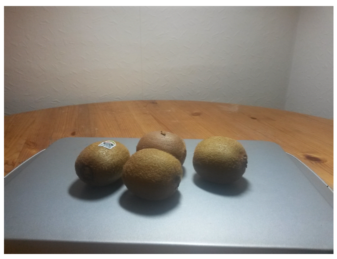
4.2 Background subtraction

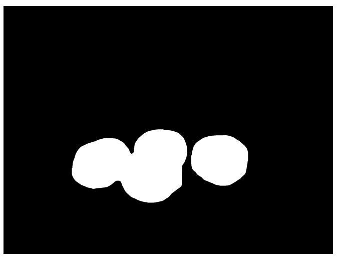
Background subtraction, also known as foreground detection, is an effective foreground segmentation technique if the background image is available. In the self-checkout setting this is the case as the background image is just an image of an empty scale. The core idea of background subtraction is to compare the known

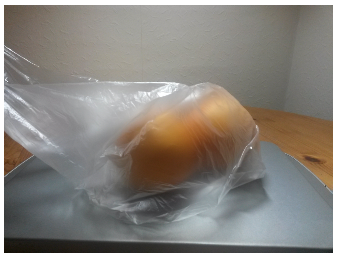
$$TPR = \frac{TP}{TP + FN} \qquad FNR = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{TN + FP} \qquad FPR = \frac{FP}{FP + TN}$$

Figure 4.2: Formulas for calculating rates - TP(R) True Positives (Rate), FN(R) False Negatives (Rate), TN(R) True Negatives (Rate), FP(R) False Positives (Rate)


(a) kiwi - original image


(b) kiwi - ground truth mask


(c) grapefruit - original image

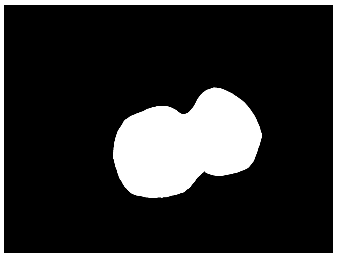
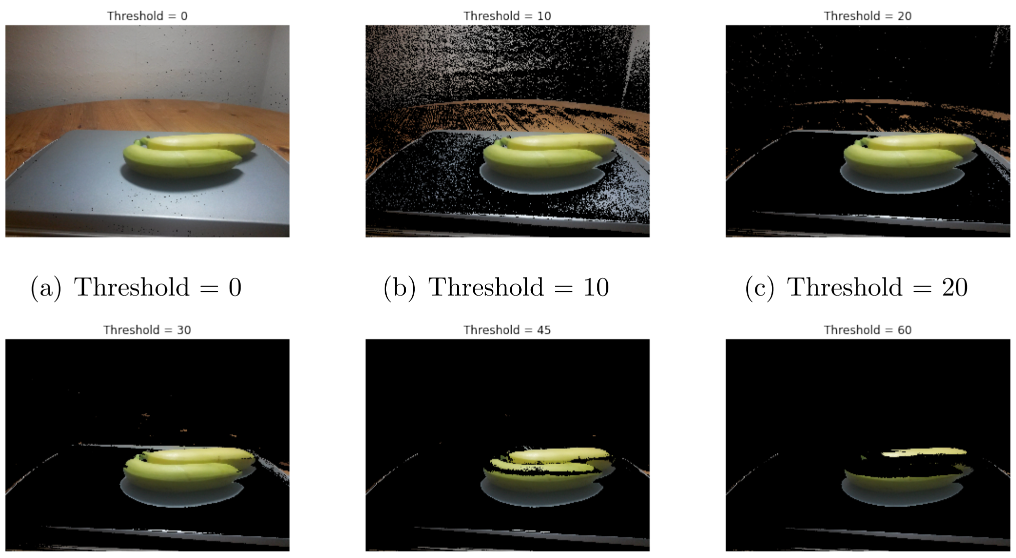

(d) grapefruit - ground truth mask

Figure 4.3: Examples of ground truth segmentation of un-bagged (4.3(a), 4.3(b)) and bagged (4.3(c), 4.3(d)) items.



(a) Threshold = 0 (b) Threshold = 10 (c) Threshold = 20

(d) Threshold = 30 (e) Threshold = 45 (f) Threshold = 60

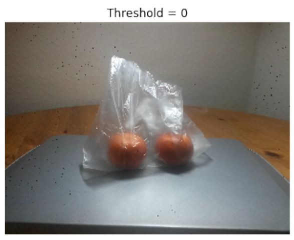
Figure 4.4: Background subtraction for varying threshold

background image with an unknown image on a pixel level and to keep only the pixels of the unknown images which are different from the corresponding background pixels by a defined threshold. In computer vision terms RGB pixel values of the background are subtracted from the RGB values of the image. An RGB pixel value represents the red(R), green(G) and blue(B) channel of a specific pixel. This substraction is shown in equation 4.1 where the absolute value is taken of the RGB pixel values difference (subscript denotes dimensions' width, height and 3 layers corresponding to RGB layer). The pixels' differences are summed into one layer and by checking for every element if the difference is greater than a given threshold, the layer is converted into a binary mask, see equation 4.2. The threshold has range 0 to 255, corresponding to the average value of RGB elements of a pixel, and, as the comparison is against the sum of 3 layers, the threshold is multiplied by 3. This mask is then applied to all three RGB layers of the original image to extract just the detected pixels. Results of applying background subtraction for six different thresholds are shown in figure 4.4 - some pixels are removed even for threshold zero, figure 4.4(a). This is because their value was exactly the same as the background.

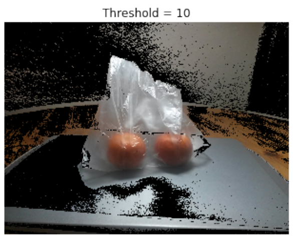
$$pixels\ differences_{(\mathbf{w},\mathbf{h},3)} = \|current\ image_{(\mathbf{w},\mathbf{h},3)} - background_{(\mathbf{w},\mathbf{h},3)}\| \tag{4.1}$$

$$mask_{(\mathbf{w},\mathbf{h},1)} = (\sum_{rgb=0}^{2} pixels\ differences_{(\mathbf{w},\mathbf{h},rgb)}) > (3 * threshold) \tag{4.2}$$

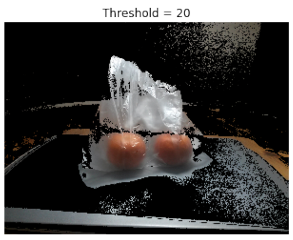
The quality of the segmentation by background subtraction is evaluated by calculating the rates described in section 4.1. The Receiver Operating Characteristic



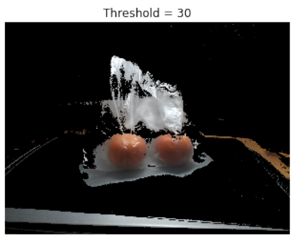
(a) Threshold = 0



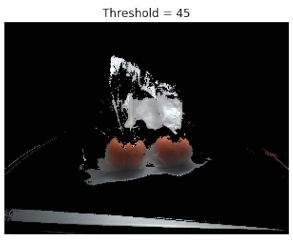
(b) Threshold = 10



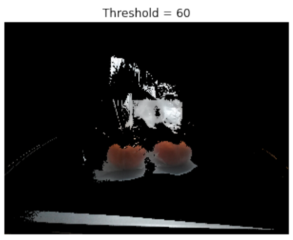
(c) Threshold = 20



(d) Threshold = 30



(e) Threshold = 45



(f) Threshold = 60

Figure 4.5: Background subtraction for varying threshold of items in a plastic bag

curve (ROC curve) is a plot of TPR against FPR at various background subtraction thresholds which illustrates the performance of binary classifier (foreground vs. background). The ROC curve which is averaged over 60 images of un-bagged fruit and vegetables from the first iteration is shown in figure 4.6. A favourable position on the plot is top left corner where the TPR is high and FPR is low. An alternative visualisation is shown in figure 4.7, which shows TPR and TNR (equivalent to 1-FPR) at various thresholds. TPR and TNR cross at threshold 40 suggesting that this is the optimal threshold. The threshold 40 is used as a starting point, however it might not be the most optimal one from the perspective of the whole fruit and vegetable classification pipeline. At threshold 40 85% of fruit pixels are preserved, but there is still 15% of unwanted FPs. Sacrificing 15% of TPR by moving the threshold to 50, reduces FPR to 5% which might be better for the classifier as there would be less confusing background pixels.

4.3 Specular reflection elimination

The next stage of fruit and vegetable segmentation from the background is the removal of specular reflection. Specularities, or specular reflections, are very bright spots caused by direct reflection of light towards the camera. Specularities depend on the orientation of the surface, angle between the light source, the point on the surface and the camera, and the material properties such as reflectance or ”shininess” (proportion of the absorbed and reflected light) [29], [30]. Once an image is captured by a digital camera, specularities appear as white or very bight pixels. White colour is in RGB colour space represented as (255,255,255) and

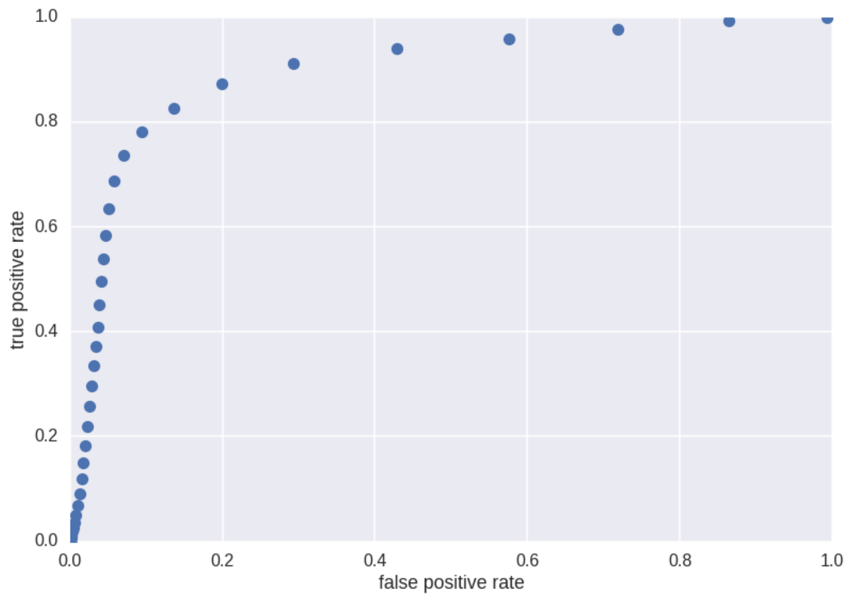


Figure 4.6: ROC curve - TPR against FPR at various background subtraction thresholds, averaged over 60 images of un-bagged fruit and vegetables.

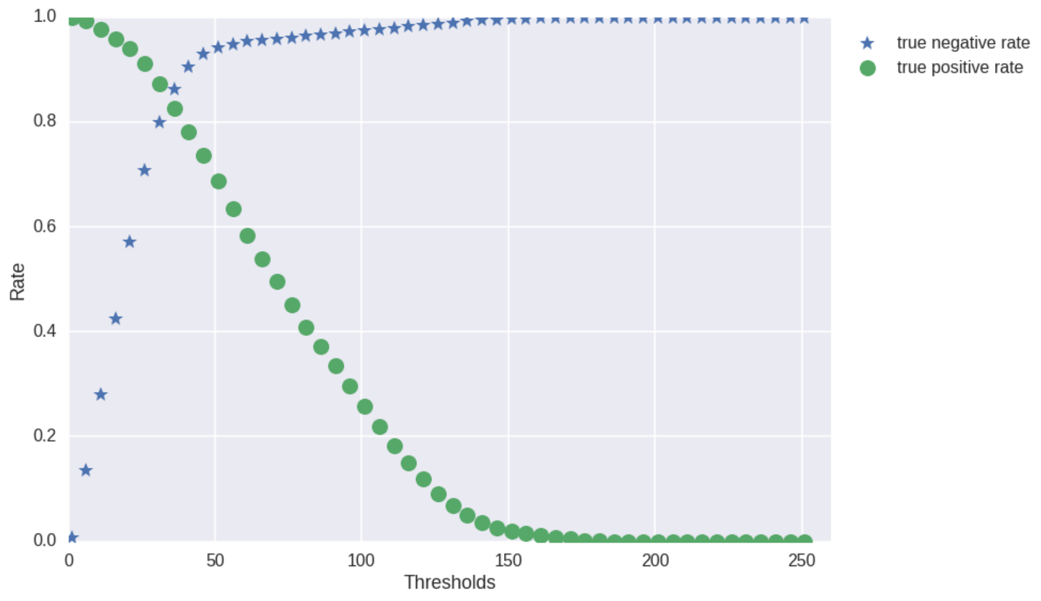


Figure 4.7: TPR and TNR at various background subtraction thresholds, averaged over 60 images of un-bagged fruit and vegetables. TPR and TNR cross at threshold 40.

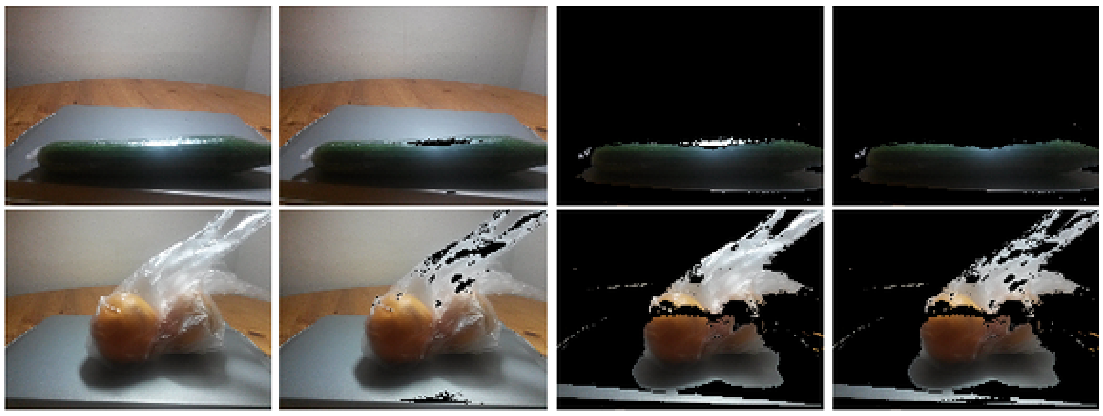


Figure 4.8: Examples of specularities removal. The first column contains the original image, specularities removal applied to it is shown in the second column. The third column shows the original image after background subtraction and the forth column specularities removal applied to the subtracted image.

thus the specularities pixels all have high RGB values. Empirical measurement showed that specularities have average RGB values in each colour channel above 230, so all pixels whose sum of RGB values is greater than 690 are discarded by specularities removal. The segmentation evaluation as described in section 4.1 does not apply to specularities removal because here items' pixels are intentionally discarded.

There are two main reasons why specularities removal is useful. Specularities often appear on a plastic bag, as it is a reflective material, and removing these specularities means removing parts of unwanted plastic bag which was not removed by the background subtraction. Depending on the surface of the item, it can contains some specularities as well. Specularities are present for citrus fruits, melons, and peppers. On the contrary, almost no specularities occur on kiwis or carrots. The extreme value pixels (specularities) are confusing for colour features described in section 5.1 as they are not representative for the colour of the fruit. By removing these specularities, holes can be introduced to a segmented item, but that is not a major problem. Examples of un-bagged and bagged items before and after specularities removal are shown in figure 4.8.

It is arguable, at which stages of segmentation the specularities removal should be applied. Typically, it should be the last stage before the features calculation. However, it might be sensible to also use this technique before applying the morphological operation described in section 4.4. Removing specularities removes parts of the bag, which can help the morphological operators to remove the left overs of the bag. An argument against applying specularities removal before morphology is that it can introduce holes to the fruit or vegetable which might be enlarged by erosion. Based on the observations, the segmentation is better when the specularities removal is applied before the morphological operations as well as at the last stage of segmentation. This is due to the fact that the proportion of the specular reflection is much lower compared to the reflections on the plastic bag (some items, like kiwi, does not produce any specular reflection).

4.4 Morphological operations on binary images

The background subtraction removes most of the background and preserves most of the item. However, there are usually a lot of unwanted background pixels spread across the image, these are called noise. They can be removed by mathematical morphology operations, which are applied on the binary mask of the subtracted image. Mathematical morphology, or just morphology, is a field that analyses spatial structures, their shape and form. It is based on set theory, integral geometry, and lattice algebra. It aims at extracting relevant structures of the image [23]. The two fundamental morphological operators are erosion and dilation. Morphology can be applied to grey-scale images, but for segmentation purposes just the binary mask is appropriate. Two inputs are required - the binary image and a structuring element also called kernel. The kernel characterises the nature of the shape. In order to remove noise and preserve the item an isotropic kernel is used, which behaves the same way in all direction. It is a square matrix filled with ones.

The erosion operator shrinks or reduces regions extracted by the background subtraction. It erodes away the boundaries of the foreground object. During erosion the kernel slides through the image. From the pixels which are 1 in the binary image are kept only those for which all the pixels under the kernel are 1. In mathematical terms the erosion combines two sets (the image and the kernel) using vector subtraction of a set of elements, as defined by equation 4.3 ($\ominus$ denotes erosion)[24].

$$X \ominus K = \{p \in \varepsilon^2 : p + k \in X \ for \ every \ k \in K\} \tag{4.3}$$

The dilation is the dual operator of the erosion, it fills or grows the regions. It changes all background pixels neighbouring the object to object pixels. Similarly as during the erosion, the kernel slides through the image and whenever at least one pixel under the kernel is 1, all the pixels under the kernel will be turned into 1. Mathematically, dilation combines two sets using vector addition as defined by equation 4.4 ($\oplus$ denotes dilation)[24].

$$X \oplus K = \{p \in \varepsilon^2 : p = x + k, x \in X \ and \ k \in K\} \tag{4.4}$$

The erosion and dilation operators are the primary morphological operators. Combining them and/or applying them multiple times forms other operators. In cases like noise removal an opening operation is used which is erosion followed by dilation. An opposite of opening is closing, which is dilation followed by erosion.

The inspection of the ROC curve and the images after applying erosion and dilation multiple times showed that good results are achieved using a kernel of 4 by 4, filled with ones and consecutively applying erosion 15 times, followed by 15 iterations of dilation. The masks before and after each morphological operation

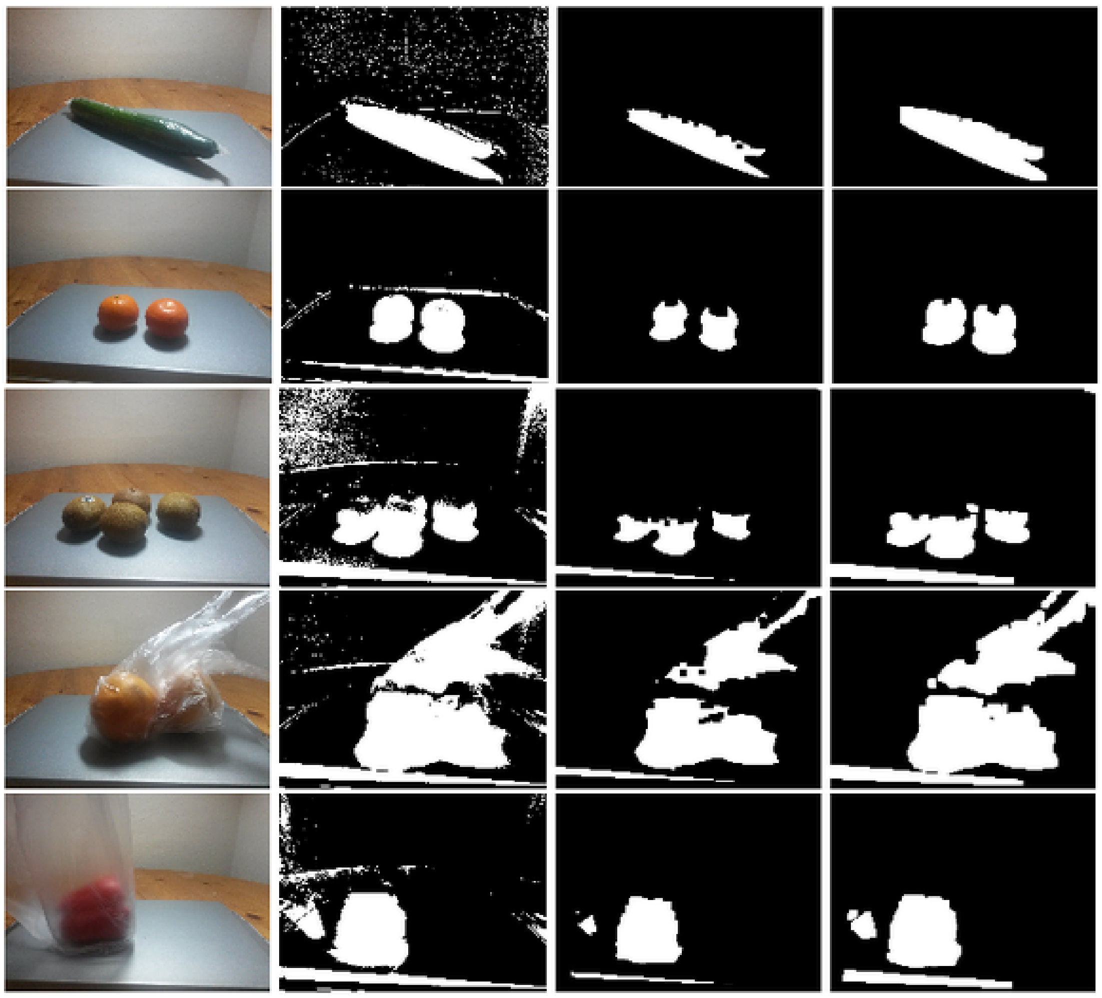
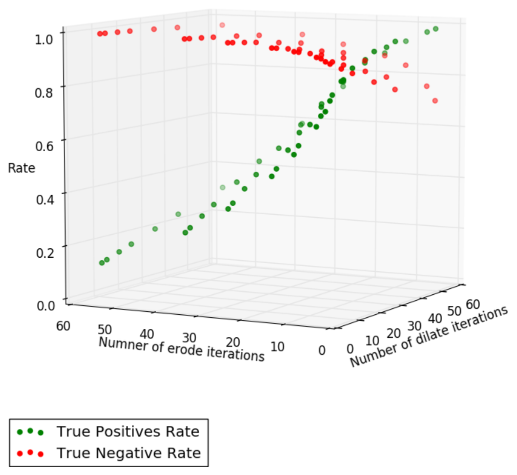
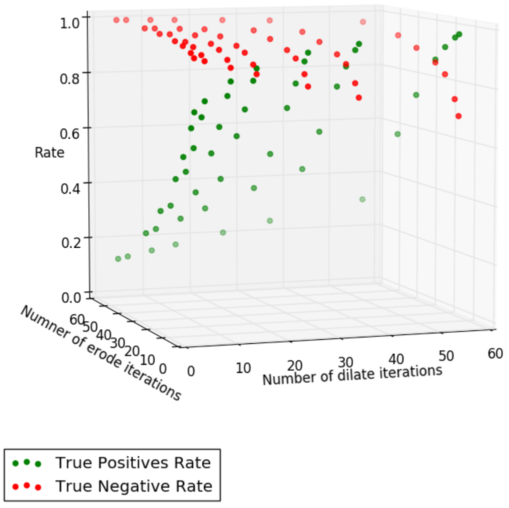


Figure 4.9: Examples of applying morphological operations. The first column contains the original image, the second image is the mask after the background subtraction, the third column is after erosion with 15 iterations, and the forth columns is after dilation with 15 iterations. In the first two cases the noise was removed successfully, in the other three cases morphology was less effective.

with reference images are shown in figure 4.9. The TPR and TNR relation for varying number of iterations of both erosion followed by dilation for 60 images of un-bagged items is shown in figure 4.10. Obviously, high numbers of iterations are rapidly decreasing one of the rates while maximising the other. The optimal number of iterations, which maximises both rates, is somewhere between 10 to 20 iterations. The same relation as in figure 4.10 is shown in figure 4.11 but rates are averaged over 30 images of bagged items. Here the rates drop faster with increasing numbers of iterations and are generally lower. This is intuitive as a large part of the plastic bag is preserved from background subtraction even after specularities removal described in section 4.3 which was applied prior to morphological operations.

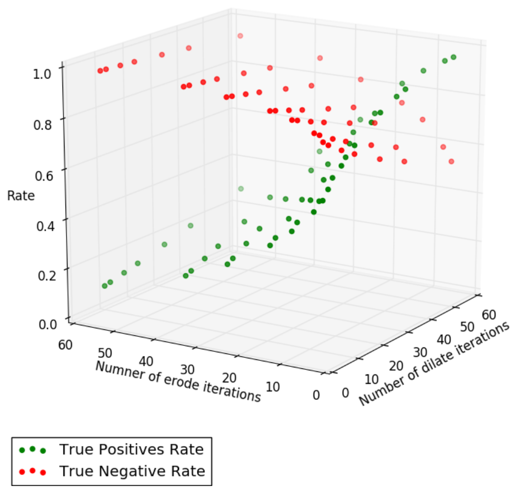


(a)

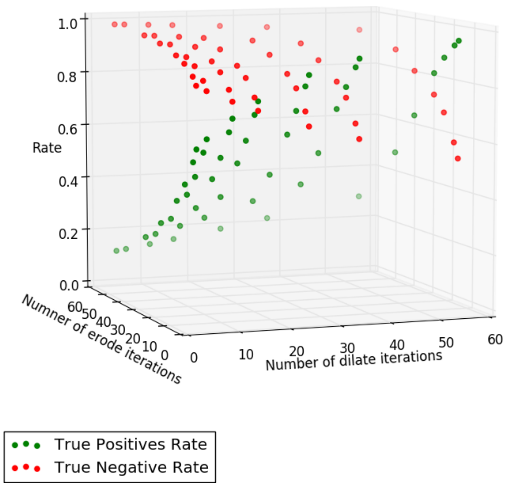


(b)

Figure 4.10: TPR and TNR relation for a varying number of erosion iterations followed by a varying number of dilation. The rates are average over 60 images of un-bagged items. The sub-figures 4.10(a) and 4.10(b) shows the same plot but from a different point of view.



(a)



(b)

Figure 4.11: TPR and TNR relation for a varying number of erosion iterations followed by a varying number of dilation. The rates are average over 30 images of bagged items. The sub-figures 4.11(a) and 4.11(b) shows the same plot but from a different point of view.

4.5 Colour clustering

The segmentation techniques described above perform well on the images of unbagged items. Even though the specularities removal 4.3, removes the shiny parts of the bag the majority of it still remains. This is addressed by colour clustering of the pixels. Clustering is a process in which a dataset is replaced by clusters, which are collections of data points that belong together [11]. The segmentation process, in fact, groups the pixels which belong together as well. For the clustering, the data is represented as feature vectors between which a distance can be computed. The clustering algorithm assigns data points into clusters so that the distance among the data points in the same cluster is minimised.

A common clustering algorithm is called k-means clustering. The k represents the number of clusters, e.g. the number of groups, into which the data is clustered and it is a parameter of the algorithm. A centroid represents the middle or the mean of a cluster. There are various methods to initialise the k centroids. One could, for example, assign them values of the extreme data points or randomly assign data points to the centroids and compute the mean of each cluster (data points assigned to the centroid). Then iteratively all the points are assigned to the closest centroid, the mean of all clusters is calculated and the centroid is moved to the calculated mean. This procedure is repeated until either the maximal number of iterations is reached or the assignment of the points to clusters does not change after an iteration. The later is called convergence.

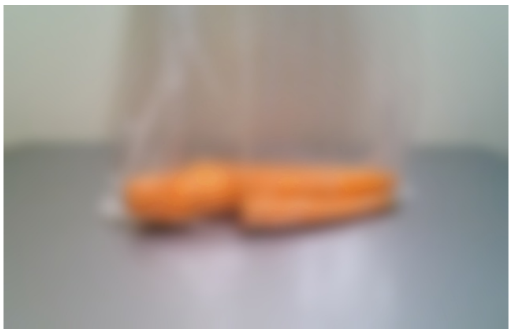
The colour of a pixel is characterised by its intensities values which are usually represented by 3 channels, see section 5.1 about colour representation. For clustering purposes, it is important that the distances between the pixels are small for pixels of similar colour and the distance is increasing as the colours gets dissimilar. This property is best satisfied by hue, saturation, and value (HSV) colour space described in section 5.1.3. To avoid confusing the clustering algorithm with a possible extreme values of the pixels, the image is blurred with a Gaussian filter. Also, the colour sticker which is in the second dataset is cut out.

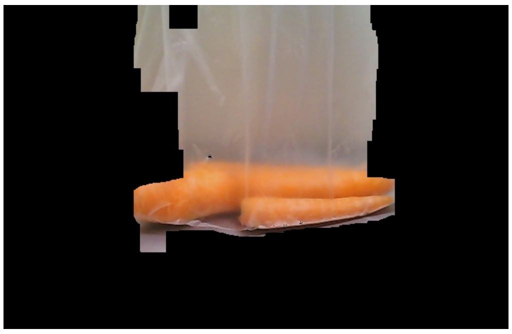
There are several approaches how the colour clustering can be used for segmentation in context of fruit and vegetable segmentation. One is to consider only the pixels which are already segmented by the techniques described above and cluster the pixels into two clusters. This should result in clusters of the item's pixels and the background, or, more specifically, the plastic bag. This approach proved to be problematic when the original segmentation was already efficient (only item's pixels are considered), as it is unable to distinguish that both of the resulting clusters already belong to the item. The second approach is to cluster all the pixels of the image, then choose only those which were segmented by the standard segmentation techniques and then select the cluster belonging to the item. These steps are visualised in figure 4.12. There are two important considerations: how many clusters should be used and how to distinguish which cluster actually belongs to the fruit.

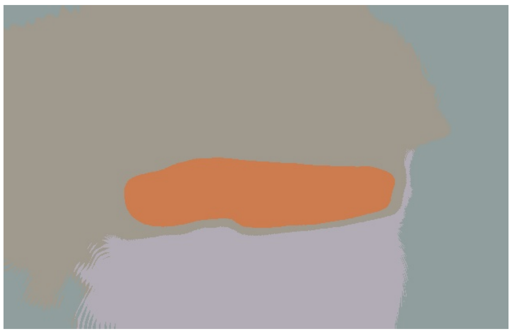
To select the number of clusters a prior knowledge about the image is used. The image usually contains about four main regions of similar colour - the item, the tray, the wall, and the plastic bag and/or wooden table if the dataset 1 is used. Empirically, it was found that four clusters works well for un-bagged items as well because the tray does not have consistent colour and is usually clustered into two clusters or mixed up with the wall pixels as it is the case in figure 8.2.

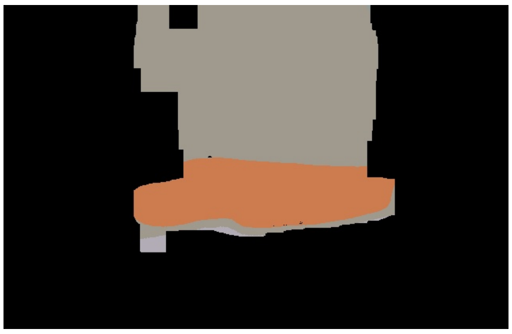
Distinguishing which cluster belong to the item and which to the background is a challenging task. A naive approach is to take the cluster which has the most pixels overlapped with the segmented region. This approach resulted in catastrophic results in many cases when the wrong cluster was selected and none of the item's pixels were preserved. This would also be the case in figure 8.3 where the plastic bag has more pixels than the carrot. The examination of the cluster centres showed that the cluster of the item has distinctive colours to the background, this is shown in figure 4.13. To obtain these plots the segmented clusters were labelled manually. With this labelled data a binary support vector machine (SVM) classifier was built using the Scikit-learn package [19]. From the clusters, which overlap with the segmentation the SVM, the classifier selects the cluster which belongs to the fruit or vegetable.

Further examples of the colour clustering stages can be found in appendix section 8.1.


(a) The original image blurred by Gaussian filter.


(b) Segmented image by background subtraction, morphology and specularities removal.


(c) The 4 clusters mapped to the original image. The colours are the mean colours of each cluster.


(d) The overlap between the segmented region and the clustered image.

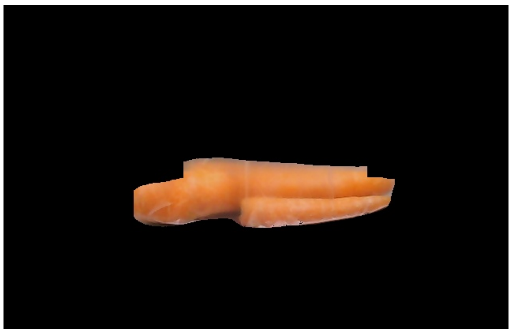
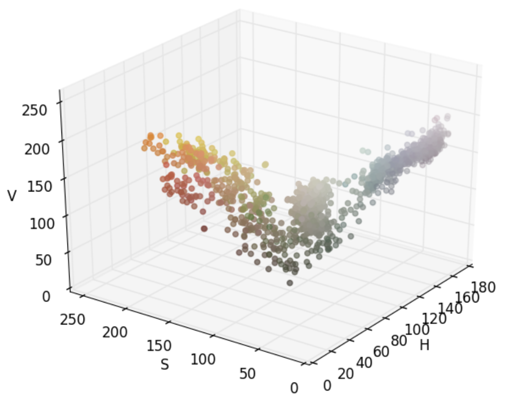
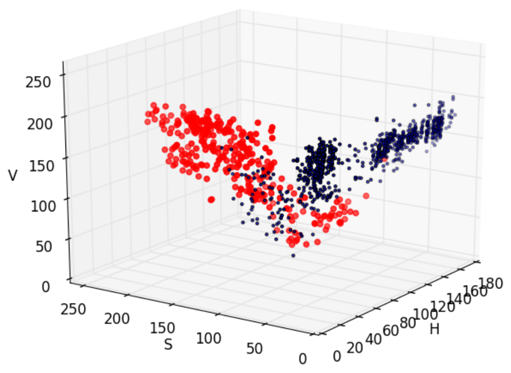

(e) The selected cluster of a carrot mapped back to the original image (before blurring).

Figure 4.12: The 5 steps of segmentation by colour clustering.



(a) The colour of the data point is the mean colour of the cluster.



(b) The red data points represent cluster centres of the items and the blue points represent the cluster centres of the backgrounds.

Figure 4.13: The cluster centres of the items and backgrounds shown in the HSV colour space. Both plots show the same data but with different colouring.

Chapter 5

Features

The segmentation process from chapter 4 identifies relevant pixels belonging to a fruit or vegetable item. Feature extraction calculates characteristic features of the relevant pixels. Features are a small set of values which should contain sufficient information to distinguish different classes of fruit and vegetables. The features are mainly based on colour (section 5.1), but shape (section 5.2) and texture (section 5.3) are considered as well.

5.1 Colour

Colour is one of the most discriminative characteristics of fruit and vegetable. Various kinds of fruit and vegetable take on colours from the whole visible colour spectrum, while at the same time, for a particular kind of fruit or vegetable the colour is relatively consistent given the same maturity level. High variability across different classes and small variability in each class makes colour an ideal feature.

The usual representation of the colour of every pixel in computer is by 3 unsigned 8-bits integers, which represent intensities of the three primary colours red (R), green (G), and blue (B). These are also called channels of the RGB colour space, which is further described in section 5.1.1. Colour can be represented in different colour spaces such as in normalised RGB colour space described in section 5.1.2 and hue, saturation, value (HSV) colour space described in section 5.1.3. Colour spaces vary in number of channels, ranges of these channels and have different advantages and disadvantages. There exist transformation functions from one colour space into another one.

To summarise the colour of the segmented pixels into features, properties of the colour histogram such as mean and variance are considered [31]. For each channel an arithmetic mean μ (equation 5.1) and its standard deviation σ (equation 5.2)of all the segmented pixels is calculated. The X in the equations is the set of all pixel values of one channel and N is the number of segmented pixels. For a three channel colour space, such as RGB, 6 features would be calculated - 2 for each channel. The three means of the channels represent the mean colour of the item. In general this might not be a good feature because averaging over several different colours says little about their original components, but as fruit and vegetable have relatively consistent colour of their whole surfaces this is a reasonable assumption to make. At the same time the standard deviations characterise how much the pixel values vary, so part of the information lost by averaging is preserved here. The different variations of the pixel values have a relation to the texture of the item.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{5.1}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{5.2}$$

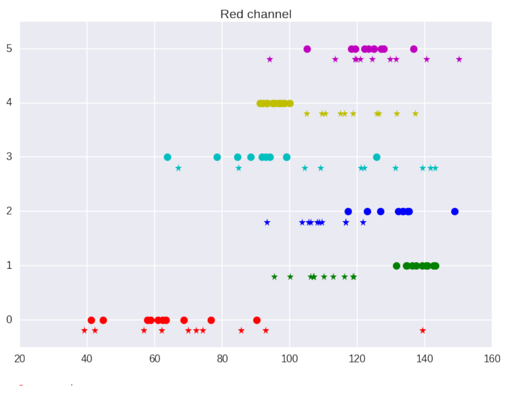
5.1.1 RGB

The RGB colour space is a linear colour space which uses three single wavelength primary colours - 645.16 nm for red (R), 526.32 nm for green (G), and 444.44 nm for blue (B) [11]. The RGB colour space is an additive colour space, in which the three primary colours are combined in various ways to form all the other colours. In theory, the available colours are represented as a unit cube, whose edges represent the R,G, and B intensities. In computers this range is discretised usually into range 0 to 255 inclusive (8 bits colours).

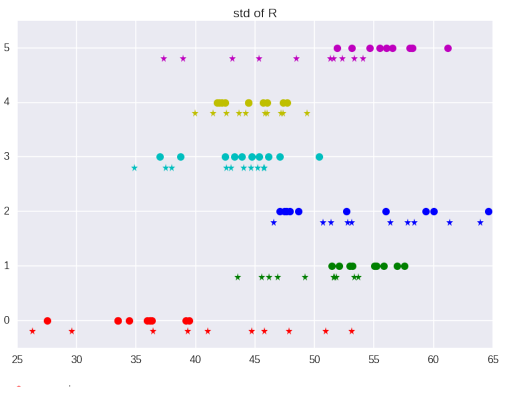
The mean and standard deviation of RGB values of the six fruits and vegetables from the first iteration are shown in figure 5.1. If only filled circles would be considered then it would be possible to draw vertical lines on the plots, so that the features of one class would be separated from the features of the other classes. This is a good sign, because it means that the features have the expressive power to discriminate between the classes. The plots serve as an evaluation method of the segmentation quality and it can be observed that the values are often shifted, much more spread, and drawing the lines would be much more difficult.

5.1.2 Normalised RGB

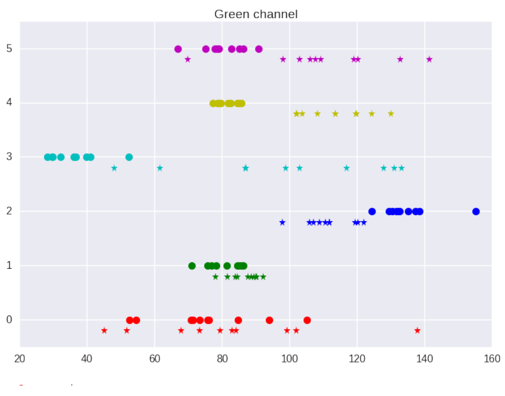
Normalised RGB colour space is a chromaticity space which removes all the information about the light intensity. For example, a bright green colour and dark green colour have the same representation in normalised RGB space. This is a very desirable property for the object classification task because it makes the colours illumination invariant. Normalised RGB is represented only by two channels r and g which are calculated from standard RGB representation by dividing



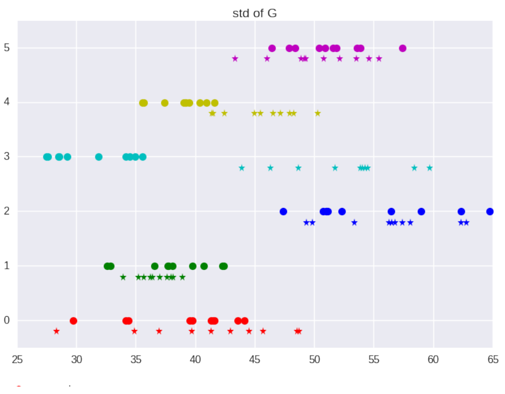
(a) Red channel



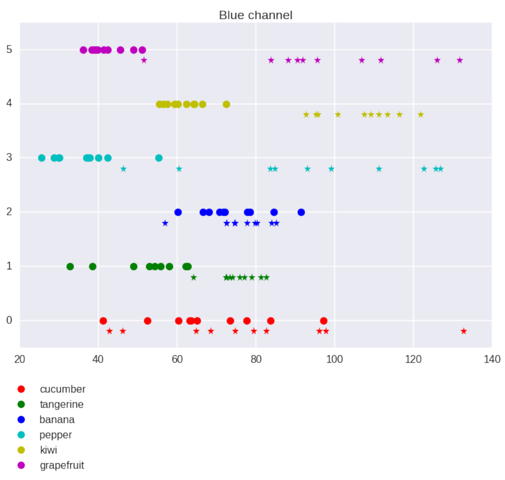
(b) Standard deviation of Red channel



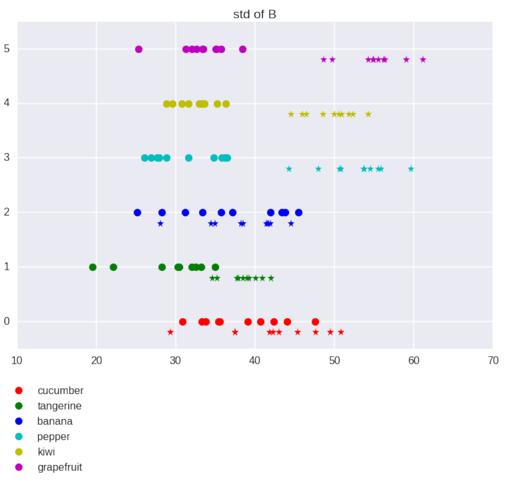
(c) Green channel



(d) Standard deviation of Green channel



(e) Blue channel



(f) Standard deviation of Blue channel

Figure 5.1: The distributions of RGB features and their standard deviation for 6 classes of un-bagged items from the iteration 1. The filled circles show the samples segmented using the ground truth masks and the stars always a little bit under them are from the segmentation calculated by the techniques from chapter 4. The six classes are cucumber, tangerine, banana, pepper, kiwi, and grapefruit, which have in respective order numbers zero to five and colours red, green, blue, cyan, yellow, and magenta.

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B} \qquad b = \frac{B}{R+G+B}$$

Figure 5.2: Equations for calculating the normalised RGB values from standard RGB colour space. Only the r and g are used as the channels of normalised RGB.

the red and green channel by the sum of all three RGB channels, see the equations in figure 5.2. The sum of the r, g, and b channels of normalised RGB is always equal to one, thus keeping only r and g preserves the whole information.

The mean and standard deviation of normalised RG values of the fruits and vegetables from the first iteration are shown in figure 5.3. In these plots, the drawing of the vertical decision line would be very straightforward for the filled circles, which used the ground truth segmentation. Again, it is a bit worse for the actual segmentation, but the spread of the values is much smaller, which can be also seen on the small range of the standard deviation plots.

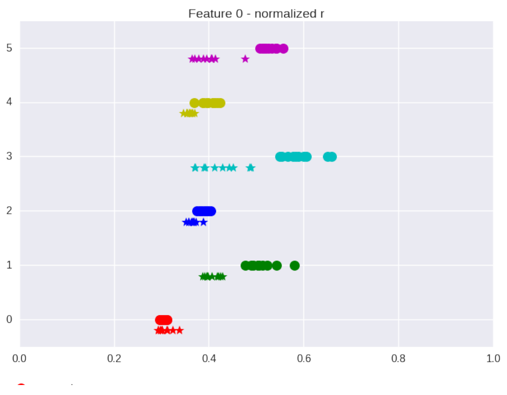
5.1.3 HSV

Hue, saturation, and value (HSV) colour space is a non-linear colour space. Hue depicts the property of a colour that varies in passing from red to green. Saturation depicts the property of a colour that varies in passing from red to pink. Value, also referred to as brightness or lightness,is the property that varies in passing from white to black. The main advantage of the HSV colour space is that the HSV coordinates captures the human intuition about the colour similarity better than other colour spaces, such as RGB.

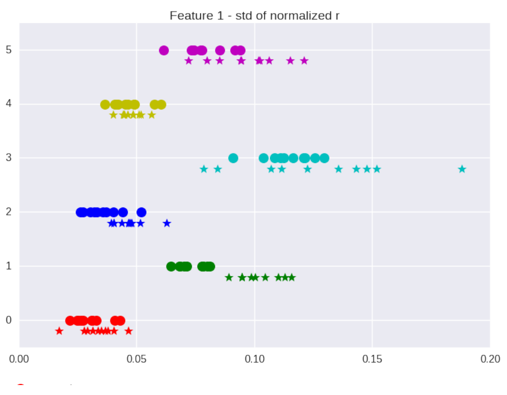
In the OpenCV package, hue is represented on range 0 to 179, saturation in range 0 to 255, and value on range 0 to 255 as well. In the literature hue is often represented as an angle on range 0 to 360. Knowing the ranges is important when comparing the values with a different source which might be using a different range.

5.1.4 Prior weight over the pixels

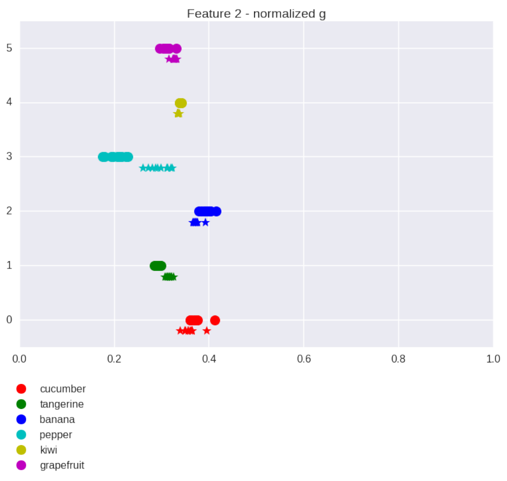
Even thought sophisticated techniques are used to segment the relevant pixels belonging to an item, they are not perfect and often some background pixels or plastic bag pixels are incorrectly preserved. An aim of the prior weight over the pixels is to reduce the bad influence of the background pixels. All segmented pixels are taken into account, but their influence on the resulting feature values of the mean and standard deviation is weighted according to *a priori* knowledge. Given a prior knowledge, a weight, a number between zero and one, is calculated for every pixel. This represents the significance for the pixel. Then, for the segmented pixels the weighted mean and weighted standard deviation is calculated according



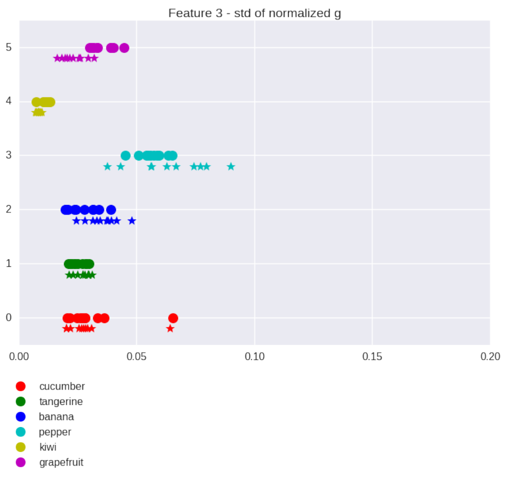
(a) Normalised Red channel



(b) Standard deviation of normalised Red channel



(c) Normalised Green channel



(d) Standard deviation of normalised Green channel

Figure 5.3: The distributions of normalised RGB features and their standard deviation for 6 classes of un-bagged items from the iteration 1. The filled circles show the samples segmented using the ground truth masks and the stars always a little bit under them are from the segmentation calculated by the techniques from chapter 4. The six classes are cucumber, tangerine, banana, pepper, kiwi, and grapefruit, which have in respective order numbers zero to five and colours red, green, blue, cyan, yellow, and magenta.

to the formulas shown in equations 5.3 and 5.4 respectively. In the weighted mean the value of every pixel is scaled (multiplied) by its weight and divided by the sum of all the weights of segmented pixels instead of dividing by the number of segmented pixels. Similarly for the weighted standard deviation the weighted mean is used and the sum of the weights is used instead of the number of pixels. If the weight of all pixels is one, then the weighted mean and standard deviation become normal arithmetic mean and standard deviation.

An assumption is made about the item's location in the image. Most of the items placed are around the centre of the tray and almost none are at the corners of the image (especially the top corners). Thus, if there are any pixels segmented close to the corners they are likely to be false positives. It would be very hard and inaccurate to define a hard boundary on locations from which the pixels should be considered and from which they should not be considered. The weights should be constructed so that they are high (close to one) around the centre of the tray and gradually decay to small values (around 0.1) close to the edges. This can be achieved by a bivariate normal distribution with the mean centred in the centre of the tray and the mean vector and covariance matrix would be scaled so that the peak would be at one and at the image edges the value would be 0.1. Then the weight of a pixel with coordinates x and y is calculated from the density function shown in equation 5.5. For the images of the first iteration, which have resolution 2048×1536, the mean vector is $[1024, 768]$ and the covariance matrix is an identity matrix multiplied by 130000.

$$\mu_w = \frac{\sum_{i=1}^{N} x_i * w_i}{\sum_{i=1}^{N} w_i} \tag{5.3}$$

$$\sigma_w = \sqrt{\frac{\sum_{i=1}^{N} w_i (x_i - \mu_w)^2}{\sum_{i=1}^{N} w_i}} \tag{5.4}$$

$$w(x,y) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}} exp(-\frac{1}{2}(\begin{bmatrix} x \\ y \end{bmatrix} - \mu)^T \Sigma^{-1} (\begin{bmatrix} x \\ y \end{bmatrix} - \mu))) \tag{5.5}$$

5.2 Shape

The shape of a fruit or vegetable can be very characteristic for some items and less for others. There exists a large number of circular, ball-like, fruits and vegetables - apples, oranges, tomatoes, and peaches. Other items are more elongated like cucumber, bananas, and carrots. Some can have a wide range of shapes and sizes like potatoes. It would be difficult to distinguish between circular items based only on the shape feature, but it can divide the classes into smaller subcategories among which it would be easier to discriminate.

Two shape features are considered - aspect ratio and circularity defined by equations 5.6 and 5.7 respectively [22]. The proportion of the smallest and the largest

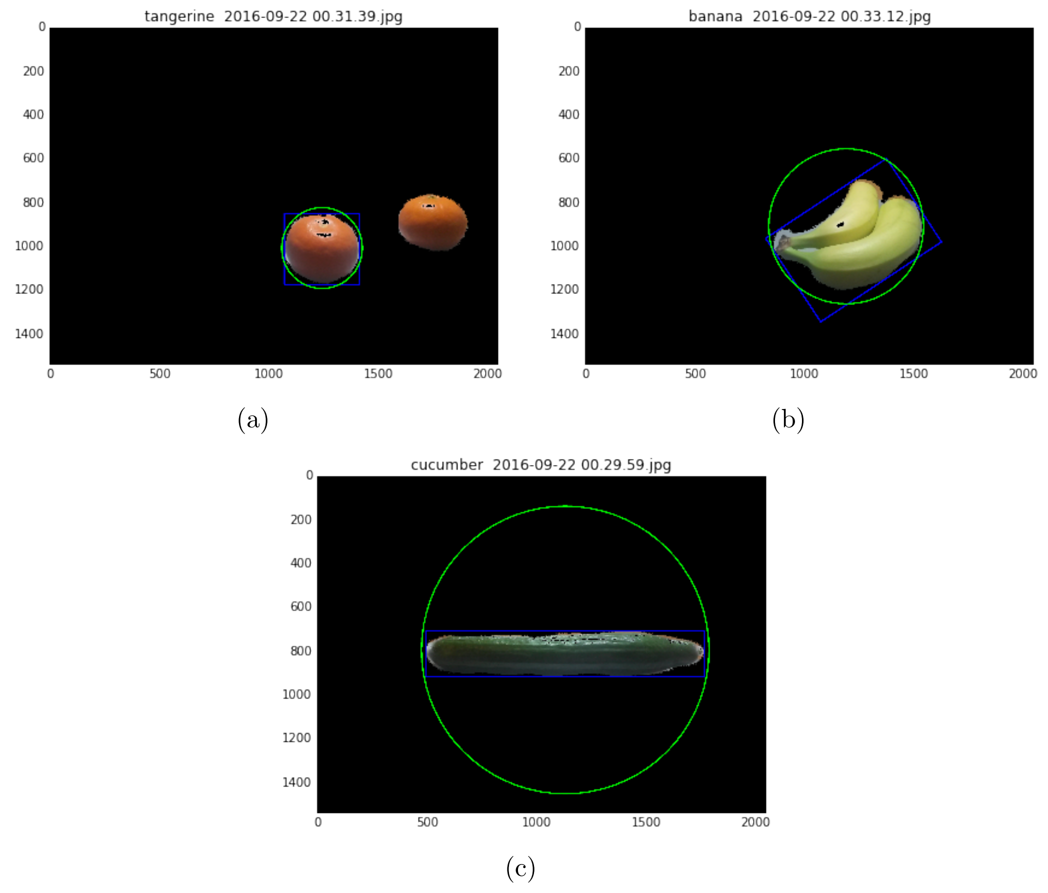

tangerine 2016-09-22 00.31.39.jpg
banana 2016-09-22 00.33.12.jpg
(a)
(b)
cucumber 2016-09-22 00.29.59.jpg
(c)

Figure 5.4: Examples of successful shape detection.

diameter in the aspect ratio characterise elongation of the shape. The circularity is a function of the perimeter P (in pixel units) and the area A (number of pixels belonging to the region). The circularity of a circle is 1 and is much lower when a region has any holes.

$$A_R = \frac{smallest\ diameter}{largest\ diameter} \tag{5.6}$$

$$circularity = \frac{4\pi A}{P^2} \tag{5.7}$$

From the practical point of view, when there is more than one region segmented, only the largest one is considered for shape features calculation. The biggest weakness of the shape features is that they work only on a single piece of fruit or vegetables. When there are more pieces, they are contained in one deformed region or they are in the plastic bag so that the shape features do not work.

Two examples of successfully identified shape features and unsuccessful ones are shown in figure 5.4.

5.3 Texture

Texture might be a useful discriminator between the classes as well. Some items are very smooth and with a consistent colour while others may contain stripes or small dots. There are many texture extraction techniques such as Grey level coocurrence matrix (GLCM), Energy filters and edgeness, Gabor filters, and Wavelet transform [21]. However, all of these share a common problem in the context of fruit and vegetable classification - the plastic bag smooths out all the texture properties. The only form of considering the texture is through the standard deviation of the colour histogram described in section 5.1.

5.4 Feature normalisation

Feature normalisation is a simple post-processing step of the feature extraction, which scales all the feature ranges to the range between -1 to 1. This is not necessary if all the features are on a comparable ranges and the classifier can deal with larger ranges. When different features are combined, like colour and shape, the normalisation is likely to be beneficial. A negative effect of not normalised features might be that size of one feature would dominate all the others no matter what values would they be taking.

Chapter 6

Classification

Classification is the final stage of the processing pipeline which produces the predicted class label of the item and/or ordered ranking of the classes according to their probabilities. The inputs to the classifier are the extracted features of the captured image and the model's parameters. The parameters are estimated from the labelled training images, which undergo the same feature extraction process as the tested image.

6.1 Gaussian Bayes Classifier

Gaussian Bayes classifier originates from the Bayes' theorem shown in figure 6.1, which states that the posterior probability of class c given its feature vector x can be calculated from the likelihood of x given class c times the prior probability of the class c over the probability of x. The class for which the posterior probability is the highest is chosen as the predicted class. While it is difficult to calculate the actual value of evidence $p(x)$, this value is not necessary for the comparison of $p(c|x)$ of the different classes, as these are all proportional to the same $p(x)$. Hence it can be left out as shown in equation 6.1. The prior probability of the class $p(c)$ can be the calculated proportion of the frequency of the class compared to the frequency of all classes. In the datasets from both iterations the classes are proportional, so the prior would be the same for all classes (one over the number of classes), hence not influencing the maximum. In practice, the prior would be calculated according to how frequently each class of fruit and vegetable is being bought. When the priors are equal, finding the class which maximises the likelihood, maximises the non-normalised posterior probability. This is called the decision rule and is shown in equation 6.2. The parameters $\boldsymbol{\mu}_c$, and Σ_c are estimates of the mean and covariance matrix of the class, which are calculated from the training data.

The likelihood $p(x|c)$ is modelled by multivariate normal, also called Gaussian, distribution (thus the name Gaussian Bayes Classifier). The probability density of the multivariate normal distribution is shown in equation 6.3. The n in the

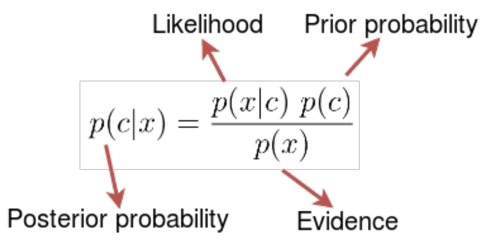


Figure 6.1: Bayes' theorem

equation is the dimension of the feature vector $\mathbf{x}$, $|\Sigma_c|$ is the determinant of the covariance matrix, and Σ_c^{-1} is the inverse of the covariance matrix. The mean vector $\boldsymbol{\mu}_c$, and the covariance matrix Σ_c are calculated from the labelled training features of each class. The mean vector has the dimension n - the number of features and it is calculated as per feature arithmetic mean of the training samples of the class c, see the equation 6.4 (the i index refers to the index of the sample $\mathbf{x}_c$, not the feature value of the feature vector $\mathbf{x}_c$). The covariance matrix has dimension $n \times n$ and each element of the matrix is calculated according to equation 6.5. The $\mathbf{f}_{c,(i)}$ is a vector containing i-th feature of all N samples of class c (the dimension is N, k is the index of an element in the $\mathbf{f}_{c,(i)}$ vector).

$$p(c|x) \approx p(x|c)p(c) \tag{6.1}$$

$$\hat{y} = \underset{c \in \{1,\dots,C\}}{argmax}\; p(\mathbf{x}|c, \boldsymbol{\mu}_c, \Sigma_c) \tag{6.2}$$

$$p(\mathbf{x}|c, \boldsymbol{\mu}_c, \Sigma_c) = \frac{1}{(2\pi)^{(n/2)}\sqrt{|\Sigma_c|}} exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \Sigma_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c)) \tag{6.3}$$

$$\boldsymbol{\mu}_c = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{c,(i)} \tag{6.4}$$

$$\begin{aligned} \Sigma_{c,(i,j)} &= cov(\mathbf{f}_{c,(i)}, \mathbf{f}_{c,(j)}) \\ &= \frac{1}{N} \sum_{k=1}^{N} (\mathbf{f}_{c,(i,k)} - \boldsymbol{\mu}_{c,(i)})(\mathbf{f}_{c,(j,k)} - \boldsymbol{\mu}_{c,(j)}) \end{aligned} \tag{6.5}$$

In summary, Gaussian Bayes classifier, in the training stage, estimates the mean vector and a covariance matrix for each class (equations 6.4 and 6.5). In the prediction stage, the likelihood that the feature vector of the test image belongs to a class is calculated for all the classes (equation 6.3). Then the class with the highest likelihood is selected as class label (equation 6.2). The likelihoods of all classes can be used to construct an ordered ranking of all classes from the highest likelihood to the lowest. In practice the log likelihood is used to avoid the underflow when a lot of small probabilities are multiplied.

6.2 Evaluation

6.2.1 Accuracy and Rank

The fruit and vegetable classification is a supervised learning task, as the labelled dataset of fruit and vegetable images is available (section 3). Hence the accuracy is the most straightforward evaluation metric, which compares the predicted label with the ground truth label and calculates the proportion of correctly classified images. Usually, it is expressed as percentage, multiplied by 100% as shown in equation 6.6.

$$Accuracy(test\ images) = \frac{number\ of\ correctly\ classified\ images}{number\ of\ test\ images} * 100\% \quad (6.6)$$

The second evaluation measure is the average rank of the correct label. The rank is the position (index) of the correct label in the sorted list of prediction from the highest to the lowest likelihood. The perfect performance would have the average rank zero, as the correct label would always appear at the zeroth index of the ordered ranking by the likelihood. Higher average rank indicates lower performance.

The third evaluation measure, related to the average rank, is the proportion of correct labels, which appear in the top three most likely classes. It can also be represented as percentage when multiplied by 100%. The top three measure is tailored to the application of fruit and vegetable recognition at the self-checkout. Offering the user just top three choices on the main screen, which contains the correct item is a huge improvement, compared to the navigation through subcategories of tens of items, and can be considered as success.

All three measures can be calculated either for the whole test set, or a subset of the test set. For example, these measures can be calculated per class. This can reveal a source of error or identify easy and difficult classes.

6.2.1.1 Confusion matrix

The confusion matrix as described in [25] is a table, which visualises the performance. On the horizontal axis (columns of the table) are the predicted classes and on the vertical axis (rows of the table) are the actual classes. On the diagonal of the confusion matrix are the correct predictions (the predicted class is equal to the actual class). The exemplar layout of the confusion matrix is shown in table 6.1. Here 3 bananas and 4 kiwis were correctly classified, 2 actual bananas were classified as kiwi and 1 kiwi as banana.

The confusion matrices are useful for evaluating how are performing different classes and which pairs or subsets of classes are often being mismatched.

		Predicted class		
		banana	kiwi	...
Actual class	banana	3	2	
	kiwi	1	4	
	...			

Table 6.1: An example of the layout of the confusion matrix.

6.2.2 Leave-one-out cross-validation

Leave-one-out cross-validation is an effective evaluation technique of a classifier when the dataset is relatively small. The usual train, test, and validation split works well when sufficient amount of data is available to train a model, fine tune it on the validation set and then obtain reliable metrics on the test set. Leave-one-out cross-validation is an alternative. Each sample of the dataset is used once for testing and all other times as a training sample. One sample at the time is taken from the dataset, the model is trained on the rest of the samples (parameters' estimation) and then the model is used to predict the one sample which was not used in the training.

This method is used throughout the experiments described in section 6.3. Even the combined dataset of bagged and un-bagged items, which have altogether 700 images is not particularly large and thanks to the leave-one-out cross-validation more insightful information can be obtained.

6.3 Experiments

A series of experiment were conducted to evaluate performance of the classifier. The summary of the experiments using the dataset from the second iteration is shown in table 6.2. For all of these experiments were used the full pipeline of consisting of background subtraction, specularities removal, morphology, colour clustering, feature extraction based on the colour space, and the leave-one-out cross-validation for all the images of the dataset. For each experiment is calculated overall accuracy, average rank of correct label and how often the correct label appear in the top three predictions.

The HSV features have the worst performance. The performance of RGB and normalised RG colour spaces are very comparable and in general high.

The extended confusion matrices of the whole dataset for RGB, normalised RG, and HSV colour spaces are show in tables 6.3, 6.4, and 6.5 respectively. A phenomenon, which can observed consistently across the confusion matrices is that items with similar colour are frequently confused. The banana is frequently confused with lemon and melon with cucumber. However, this affect the top three prediction accuracy very little.

Features	**N img**	**(un)-bagged**	**Acc (%)**	**Avg pos**	**Top three (%)**
RGB	350	bagged	79.43	0.36	97.71
Norm RG	350	bagged	77.71	0.35	98.86
HSV	350	bagged	63.14	0.76	94.57
RGB	350	un-bagged	96.57	0.04	100.00
Norm RG	350	un-bagged	95.71	0.05	100.00
HSV	350	un-bagged	90.28	0.16	99.43
RGB	700	both	85.71	0.25	98.57
Norm RG	700	both	82.86	0.24	99.29
HSV	700	both	74.29	0.49	97.00

Table 6.2: Summary of the experiments performed using the dataset from the second iteration. The columns in the respective order are the features used, number of images, whether the items were bagged or un-bagged, the over all accuracy, the average position of the correct label, and the percentage of the labels in top three predictions.

Table 6.3: The whole dataset of un-bagged and bagged items using RGB colour space.

	cucumber	tangerine	banana	pepper	kiwi	grapefruit	onion	lemon	tomato	potato	grapes	carrot	melon	garlic	Accuracy	Average position	Top three
cucumber	40	0	0	0	0	0	0	0	0	0	0	0	3	7	80.0	0.32	100.0
tangerine	0	48	0	0	0	0	0	0	0	0	0	2	0	0	96.0	0.36	96.0
banana	0	0	34	0	1	1	0	8	0	1	4	0	0	1	68.0	0.82	92.0
pepper	0	0	0	42	0	0	0	0	8	0	0	0	0	0	84.0	0.54	98.0
kiwi	0	0	0	1	46	0	0	0	1	0	0	0	0	2	92.0	0.28	98.0
grapefruit	0	2	2	0	0	43	1	1	0	1	0	0	0	0	86.0	0.7	98.0
onion	0	0	0	0	5	0	42	0	0	1	0	0	0	2	84.0	0.64	98.0
lemon	0	0	4	0	0	2	0	43	0	1	0	0	0	0	86.0	0.48	96.0
tomato	0	0	0	4	0	0	0	0	46	0	0	0	0	0	92.0	0.48	98.0
potato	0	0	2	0	0	2	1	0	0	41	0	0	1	3	82.0	0.58	94.0
grapes	0	0	2	0	0	0	0	0	0	0	45	0	0	3	90.0	0.74	98.0
carrot	0	1	0	0	0	0	1	0	0	0	0	48	0	0	96.0	0.4	96.0
melon	8	0	0	0	0	0	0	0	0	0	1	0	41	0	82.0	0.26	98.0
garlic	5	0	0	0	0	0	1	0	0	1	2	0	0	41	82.0	0.24	98.0
															85.71	0.49	97.00

Table 6.4: The whole dataset of un-bagged and bagged items using normalised RG colour space.

	cucumber	tangerine	banana	pepper	kiwi	grapefruit	onion	lemon	tomato	potato	grapes	carrot	melon	garlic	Accuracy	Average position	Top three
cucumber	37	0	0	0	0	0	0	0	0	0	1	0	6	6	74.0	0.32	100.0
tangerine	0	41	0	0	0	0	2	0	0	0	0	7	0	0	82.0	0.36	96.0
banana	0	0	36	0	0	1	0	8	0	0	5	0	0	0	72.0	0.82	92.0
pepper	0	0	0	47	0	0	0	0	3	0	0	0	0	0	94.0	0.54	98.0
kiwi	1	0	0	0	37	0	8	0	0	3	0	0	0	1	74.0	0.28	98.0
grapefruit	0	2	3	0	0	41	1	1	0	2	0	0	0	0	82.0	0.7	98.0
onion	0	2	0	0	11	0	35	0	0	1	0	0	0	1	70.0	0.64	98.0
lemon	0	0	9	0	0	1	0	40	0	0	0	0	0	0	80.0	0.48	96.0
tomato	0	0	0	2	0	0	0	0	48	0	0	0	0	0	96.0	0.48	98.0
potato	0	0	2	0	2	0	0	0	0	44	0	0	0	2	88.0	0.58	94.0
grapes	3	0	1	0	0	0	0	0	0	0	40	0	5	1	80.0	0.74	98.0
carrot	0	3	0	0	0	0	1	0	0	0	0	46	0	0	92.0	0.4	96.0
melon	3	0	1	0	0	0	0	0	0	0	3	0	43	0	86.0	0.26	98.0
garlic	2	0	0	0	1	0	0	0	0	1	1	0	0	45	90.0	0.24	98.0
															82.86	0.49	97.00

Table 6.5: The whole dataset of un-bagged and bagged items using HSV colour space.

	cucumber	tangerine	banana	pepper	kiwi	grapefruit	onion	lemon	tomato	potato	grapes	carrot	melon	garlic	Accuracy	Average position	Top three
cucumber	38	0	0	0	0	0	0	0	0	0	1	0	3	8	76.0	0.32	100.0
tangerine	0	43	0	2	0	0	0	1	1	0	0	3	0	0	86.0	0.36	96.0
banana	0	0	33	1	2	3	1	5	0	3	2	0	0	0	66.0	0.82	92.0
pepper	0	0	1	30	1	0	1	0	17	0	0	0	0	0	60.0	0.54	98.0
kiwi	0	0	0	1	43	0	0	0	0	0	3	0	2	1	86.0	0.28	98.0
grapefruit	0	1	5	2	0	30	0	6	0	1	0	5	0	0	60.0	0.7	98.0
onion	0	0	0	2	4	0	36	0	1	4	2	0	0	1	72.0	0.64	98.0
lemon	0	0	2	0	0	7	0	40	1	0	0	0	0	0	80.0	0.48	96.0
tomato	0	0	0	16	2	0	1	0	31	0	0	0	0	0	62.0	0.48	98.0
potato	0	0	3	0	1	0	3	0	1	37	1	1	1	2	74.0	0.58	94.0
grapes	1	0	10	0	4	0	0	1	0	2	28	0	1	3	56.0	0.74	98.0
carrot	0	3	0	0	0	4	0	0	0	0	1	42	0	0	84.0	0.4	96.0
melon	4	0	1	0	2	0	0	0	0	0	1	0	42	0	84.0	0.26	98.0
garlic	2	0	0	0	0	0	0	0	0	1	0	0	0	47	94.0	0.24	98.0
															74.29	0.49	97.00

Chapter 7

Discussion

7.1 Future work

The proposed classification pipeline satisfies all the essential requirements for the live classification of the fresh fruits and vegetables in the self-checkout environment. The main future challenges are to increase robustness of the classifier and to improve the classification of the bagged items.

Currently, the methods rely on the static background, which can be usually satisfied, but there can be situations in which an unexpected object appears in the field of view. This can be a hand of a shopper, or the next product, which is prepared to be scanned next. This would be confusing for the current classifier mainly in the colour clustering of the segmentation stage.

As there is a very limited pool of features which can be used for the bagged items classification an additional feature might be beneficial. One such feature could be the weight of the items from the scale. This can be compared to the estimated size of the items and could help distinguish lighter items such as garlic which, thanks to its colour, is a challenging item, from heavier ones.

The current classifier is very good in predicting the correct label in the top three. One reason for this is that there are no groups greater than tree classes which are expected to have similar colour. If only the first prediction is considered, bananas and lemons get confused easily. This might be addressed by the item's specific features. For the un-bagged items texture can improve classification significantly and for the bagged items it remains an open question.

The deep convolutional neural networks are performing very well in the computer vision classification tasks when a large amount of training data is available. This data should ideally be collected in the supermarket over a period of time. An alternative to training the full convolutional network from scratch is to use a pre-trained neural network for the object recognition. The hidden layer of the pre-trained network can be used as feature extractors and only the last layer would be trained to fine tune the fruit and vegetable recognition. This approach is called transfer learning.

7.2 Summary

Fruit and vegetable classification pipeline from the images was developed in this report. The task is motivated by the practical application at the self-checkout terminal in supermarkets to ease buying the fresh produce sold by weight. The pipeline consist of three main stages, namely segmentation, feature extraction, and classification. Each stage was addressed and optimised in order to obtain the best overall classification performance. The major challenge presented is the classification of the fresh produce items placed into the plastic bag. The usually effective segmentation technique background subtraction is not sufficient to separate the items and the background overlaid by the see-through plastic bag. This challenge was addressed by segmentation thought colour clustering, combining it with usual segmentation techniques, and the supervised support vector machine classifier to distinguish whether a cluster contains background or foreground pixels. The plastic bag reduces or completely removes characteristic features of the fruits and vegetables, mainly the texture. As it is not possible to distinguish separate items placed into the plastic bag the shape feature, which might have discriminate between oval and elongated items is not usable. The features are based on the colour of the segmented pixels. Three colour spaces were compared.

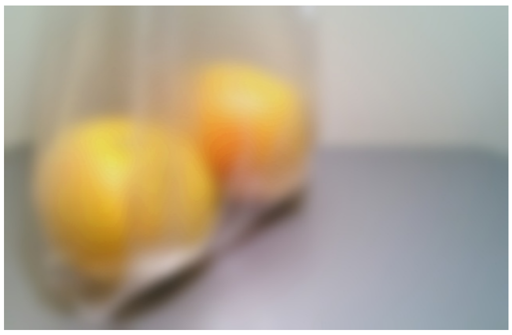
The final performance of the Gaussian Bayes classifier is reasonably high - for the mixed dataset of un-bagged and bagged items with 14 classes the over all accuracy is at 85% considering only the top prediction and order of 98% when first three predicted labels are considered. This can be considered sufficient for the self-checkout application.

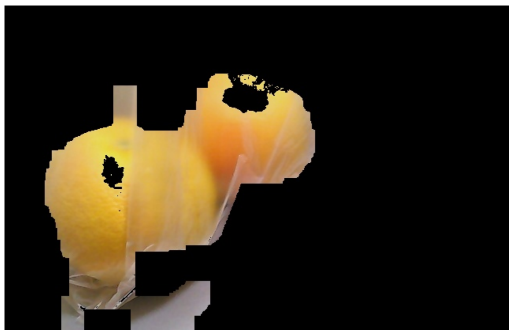
Even though the position, the number of the items presence of the plastic bag was varied across the images, the conditions of the dataset creation were relatively controlled in terms of light and the background. Varying these conditions in the real environment may decrease the performance of the classifier (or some stage of the processing pipeline). The normalised RG colour space should be invariant to changes in lightness, and thus perform well even under different light condition. Nonetheless, all images in the second dataset contain a reference colour sticker if an additional colour correction would be required.

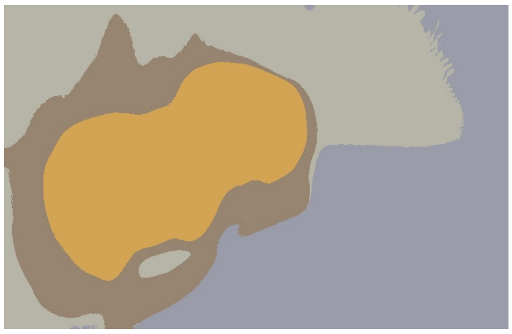
Chapter 8

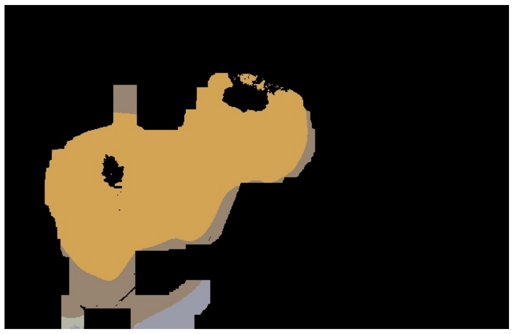
Appendix

8.1 Colour clustering examples


(a) The original image blurred by Gaussian filter.


(b) Segmented image by background subtraction, morphology and specularities removal.


(c) The 4 clusters mapped to the original image. The colours are the mean colours of each cluster.


(d) The overlap between the segmented region and the clustered image.

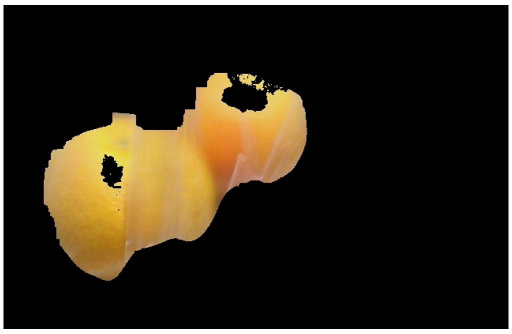
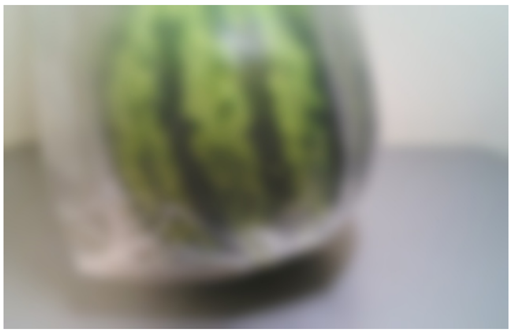
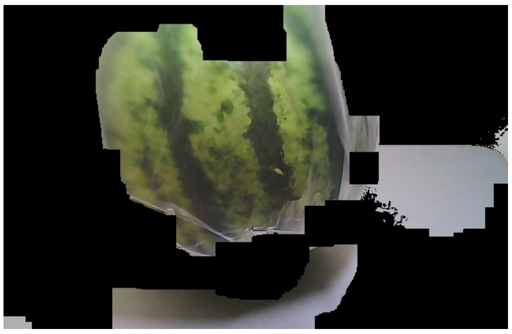
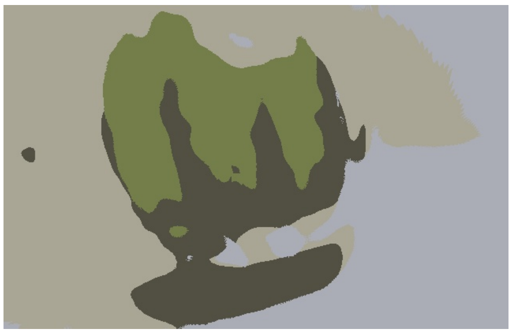
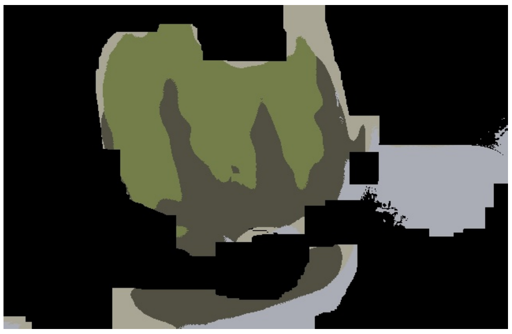

(e) The selected cluster of a carrot mapped back to the original image (before blurring).

Figure 8.1: The 5 steps of segmentation by colour clustering - grapefruit.


(a) The original image blurred by Gaussian filter.


(b) Segmented image by background subtraction, morphology and specularities removal.


(c) The 4 clusters mapped to the original image. The colours are the mean colours of each cluster.


(d) The overlap between the segmented region and the clustered image.

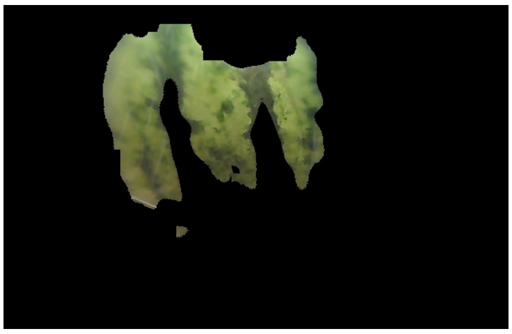
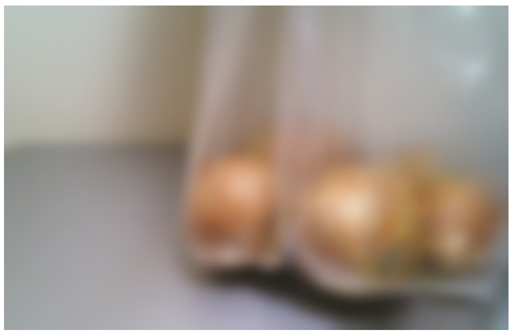
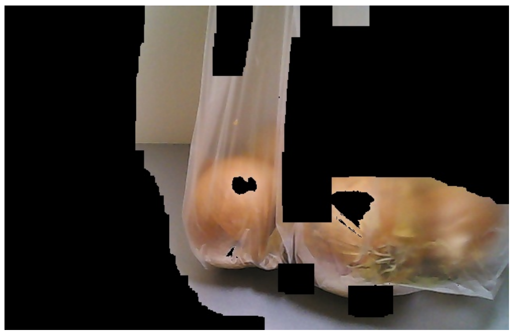
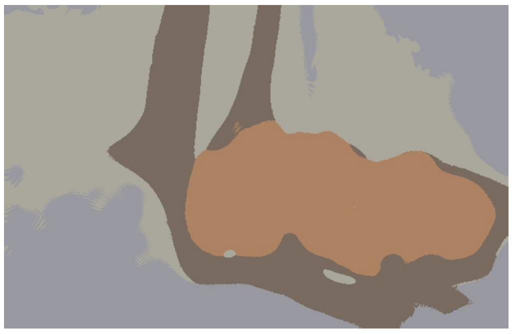
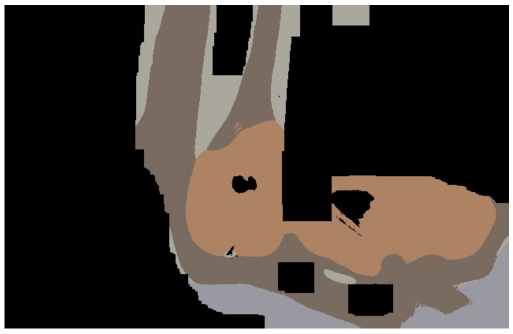

(e) The selected cluster of a carrot mapped back to the original image (before blurring).

Figure 8.2: The 5 steps of segmentation by colour clustering - melon.


(a) The original image blurred by Gaussian filter.


(b) Segmented image by background subtraction, morphology and specularities removal.


(c) The 4 clusters mapped to the original image. The colours are the mean colours of each cluster.


(d) The overlap between the segmented region and the clustered image.

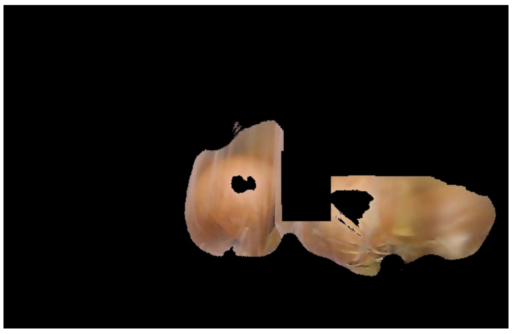

(e) The selected cluster of a carrot mapped back to the original image (before blurring).

Figure 8.3: The 5 steps of segmentation by colour clustering - onion.

Bibliography

[1] Adrian Rosebrock. Displaying a video feed with opencv and tkinter. http://www.pyimagesearch.com/2016/05/30/displaying-a-video-feed-with-opencv-and-tkinter/. Accessed: 2017-01-20.

[2] S Arivazhagan, R Newlin Shebiah, S Selva Nidhyanandhan, and L Ganesan. Fruit recognition using color and texture features. *Journal of Emerging Trends in Computing and Information Sciences*, 1(2):90–94, 2010.

[3] Claire Carter. Shoppers steal billions through self service tills. http://www.telegraph.co.uk/finance/personalfinance/household-bills/10603984/Shoppers-steal-billions-through-self-service-tills.html, January 2014. Accessed: 2017-03-03.

[4] Cash and Credit Registers. Cash and credit registers. http://americanhistory.si.edu/collections/object-groups/cash-and-credit-registers. Accessed: 2017-01-24.

[5] Lukas Danev. MInf Part 1: Exploratory analysis of the brains resting state functional connectivity in fMRI data. Master's thesis, The University of Edinburgh, 2016.

[6] Bjørn S Dissing, Line H Clemmesen, Hanne Løje, Bjarne K Ersbøll, and Jens Adler-Nissen. Temporal reflectance changes in vegetables. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1917–1922. IEEE, 2009.

[7] Shiv Ram Dubey and Anand Singh Jalal. Fruit and vegetable recognition by fusing colour and texture features of the image using machine learning. *International Journal of Applied Pattern Recognition*, 2(2):160–181, 2015.

[8] Shiv Ram Dubey and AS Jalal. Robust approach for fruit and vegetable classification. *Procedia Engineering*, 38:3449–3453, 2012.

[9] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[10] R.B Fisher, K. Dawson-Howe, A. Fizgibbon, C.Robertson, and E. Trucco. *Dictionary of Computer Vision and Image Processing*. Wiley, 2005.

[11] David A Forsyth and Jean Ponce. *Computer vision: a modern approach.* Pearson Education, 2012. chapter 3.

[12] A Gopal, R Subhasree, Venkatesh K Srinivasan, NK Varsha, and Sumathi Poobal. Classification of color objects like fruits using probability density function (pdf). In *Machine Vision and Image Processing (MVIP), 2012 International Conference on*, pages 1–4. IEEE, 2012.

[13] D.R. Humble. Self-checkout of produce items, October 16 1990. US Patent 4,964,053.

[14] D.R. Humble. System for self-checkout of bulk produce items, June 20 1995. US Patent 5,426,282.

[15] Antonio Ramón Jiménez, Anil K Jain, R Ceres, and JL Pons. Automatic fruit recognition: a survey and new results using range/attenuation images. *Pattern recognition*, 32(10):1719–1736, 1999.

[16] N.P. Johnson. Automated self-service checkout system, November 29 1988. US Patent 4,787,467.

[17] Hu-Lin Kuang, Leanne Lai Hang Chan, and Hong Yan. Multi-class fruit detection based on multiple color channels. In *2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pages 1–7. IEEE, 2015.

[18] Fernando López-García, Gabriela Andreu-García, José Blasco, Nuria Aleixos, and José-Miguel Valiente. Automatic detection of skin defects in citrus fruits using a multivariate image analysis approach. *Computers and Electronics in Agriculture*, 71(2):189–197, 2010.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Anderson Rocha, Daniel C Hauagge, Jacques Wainer, and Siome Goldenstein. Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture*, 70(1):96–104, 2010.

[21] LA Ruiz, A Fdez-Sarría, and JA Recio. Texture feature extraction for classification of remote sensing data using wavelet decomposition: a comparative study. In *20th ISPRS Congress*, volume 35, pages 1109–1114, 2004.

[22] Woo Chaw Seng and Seyed Hadi Mirisaee. A new method for fruits recognition system. In *Electrical Engineering and Informatics, 2009. ICEEI'09. International Conference on*, volume 1, pages 130–134. IEEE, 2009.

[23] Pierre Soille. *Morphological image analysis: principles and applications.* Springer Science & Business Media, 2013.

[24] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. CL Engineering, 1998.

[25] Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77 – 89, 1997.

[26] Da-Wen Sun. *Computer vision technology for food quality evaluation*. Academic Press, 2016. chapters 11-15.

[27] NCR Corporation Whitepaper. Self-checkout: a global consumer perspective. `https://www.ncr.com/sites/default/files/white\_papers/RET\_SCO\_wp.pdf`, 2014. Accessed: 2017-01-24.

[28] Wikipedia. Barcode. `https://en.wikipedia.org/wiki/Barcode`. Accessed: 2017-01-24.

[29] Wikipedia. Reflectance. `https://en.wikipedia.org/wiki/Reflectance`. Accessed: 2017-03-23.

[30] Wikipedia. Specularity. `https://en.wikipedia.org/wiki/Specularity`. Accessed: 2017-03-23.

[31] Yudong Zhang and Lenan Wu. Classification of fruits using computer vision and a multiclass support vector machine. *Sensors*, 12(9):12489–12505, 2012.