

THE UNIVERSITY OF HULL

A Food Monitoring System using  
Image Segmentation and Machine  
Learning

being a Thesis submitted for the Degree of Doctor of Philosophy  
in Medical Engineering In the University of Hull

BY

MAHMUD I ALATAWI

## **Acknowledgement**

First and foremost, I would like to thank my mother, Fatimah Alatawi, for her support and continued faith throughout my PhD. I will never be able to express my gratitude for her support and inspiration.

I am deeply grateful to my supervisor, Dr Kevin S. Paulson, who gave me the golden opportunity to finish this wonderful project. I would also like to thank my second supervisor, Dr Amadou Gning, for his useful advices.

I would like to thank Prof Michael Fagan for his support and assistance.

Finally, Alhamdulillah, all praises and thanks to Allah for giving me the strength and courage to complete my thesis.

## **Abstract**

In recent years, the number of people requiring dietary monitoring has risen, and dietary records are an essential part of the diagnosis and treatment of many health problems. This project designs and evaluates a food monitoring system for use in institutions such as hospitals. It includes hardware for image capture and data processing, as well as software components such as a food recognition system, a food and nutrition database, and an interface to allow users to interrogate and interact with the system. Its objectives are to specify, acquire and evaluate an image capture system suitable for an institution such as a hospital; to develop and evaluate a food recognition system compatible with the image capture system; and develop and evaluate a system to estimate the amount of food eaten by comparing images before and after eating.

A novel method is developed based on image segmentation and a machine learning algorithm. The first stage is to remove any non-food object from the image: the colour technique was tested using almost 3000 images, and yielded near 98% accuracy. The second stage used the K-means++ clustering algorithm to group parts of the image into coherent regions, each assumed to be a food type; the average accuracy for all types of food was 94%. The third stage identified the foods within each segment of the image, through machine learning; the best accuracy for food classification was 98.7%. This algorithm was able to estimate food eaten with 86% accuracy. Tests indicate that this automated system could replace the paper-and-pen approach used in Hull and East Yorkshire Hospitals, and yield similar or better nutritional metrics.

## **Publications**

The following conference articles have been produced during the PhD research:

1-The project (A Food Recognition System for Dietary Monitoring) presented in the 6th International Conference in Health Care and Life Science Research (ICHLSR). The conference has headed in London, United Kingdom during September 18-19, 2015.

## Table of contents

Acknowledgement .....	i
Abstract .....	ii
Publications .....	iii
Table of contents .....	iv
Dedication .....	viii
List of figures .....	ix
List of tables .....	xiii
Chapter 1 Introduction.....	14
1.1 Project Background .....	14
1.2 Problem Definition and Justification.....	17
1.3 Aims and Objectives.....	18
1.4 Thesis Contribution .....	19
1.5 Thesis Outline.....	20
Chapter 2 Literature Review .....	21
2.1 Introduction .....	21
2.2 Paper Recording Methods .....	22
2.3 On-Body Sensing: .....	22
2.4 Microwave Food Identification .....	25
2.5 Image processing and computer vision .....	25
2.5.1 First stage: Extract Image Descriptors .....	33
Step one: Key points extraction .....	33
Step two: Key point description.....	35
Step Three: Clustering and creating a visual word dictionary. ....	36
2.5.2 Second Stage: Training the Classifier .....	36
2.6 Research Gaps.....	39
Chapter 3 Background on Image Processing and Machine Learning .....	41
3.1 Introduction .....	41
3.2 Digital Images.....	41
3.3 Colour Spaces.....	42
3.3.1 Colour Model, Space, CIE 1931 .....	42

3.3.2	Adobe RGB, sRGB .....	43
3.3.3	CIELAB Colour Model .....	43
3.3.4	HSV colour model.....	44
3.4	Digital Image Segmentation .....	44
3.4.1	Similarity Segmentation .....	45
3.4.2	Discontinuity Segmentation.....	45
3.5	Thresholding.....	45
3.5.1	Threshold Different Objects.....	46
3.5.2	Global and Local Thresholding .....	48
3.5.3	Automatic Thresholding.....	49
3.6	Machine learning .....	49
3.6.1	A brief history of machine learning.....	50
3.7	What is Machine Learning?.....	53
3.8	Machine Learning Applications.....	54
3.8.1	Recommendation Systems: .....	54
3.8.2	Image Analysis:.....	54
3.8.3	Text Analysis.....	55
3.9	How Does ML Work?.....	55
3.9.1	Form Training Dataset.....	56
3.9.2	Step 1: Feature Detection and Extraction.....	57
	SIFT and SURF.....	57
	How does SIFT work? .....	58
3.9.3	Step 2 Cluster Features to Create a Dictionary of Visual Words. ....	64
3.9.4	Step 3 Train a Classifier and Verify.....	66
3.9.5	Step 4 Evaluate Classifier Performance. ....	68
	Chapter 4 Cluster Analysis of collected Results .....	70
4.1	Overview .....	70
4.2	Supervised learning.....	71
4.3	Unsupervised learning .....	71
4.3.1	Clustering .....	72
4.3.2	Overlap.....	73
4.3.3	Hierarchical or flat clustering.....	73
4.3.4	Goals.....	73
4.4	K-means.....	74

4.4.1	Steps.....	75
	Step one: initialization of centroids. ....	75
	Step two: Calculate distances. ....	76
	Step four: Update Centroids. ....	78
4.4.2	Clusters number (K value):.....	78
	Silhouette .....	79
	The Gap method .....	80
	Davies Bouldin Criterion.....	82
4.5	Conclusions .....	84
Chapter 5 Results of initial experiments on food images .....		85
5.1	Introduction .....	85
5.1.1	Datasets used in this project.....	85
5-1.1.1	The Portland database (2016 mid year) .....	86
5-1.1.2	The Thwaite Hall database (Year end 2016 and mid year 2017).....	88
5-1.1.3	The School of Engineering database ( 2018 earlier year ).....	89
5.2	The First Experiment .....	91
5.2.1	Step one: Image preparation. ....	91
5.2.2	Step two: Segment objects using the three masks.....	94
	Mask One: Intensity Mask.....	94
	Mask Two: Texture Mask. ....	95
	Mast Three: Colour Mask.....	99
5.3	Challenges and limitations. ....	104
5.4	The second experiment.....	108
5.4.1	The first location: .....	108
5.4.2	The second location: .....	111
5.5	Conclusions .....	113
Chapter 6 The Proposed Algorithm .....		115

6.1	Introduction .....	115
6.2	Identify Foreground and Background .....	115
6.2.1	Circle Hough Transform (CHT) technique: .....	116
6.2.2	Texture technique: .....	117
6.2.3	Colour Technique .....	119
6.3	Foreground Segmentation .....	123
6.3.1	Selecting Cluster Centres (Seeds) .....	123
6.3.2	Distance measure .....	123
6.3.3	Cluster Number .....	124
6.4	Food segmentation .....	126
6.5	Food Type Identification .....	127
6.5.1	Step one: Training .....	128
6.5.2	Step two: Feature extraction. ....	128
6.5.3	Step Three: Cluster features to create a dictionary of visual words. ....	130
6.5.4	Step Four: Train a Classifier to identify food types.....	131
6.5.5	Step Five: Evaluate classifier performance using a confusion matrix.....	132
6.5.6	Step Six: Prediction step and results.....	133
6.5.7	Estimating the proportion of food eaten.....	136
6.6	Conclusions .....	139
Chapter 7 Discussion and Future Work .....		140
7.1	Introduction .....	140
7.2	Specification of the Proposed System .....	140
7.3	Discussion.....	142
7.4	Conclusion .....	145
7.5	Future Work .....	147
References.....		148
Appendix A food record chart.....		153



## **Dedication**

*I would like to dedicate this work to my parents,*

*Ibrahim and Fatemeh*

## List of figures

Figure 1: Example waveform produced by eating potato chips, through the chewing, clean up and conversation phases. Upper plot: sound waveform for three activity stages. Lower plot: chewing cycle detection result .	24
Figure 2: A comparison of the performance of feature detection techniques	34
Figure 3: A comparison of the performance of image descriptors of different sizes and combinations of sizes.	35
Figure 4: (a) Original image in greyscale format. (b) The histogram. (c) Thresholded image in binary format	46
Figure 5: Illustrates the multi-level thresholding.	47
Figure 6: ML workflow Step 1 is to detect, extract, and describe features. Step 2 creates a dictionary of visual words. Step 3 trains the classifier using features extracted from the second step and creates the model. Step 4 tests the classifier using a test image to identify the food type in the image.	56
Figure 7: Create a scale spaces	59
Figure 8: Laplacian and Gaussian approximation	60
Figure 9: Finding interest points	61
Figure 10: Removes unwanted key points. a) all key points. B) high contrast key points c) after edge artefact removal	62
Figure 11: Calculate key point orientation	63
Figure 12: 128-element SIFT feature vector. We looking 16*16 neighbourhood of the key point. but in this example, we compute relative orientation and magnitude in an 8*8 neighbourhood plus divide it into 4 sub windows.	65

Figure 13: SURF approximation.” The  $9 \times 9$  box filters in Fig. 1 are approximations for Gaussian second order derivatives with  $\sigma = 1.2$  and represent our lowest scale (i.e. highest spatial resolution). We denote our approximations by  $D_{xx}$ ,  $D_{yy}$ , and  $D_{xy}$ ” .....66

Figure 14: Confusion matrix for car classifier. Green indicates correct classification while red is misclassified.....69

Figure 15: shows the amount of data created online every 60 seconds, b. Shows the intersection of different disciplines .....71

Figure 16: Comparing supervised learning vs unsupervised learning .....72

Figure 17: K-means algorithm steps. ....74

Figure 18: Top figure: Comparison of random initialisation and K-means. Random initialisation may give low-quality results. Bottom figure: Shows seeding progress using K-means++ ( steps 1to 4).....76

Figure 19: Compute new centroids .....78

Figure 20: Gap method example .....81

Figure 21: Example of gap method failing to estimate K value. ....82

Figure 22: Optimal cluster number selection using the Davies Bouldin algorithm.....84

Figure 23: Flowchart describing the algorithm steps.....93

Figure 24: Original image in greyscale and histogram. The image has 13 different types of food. ....94

Figure 25: Entropy histograms for four channels. Top left for grayscale; top right the red channel; bottom left; the green channel; bottom right; the blue channel. ....96

Figure 26: Egg identified using entropy filter.....97

Figure 27: Histograms of local ranges for grayscale and RGB channels.....98

Figure 28: Histograms of local standard deviations for grayscale and RGB channels. ..99

Figure 29: Histogram for A and B values for the image in figure (Figure 24) .....100

<i>Figure 30: Histogram of the A and B values for tomato. ....</i>	<i>101</i>
<i>Figure 31: Four radish masks. The Radish Mask consists of four masks.....</i>	<i>103</i>
<i>Figure 32: Final mask and extracted radish in colour. ....</i>	<i>104</i>
<i>Figure 33: Examples of extracted foods. ....</i>	<i>104</i>
<i>Figure 34: Example 1 shows challenges and limitations of the multi mask method. ...</i>	<i>106</i>
<i>Figure 35: Example 2 shows challenges and limitations of the multi mask method. ...</i>	<i>107</i>
<i>Figure 36: The Portland data collected in different light conditions. The figure shows two examples, A and B. For both examples the plate image uses captured in different locations in the dining area. ....</i>	<i>109</i>
<i>Figure 37: The Portland data collected in different food arrangements on the same plate. The figure shows two examples, A and B. For both examples the same food was rearranged for each image. ....</i>	<i>109</i>
<i>Figure 38: The SVM classifier identified most of the data correctly, with 76% accuracy. The green (diagonal) indicates correct classification and the red means incorrect classification. ....</i>	<i>110</i>
<i>Figure 39: Shows images for food meals from Thwaite Hall. For day one and day two the user can choose one of three options either A, B, or C. Or he can choose any other types of foods.....</i>	<i>112</i>
<i>Figure 40: Confusion matrix for Thwaite Hall experiment. ....</i>	<i>113</i>
<i>Figure 41: shows CHT performance in detecting the plate in the image. ....</i>	<i>117</i>
<i>Figure 42: Illustration of 3x3-range calculation ....</i>	<i>118</i>
<i>Figure 43: Illustration of texture segmentation applied to test images. . ....</i>	<i>119</i>
<i>Figure 44: Illustration of the food plate represented in nine images for three colour spaces. ....</i>	<i>120</i>
<i>Figure 45: Illustration of the steps in applying the colour segmentation method. ....</i>	<i>121</i>

Figure 46: Eight examples of automated foreground identification..	122
Figure 47: Four different examples of the final results of food segmentation. ....	126
Figure 48 Samples images from the dataset used to train the classifier. shows a sample of images used to train the classifier..	130
<i>Figure 49: The visual word histogram of chips, Pea, and Thai Fish Cake. The x-axis represents visual word index which consists of 800 words and y-axis represents frequency of occurrence.</i>	131
<i>Figure 50: The confusion matrix shows the classifier performance in identifying food data. The diagonal (green) indicates the good performance of the model. The off-diagonals have been misclassified.</i>	133
Figure 51: A example of food segmentation and identification. ....	134
Figure 52: (a) Shows the image of the plate before eating. The plate contains rice, Thai fish cake, and peas. (b to d) Are the segmented images for the image before eating. (e) Shows the image of the plate after eating. The plate contains the rice only, which means that the patient ate the Thai fish cake and the peas. The algorithm estimates the food eaten by comparing the food images before and after eating. The results in Excel format as shown in Table 11.	135
Figure 53 shows six examples(a to f ) of data set used to test the system accuracy to estimate the proportion of food eaten.	138
Figure 54: Trolley system	142

## List of tables

Table 1: The system calculates nutritional and calorie content using a nutritional database .....	27
Table 2: Classification results for five food categories: Indian, American, Italian, Mexican, and Thai.....	31
Table 3: Comparison of food recognition systems. ....	38
<i>Table 4: Characteristics of Classifier Types .....</i>	<i>67</i>
Table 5: Dataset sizes and performance of k value classifier. ....	125
<i>Table 6: Comparison of methods to estimate the k value. ....</i>	<i>125</i>
Table 7: Segmentation accuracy. ....	127
<i>Table 8: Dataset used to train the classifier.....</i>	<i>129</i>
Table 9: Summary comparison between 21 trained classifier (model).....	132
Table 10: The final results of food nutritional values estimation summarized as an Excel table. A shows food types. B weight estimated. C to G illustrates the nutritional values. ....	134
Table 11: illustrates the estimation of food intake. The quantity of food eaten is estimated from the difference between the areas spanned by food types in images before and after eating. ....	135
Table 12 food eaten report where 1 means all of food is eaten and 0 means no food eaten. ....	138

# Chapter 1 Introduction

## 1.1 Project Background

In recent years, there has been an increase in the number of people requiring dietary monitoring. Dietary records are an essential part in the diagnosis and treatment of many health problems, such as cardiac disease, dietary deficiencies and allergies, diabetes, obesity and malnutrition. At the Castle Hill hospital, the food intake by patients is recorded by nursing staff manually on a food record chart (see Appendix A), developed by Dr Tina McDougall, Head of Nutrition and Dietetics for Hull and East Yorkshire Hospitals. The chart records the amount eaten in five categories: nil, 1/4, 1/2, 3/4, and all. Nil means that the patient does not eat at all. If the patient eats part of the meal, a nurse estimates the amount to the nearest multiple of 25%. The chart is divided into meal categories and covers 24 hours. The meal categories are breakfast, snacks AM, lunch, dessert, snacks PM, evening meal, dessert, supper and miscellaneous (Green & McDougall, 2002).

The Castle Hill Hospital food record chart is a good example of food recording systems currently used in many hospitals. The disadvantages of such a system are that it needs to be completed manually by hospital staff. Other similar hospital documentation systems (such as x-ray management) use electronic archiving and recording. Hence, most patients' data has to be saved in an electronic format. Consequently, the food intake data has to be further inputted into electronic format, either manually or by text recognition software. This is costly and can introduce human error. In conclusion, food recording charts do not meet modern hospital requirements. Current work at Castle Hill, in collaboration with the University of Leeds, is exploring the replacement of paper recording with recording the same information onto a tablet computer. However, this still requires the nursing staff to assess the amount of food eaten and input this into the tablet computer.

In elderly people, many chronic diseases are associated with malnutrition (Gariballa & Forster, 2008). Good food records in hospital and in the community after discharge provide a better evaluation of patient nutrition. These records help to identify those at risk of malnutrition. In addition, they may be used for educational purposes (Gariballa & Forster, 2008), or for catering management.

A newspaper article published in August 2010 showed that each year (Graeme, 2010), around 175,000 people are admitted to hospitals suffering from malnutrition. About 185,000 leave hospitals malnourished. The data show that 10,000 malnutrition cases associated with the stay in a hospital. According to this article, the nurses have no time to help patients to eat their meals. More than that during 2007 about 239 died in hospital because of malnutrition. Financially each year the cost of malnutrition to the NHS is £7.3 billion (Daniel, 2010).



*“The Protected Mealtimes Initiative (PMI) was a national initiative that formed part of the Better Hospital Food Programme. The purpose of the PMI was to allow patients to eat their meals without unnecessary interruption and to focus on assisting those patients unable to eat independently”* (National Patient Safety Agency’, (NPSA), 2007). A pilot study (Green & McDougall, 2002) was conducted at the Castle Hill hospital to study how protecting mealtimes affects elderly hospital patients. It found that protecting mealtimes helped in preventing weight loss (0.19 kg/week compared with 0.25 kg/week) and reduction in hand grip strength (0.53 kg v 0.60 kg). Mid-arm circumference increased with mealtime protection (0.03 cm/week), whereas a reduction (0.02 cm/week) occurred in the control group (P = 0.056). Interestingly, it did not find protecting mealtimes to improve the food intake (calories: 1121/day vs 1275/day; protein: 44 g/day vs 50 g/day) (Green & McDougall, 2002).

Overweight and obesity have become major health problems in rich countries like the USA and UK. Obesity can lead to cardiovascular disease, cancer and diabetes. In the United States, obesity reached a total of 36 % of adults and 17% of adolescents in 2009-2010 (Flegal et al., 2012). Self-monitoring of food intake is one of the best approaches for overweight and obesity management (Yon et al., 2006).

The solutions proposed in the last few years do not meet the requirements of hospitals and home care. For example, mobile apps need various steps to record food. Furthermore, the existing mobile apps are not fully automated, as the user has to manually select a type of food from a database (Matsuda et al., 2012) .

Another example of a food recognition system is the so-called Foodi. A group of researchers from collaborating universities (Institute of Health and Society, The Glasgow School of Art and Newcastle University) have been working to develop this hospital food

recording system since 2009. The authors state that the system accuracy needs to be improved, but do not give any details. Furthermore, a nurse assistant is required to use the system. Foodi is not accurate and not an automated system (Moynihan et al., 2012).

The problem of recognizing the type and amount of food on a plate has not been entirely solved and it is still an open research problem. A fast, accurate and portable system would have application in a wide range of areas.

## **1.2 Problem Definition and Justification**

Building an accurate and automated food recording system is not an easy task. This project explores an image analysis approach to record food intake. There are several challenges that make it difficult to identify food from an image. Difficulties relate to the very wide range of foods available, as well as the limitation of identifying food by appearance alone. Food characteristics such as colour, texture and shape are not consistent even for food of the same category for example fried potato can be of a wide range of shapes and colours. The accuracy of food identification is affected by the food style (Bettadapura et al., 2015). A food recognition accuracy of 80% was achieved for South Asian food, but for Thai food, the accuracy was only 43%. Bettadapura postulated that this "could be due to the fact that there is a low degree of visible variability between food types belonging to the same cuisines" (Bettadapura et al., 2015: 6). A range of foods could be hidden beneath a strongly coloured sauce. Some food ingredients are used in most meals and this creates similarity in food colour and texture. This makes it a difficult task for humans to distinguish between food types by their appearance. Variation of a particular food as appearance in colour and texture is a challenge. Furthermore, mixing foods and the associated change in food appearance can make the

same plate of food look quite different. For example, adding ketchup to chips changes appearance radically. The bread rather than the filling determine the appearance of sandwiches, and so type of sandwich is difficult to determine from an image. The same is true for battered and fried foods, where the shape is the only clue to the interior.

Another difficulty is the ambient environment, which includes variables such as brightness, distance between food and camera, as well as camera angle. Light in rooms is not homogenous and it varies from time to time and from place to place. Room lighting changes from day to night and from the middle of a room to near a window. Distance variation between camera and food makes it difficult to calculate food size accurately. Additionally, the position of the camera relative to the food is very important. Some aspects of the food will not be visible from above, e.g. stacks of food, while other food will not be identifiable from the side, e.g. food in bowls.

Although there are difficulties, the problems are surmountable. A great advantage of institutional food is that the range of food available will be constrained and the range available on any particular day may be very small. Furthermore, there will be correlations between foods on a particular plate, e.g. fish and chips, curry and rice, vegetables and noodles. These correlations should greatly increase the accuracy of food recognition. Furthermore, the accuracy required for a useful system is quite low. Manual systems have proven their worth when they can estimate the proportion of food eaten to the nearest 25%.

### **1.3 Aims and Objectives**

The principal aim of this project is to design and evaluate a food monitoring system for use in institutions such as hospitals. Such a system has several components including

the hardware for image capture and data processing, as well as software components such as a food recognition system, a food and nutrition database, and an interface to allow users to interrogate and interact with the system. The objectives of this project are to:

- Specify, acquire and evaluate an image capture system suitable for an institution such as a hospital;
- Develop and evaluate a food recognition system compatible with the image capture system;
- Develop and evaluate a system to estimate the amount of food eaten by comparing images before and after eating.

The demonstration system will not include the database and interface of a commercial system, but will be adequate to demonstrate the potential of such a system in an institutional environment such as a hospital or university.

## **1.4 Thesis Contribution**

A new, automated food monitoring system has been developed and tested. It is able to recognise food items present on a plate. The system is specifically designed for use in institutions such as hospitals, schools and prisons. New knowledge is identified on the complexities of automated food identification.

An algorithm has been developed may be trained to operate in institutions with a wide range of food types. Furthermore, it has been designed to operate with images experiencing a wide range of lighting conditions and background surfaces.

## 1.5 Thesis Outline

The outline of the thesis is provided by the chapter summaries below.

**Chapter 2** reviews the previous work on food monitoring. The literature review enables us to learn from previous work and defines gaps in the current knowledge. The chapter presents the technical specifications of the proposed system.

**Chapter 3** introduces methods and techniques that contribute to solving the problem of recognising food in images acquired by digital camera. The chapter introduces the terminology and techniques that are addressed in more detail later on. Topics covered include digital images, colour spaces, digital image segmentation, thresholding, and Machine Learning (ML). The key principles and popular applications of ML are provided.

**Chapter 4** reviews clustering algorithms, in particular K-means.

**Chapter 5** describes two experiments conducted using different algorithms in image processing and Machine Learning

**Chapter 6** presents the proposed food recognition algorithm. The algorithm steps are image pre-processing, image feature extraction, image segmentation and identification, food feature extraction, and the display of results.

**Chapter 7** System specification, testing, conclusion, discussion, and future work

# Chapter 2 Literature Review

## 2.1 Introduction

Food recording is essential in treating diseases such as malnutrition, diabetes and obesity. Good food records in hospital and in the community after discharge provide a better evaluation of patient nutrition. These records help to identify those at risk of malnutrition. Malnutrition can lead to poor health and identically poor health can lead to malnutrition. The challenge is to identify which one is the cause or effect (Weinsier &Heimbürger, 1997)

This chapter reviews and discusses different approaches to food recording developed by researchers. Each approach has its own objectives, advantages and disadvantages. Some systems, such as food logs, have been developed to meet user needs. Obese patients can use a food log to record their daily meals (Aizawa et al., 2013). Moreover, there are systems developed to help users to improve their eating habits.

A variety of methods has been developed to estimate food intake, ranging from reports based on weighing food portions through to expensive 24-hour monitoring (Flegal et al., 2012) .Four set of methods have been identified. The first approach is a manual method using paper and pen. The second uses on-body sensors attached to the arm, ear and neck to monitor food intake. The third approach is a microwave based measurement

system. The last method analyses food images using image processing and computer vision.

The next few pages discuss the most common approaches to food recognition starting with the simplest and oldest method: paper and pen, and ending with the most recent methods using Machine Learning.

## **2.2 Paper Recording Methods**

Food intake can be recorded using pen and paper, normally on a form designed by a particular hospital or care home to meet specific patient requirements. Medical staff or the patient can be used to complete the report. The resulting reports are on paper and these latter are difficult and costly to process. Even so, some hospitals, such as Castle Hill Hospital in Hull, still use this approach. This is mainly it is fast deploy and cheap and simple to introduce. However, self-monitoring is difficult for patients as they need to keep paper and pen with them at all times (Martin et al., 2009). There are also concerns on the accuracy of self-reported food intake.

## **2.3 On-Body Sensing:**

This approach uses sensors attached to different parts of the human body such as ears, arms and neck. Each sensor records a specific human activity during eating.

Oliver (Oliver, 2008) introduced a system to help patients improve their eating habits by monitoring food intake and providing meal suggestions, as well as lifestyle recommendations. The system uses three sensors to monitor food intake. The first sensor monitors arm movement. Inertial sensors on the wrists and upper back of

participants monitor arm movement. Each time a user's hand moves from dish to mouth, a sensor detects and counts the movement. The system estimates the size of the food morsel carried by each movement. Then the system calculates the meal size by multiplying the number of movements by food size for each movement.

The second monitored activity in the system is chewing, using an ear-mounted microphone to record food breakdown sounds. During chewing, different types of foods produce different sounds. For example, the sound produced by eating boiled potato is different from the sound produced by eating chips (see Figure 1). The system analyses recorded sounds to identify the food type (Oliver, 2008). During chewing, the muscle movement produces sounds considered to be noises that can affect accuracy. Additionally surrounding noise, like talking, can affect accuracy (Oliver, 2008).

The third activity is swallowing, which is monitored using Electromyography (EMG), and a stethoscope microphone. The system measures food intake by measuring the volume swallowed by the user (Oliver, 2008).

The system can be used to improve eating habits. For example, when the user moves his hand quickly, this means that the user is eating too fast. In this case, the system alerts the user to slow down. This improves eating habits and helps to control weight.



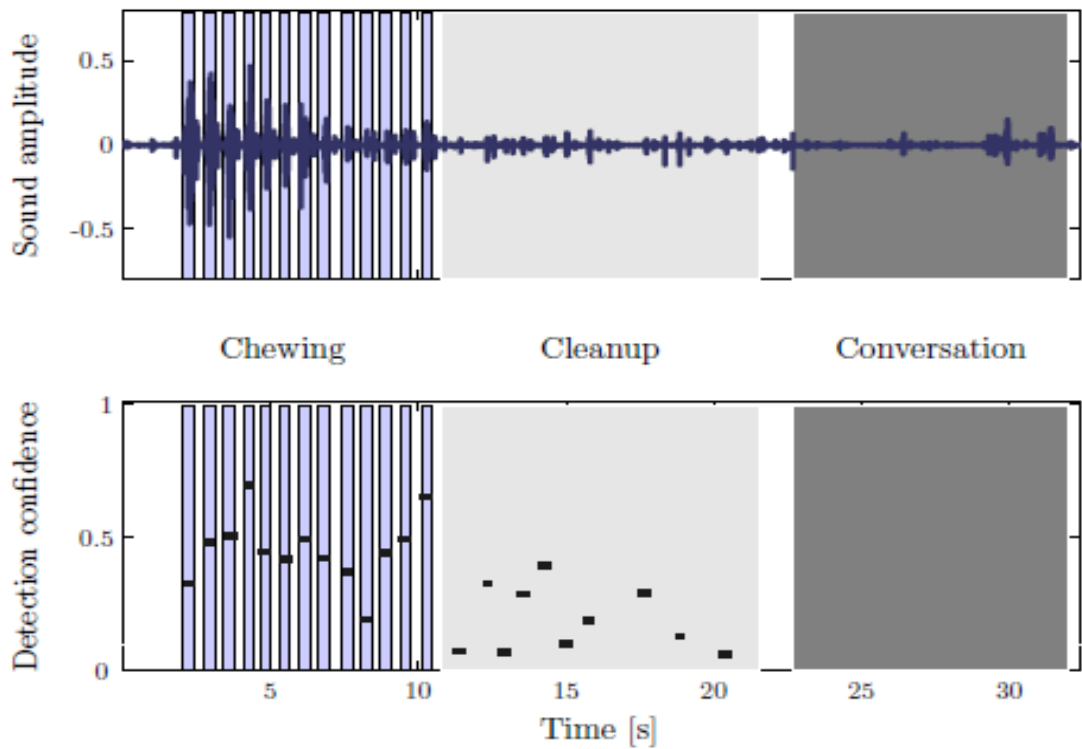


Figure 1: Example waveform produced by eating potato chips, through the chewing, clean up and conversation phases. Upper plot: sound waveform for three activity stages. Lower plot: chewing cycle detection result (Oliver, 2008).

One of the disadvantages of this approach is that it is not comfortable for the participant. In addition to that, hand movements for other non-eating activities can be misclassified as an eating movement. Moreover, ambient sounds can affect system accuracy. The number of foods that can be identified is restricted. The system can recognize only 19 types of foods. General accuracy achieved was 83%.

Furthermore, Fontana et al. (2014) proposed a similar system, considered to be the first wearable system, which can monitor food intake over 24 hours while the participant lives a normal life. The Automatic Ingestion Monitor (AIM) uses three sensors to monitor food intake: a hand movement sensor, a jaw motion sensor and an accelerometer.

## 2.4 Microwave Food Identification

The *General Electric Global Research* team has produced a system to measure food calories using microwaves that travel through the food. The device uses a relationship developed by the team over three years. The prototype unit is still under development and laboratory tests. The unit produces low intensity microwaves that travel through water and fat. During propagation through food they pick-up a specific signature which is analysed by the unit to calculate calories. This approach is still limited to certain types of foods, which are those containing water and fat. Other types of food, such as meat, fruits and greens, are not reported (Griffiths, 2014).

## 2.5 Image processing and computer vision

Automated food recognition has become a popular topic in computer vision and image processing. A number of studies have reported object recognition systems in the last decade. Even so, food monitoring and tracking is still a difficult task. Results are still limited in accuracy and the number of food types recognised is limited. The best reported accuracy achieved was 84% by He et al, (2016), for a restricted selection of foods.

Automated food recognition faces several challenges. Food similarity in colour and appearance makes it difficult for humans and computers to identify food types from the visual appearance. Furthermore, the number of food categories is very large and it differs from one country or region to another. The vast range of food types around the world makes it difficult to develop a global system. One of the major challenges is that a single food can appear quite different. For example, when cooking chips for different periods, the chips change in colour and appearance. Also there are mixed foods, e.g.

pasta with meat or vegetable sauces, and even individual foods can overlap and mix on a plate making it hard to recognise food and estimate size. Categorising food from images leads to other problems associated with image parameters. Images acquired in different environments, such as home, restaurants, institutions or workplace will have different image quality in terms of lighting, background, angle and scale.

The following bullets describe the most popular image processing based approaches:

- A study published in February 2014, sponsored by The Institute of Electrical and Electronics Engineers (IEEE), describes how researchers used image processing to calculate nutrition and calories from a food image. The system uses a built in camera, similar to the one in smartphones, to record an image before and after eating. The differences between the images are used to measure the consumption. The system goes through three main steps (Pouladzadeh et al., 2014). After acquiring the image, the system uses image processing and texture segmentation to identify food types like fish, vegetable and rice. The system uses two images, one from the top and one from the side, to estimate the volume of each portion (e.g., chicken, rice, apple). The image from the top provides the area and the image from the side is used to calculate depth.

After the system identifies the food portions and measures the volume, the system calculates nutritional and calorie content using a nutritional database, See Table 1.

*Table 1: The system calculates nutritional and calorie content using a nutritional database (Pouladzadeh et al., 2014) .*

Food Name	Measure	Weight (grams)	Energy (kcal)
Apple with skin	1	140	80
Potato ,boil, no skin	1	135	116
Orange	1	110	62
Tomatoes, raw	1	123	30
Bread white, commercial	1	100	17
Cake	1	100	250
Egg	1	150	17
Cucumber	1	100	30
Banana	1	100	105
Orange	1	110	62

- Food log is a website service developed by Aizawa et al. (2013) from The University of Tokyo. The web application (<http://www.foodlog.jp>) allows users to upload meal images using a mobile phone or computer. A special algorithm analyses food images and estimates food balance. The result and images are saved in the user's account. This service is useful for overweight people who need to record and analyse their daily meals. The authors claim that their service is the only one available to the public. Food is categorised into five categories based on the Food Pyramid and MyPyramid specification from the U.S. Department of Agriculture. Food categories are: meat/fish/beans, grains, dairy products, fruit and vegetables (Kitamura et al., 2009). In the pre-processing stage, an uploaded image from the user is resized to

320\*240 pixels. The resulting image is then divided into 300 blocks. The system uses image features to assign each block into one of the above five categories, or into an unidentified category. The algorithm uses image colour and texture features to analyse food images. Extracted features are used to create 552 dimensional feature vectors. A Support Vector Machine (SVM) is then used to classify uploaded images into two categories: food and non-food. A machine learning algorithm known as Adaptive boosting (*AdaBoost*) classifier is used to classify food images into the different food categories. The system allows users to correct estimated results using a hybrid Bayesian framework and machine learning. The author claims that accuracy improved from 89% to 92 % after updating the system using personal data. The algorithm also uses personal eating habits (*e.g.*, one user likes to eat fruit at dinner) as a part of food image analysis. This system helps to predict user-eating activities and considers them in food image analysis. This information improves system performance to analyse food images for a specific person (Aizawa et al., 2013).

- Joutou and Yanai (2009) developed a food recording system to track users eating habits. The system is able to recognize 50 different types of Japanese food. The best classification accuracy achieved was 61.34%. The algorithm extracts image features, based on colour and texture, and these are passed to Machine Learning algorithms (see section 3.9). To train the system, images were collected from the Internet and labelled manually. For each of 50 types of food, 100 relevant images were selected yielding a total of 5000 images. The Multi-Kernel Learning (MKL) method is then used to combine extracted features and find the proper mixing weight of features. A Support Vector Machine (SVM) is furthermore used in the classification stages. The authors claim that this system is the first system able to classify food images in a practical way (Joutou & Yanai, 2009).

- Hoashi et al. (2010) continued the development of the previous method to increase the number of food categories from 50 up to 85. A gradient histogram was added as a new image feature. The classification accuracy achieved was similar to that of the previous paper (62.52%) (Hoashi et al., 2010) .
- Bossard et al.(2014), developed a food monitoring system called Food-101. Their food recognition system identifies 101 types of food. The algorithm uses a huge dataset containing 101000 images. One of the advantages of Food-101 is that the dataset is available online at <http://www.vision.ee.ethz.ch/datasets/food-101/>. The authors collected the dataset from a website (foodspotting.com) which provides real world food images. The website helps users from all over the world to upload their food images. Although the dataset is large, the image quality is poor. Mistakes in data labelling and colour intensity are common. This means that there is noise that affects system performance. Large quantities of data are essential when training classification algorithms, but the data needs to have a minimum quality. Mislabelled or noisy data reduces the performance of the classifier (Bossard et al., 2014) . In preparing for feature detection, images are segmented into smaller parts. There are several approaches to achieving this, including the grid method and the super pixel method. Super pixel segments the image into groups of pixels. The pixels in each group are similar to each other. " Superpixel algorithms group pixels into perceptually meaningful atomic regions, which can be used to replace the rigid structure of the pixel grid" (Achanta et al., 2012). To make the segmentation (see Section3.4) and classification tasks less complex, the Food-101 algorithm uses a few dozen super pixels instead of thousands of sliding windows. The super pixel method minimises the number of detectors and makes clustering and classification simpler. The algorithm extracts LAB colour (see Section 3.3) and Speeded up Robust Features (SURF) for each

super pixel (Bossard et al., 2014). The Random Forest method is used to cluster super pixels. For each cluster, the top leaves are used to train a linear binary SVM. The average accuracy achieved is 50.76%.

- Bettadapura et al. (2015) developed a system targeting people eating outside the home. In America, 80% of people eat out at least once a month. The algorithm identifies the restaurant name and location by accessing details of a food image. One assumption is that the subject is equipped with a smartphone and this can provide the image's location using GPS sensor. After identifying the restaurant name and location, an algorithm uses the restaurant menu to analyse food images. The system limits the types of food to those from the restaurant menu. This constraint improves the accuracy of the classification process (Bettadapura et al., 2015). The system was tested using 10 restaurants and five food categories: Italian, Mexican, American, Thai, and Indian. Two colour features and four Scale Invariant Feature Transform (SIFT) (see section 3.9.2) features were extracted from the food image. The algorithm uses SMO-MK multi-class SVM classification as a framework. Results show that there is a clear variation in detection accuracy for different food categories. Table 2 shows summary test results for the six descriptors and MKL (Multiple Kernel Learning) classifier. The six descriptors are Colour Moment Invariant (CMI), C-invariant (normalised opponent colour space), C-SIFT, Hue-Histogram (HH), Opponent SIFT (O-S), RGB-SIFT, and SIFT. One of the disadvantages of this approach is that it needs weekly updates to train the classifier. The study showed that the food category affects algorithm performance dramatically. Table 2 shows the results of classification of five food categories. Indian food is identified to a high accuracy: 80.8%, while Mexican food is identified to a very low accuracy: 43.3%.

Table 2: Classification results for five food categories: Indian, American, Italian, Mexican, and Thai. (Bettadapura et al., 2015).

	CMI	C-SIFT	HH	O-SIFT	RGB-SIFT	SIFT	MKL
American	45.8%	51.7%	43.3%	43.3%	37.5%	29.2%	67.5%
Indian	44.2%	74.2%	55.0%	59.2%	69.2%	65.0%	80.8%
Italian	33.3%	52.5%	67.5%	74.2%	66.7%	49.2%	67.5%
Mexican	36.7%	35.8%	20.8%	37.5%	24.2%	33.3%	43.3%
Thai	27.5%	36.7%	25.0%	33.3%	50.8%	30.8%	50.8%

- He et al. (2016) developed a food recognition system called DiteCam, which is based on identifying food ingredients. Many types of foods are similar in appearance, shape and colour. For example, pasta is cooked in different ways to give a different taste and appearance. However, pasta has the same ingredients even if it is cooked in different ways. DiteCam uses this observation to classify foods according to their ingredients. Each type of food is classified according to its special combination of ingredients. Ingredients are detected based on their shape, texture and location. The Multi kernel SVM classifier is used to classify food types. The algorithm uses a database of 55 types of American food including foods like pasta, salad, cookies and drinks. In total 15262 images are used to train and test the system. The authors claim that the recognition precision of the algorithm was 90% for general food types (He et al., 2016) .



- Kawano and Yanai (2015) proposed a real time food recognition system called FoodCam that uses a set of local features representation called Fisher vector. Usually, food recognition systems send food images to high performance servers for analysis. These processes consume time and require connection to the Internet. The paper proposed two methods for a mobile, real-time, food identification system. The method uses two approaches to detect features, followed by a linear SVM classifier. These methods are suitable for smartphones in terms of memory processing time and accuracy (Kawano & Yanai, 2015).
- A multi-disciplinary research team (The Institute for Ageing and Health, Newcastle University, Glasgow School of Art and University of Reading) designed a food service system called Hospital Foodie. The work lasted three and a half years from 2009 until 2012. The project aimed to find a solution to malnutrition during hospital residence. It started by exploring opportunities for improvements. The team referred to previous studies and interviewed health care staff, patient's families and friends. Subsequently, a prototype image based food monitoring system was built and evaluated. The current system is not accurate in calculating food intake. Calculating food intake is the core of any malnutrition system. Unfortunately, the current prototype does not provide accurate information about calories and nutrition consumed (Moynihan et al., 2012).
- Anthimopoulos et al.(2014) proposed a food recognition approach based on the Bag of Features model. The system aimed to help diabetic patients estimate the amount of carbohydrate in meals. A dataset of 4868 colour images was collected from the Internet. They were categorised into 11 food classes: bread, breaded food, cheese, egg products, legumes, meat, pasta, pizza, potatoes, rice, and vegetables. The

number of images varies from one class to another. For example, the image set for meat includes 174 images and the pizza set has 731 images.

In general, image processing approaches for food recognitions consists of two main stages. The first stage is image description, while the second stage is training the classifier. The stages are described in detail in the following paragraphs. We use the paper (Anthimopoulos et al., 2014) as a case study to explain the steps.

### **2.5.1 First stage: Extract Image Descriptors**

The paper (Anthimopoulos et al., 2014) uses the Bag of Features model to extract descriptors as follows:

#### **Step one: Key points extraction**

Three different approaches are tested to find the most effective technique to detect key points. The three techniques were random sampling, SIFT interest point detectors, and dense sampling. Experimentation showed that SIFT did not give a sufficient number of key points compared to the other techniques. SIFT Dense sampling and the random sampling showed similar performance. (See Figure 2).

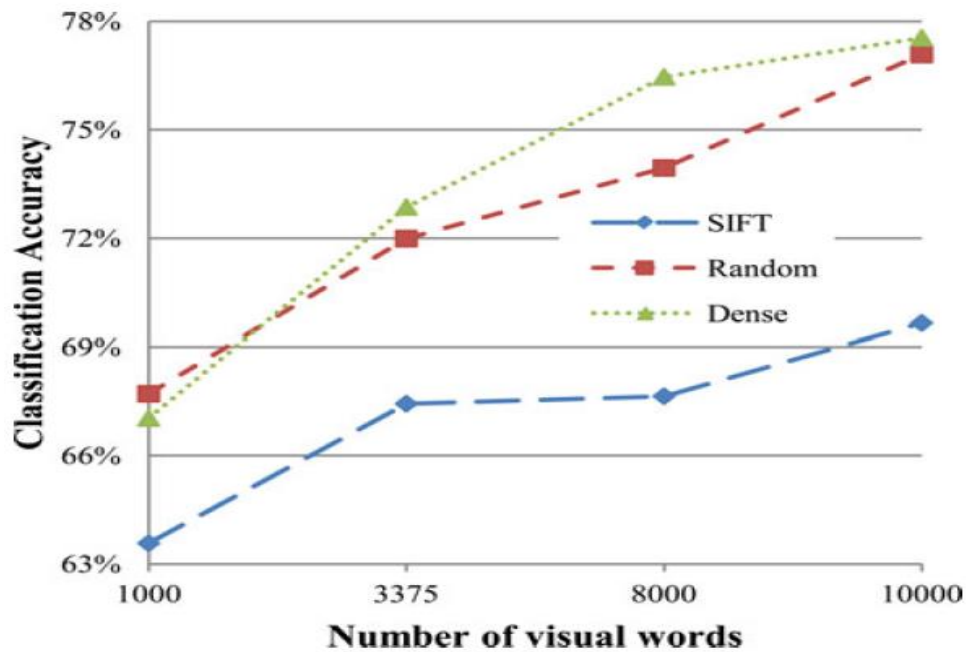


Figure 2: A comparison of the performance of feature detection techniques (Anthimopoulos et al., 2014).

The paper did not consider other approaches such as super pixel and the grid method. For example, Bossard et al. (2014) found that using a super pixel approach is more effective for features detection since it minimises features and makes clustering simpler and more accurate (Bossard et al., 2014) .

The SIFT feature detector is considered to be effective and accurate for image matching. However according to the authors, SIFT was not able to detect sufficient features to train the classifier, with the training images used. SIFT is widely used and considered to be one of the best feature detectors. It has performed well in other food recognition algorithms, although with different styles of food (Anthimopoulos et al., 2014).

A second experiment was conducted to find the best descriptor size to produce optimal results, and to determine if combinations of different descriptor sizes can give better results. Dense sampling techniques were used. The descriptor sizes were 16, 24, 32, and

56. The results showed that small descriptor sizes describe images better than large descriptor sizes. The authors argued that smaller scales yield a larger number of descriptors, which helps to cluster features and creates sufficient visual words. The experiment also showed that the best combination of different scales was 16, 24, and 32. (See Figure 3).

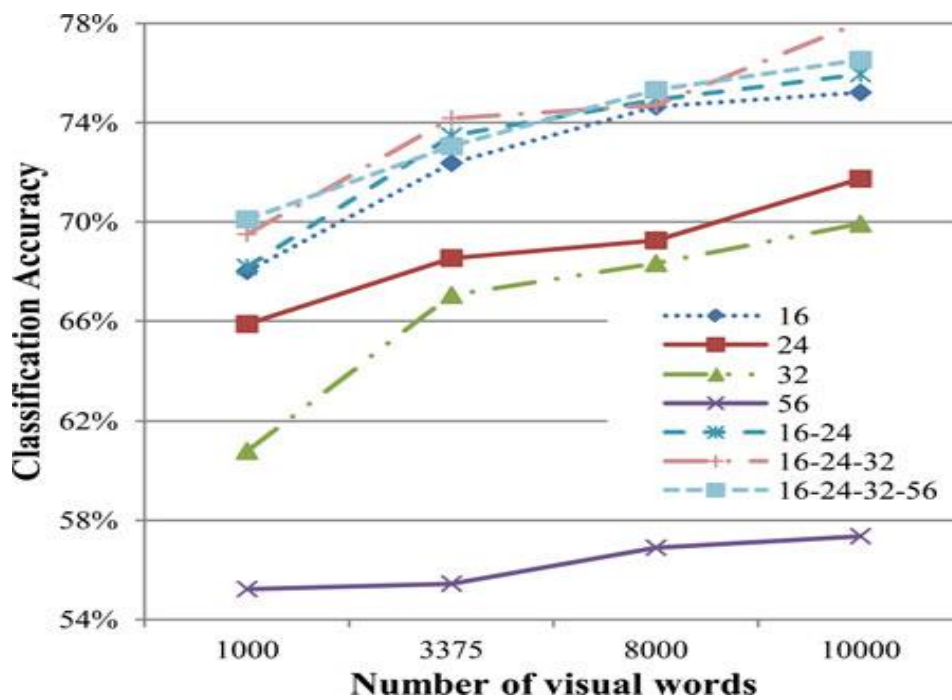


Figure 3: A comparison of the performance of image descriptors of different sizes and combinations of sizes (Anthimopoulos et al., 2014).

### Step two: Key point description

A series of experiments compared different types of descriptors. Half of them were colour descriptors and the others were SIFT and some colour descriptors. The experimental results showed that descriptors based on SIFT gave the best results compared to descriptors based on colour, possibly because SIFT was less sensitive to

variation in image intensity. HSV (see section 3.3.4) SIFT descriptors achieved the best results with an accuracy of 77.8%.

The paper did not mention other colour spaces like LAB (see section 3.3.3), which is less sensitive to light variation. In LAB the image intensity variation is largely localised into the L dimension while A and B describe colour. Descriptors that put more weight on A and B are less sensitive to light intensity.

### **Step Three: Clustering and creating a visual word dictionary.**

Three main factors affect the clustering stage: descriptor number, clustering technique and cluster number. This experiment showed that the clustering technique and cluster number have a large impact on clustering results.

Additionally the paper compares the K-means and hierarchical K-means (HK-means) clustering techniques. HK-means clustering is faster than K-means and gives almost the same accuracy. A numerical test compared the computational intensity and accuracy of the K-means and HK-means algorithms when used to cluster features and create a visual word dictionary. K-means took 22 hours to yield 78% accuracy. HK-means took only 16 minutes to yield 77.6% accuracy (Anthimopoulos et al., 2014). That means that HK-means is faster than K-means by almost two orders of magnitude and gives similar results. Moreover, the experiment showed that when the number of visual words increases, recognition accuracy rate increases.

### **2.5.2 Second Stage: Training the Classifier**

The next stage tested three types of classifier. The authors chose Support Vector Machine (SVM), Artificial Neural Networks (ANNs), and Random Forests (RFs). Results showed that the linear SVM and non-linear ANN with one hidden layer, both yielded

adequate classification compared to complex classifiers like the Radial Basis Function (RBF) kernel SVM , SVMx2, ANN with one hidden layer (ANNwh), and RF. All algorithms were implemented and tested in Matlab.

Table 3: Comparison of food recognition systems.

Paper	Data used in learning and testing	Number of Food categories	Features used for training and testing	Classifier	Results
Food log (Aizawa et al., 2013)	Users upload their food images using a website.	Five food categories: meat/fish/beans, grains, dairy products, fruit and vegetables.	Colour, circle features, and Bag of Features (BoF). All features were merged into a 552-dimensional feature vector.	SVM is applied to classify the image into the binary classes, namely, food or non-food. AdaBoost classifier used to classify into 5 food categories.	After updating the model with personal data, accuracy improved from 89% to 92%.
A food image recognition system with multiple kernel learning (Joutou & Yanai, 2009)	Data collected and labelled manually from the internet. For each type of food, 100 relevant images selected: 5000 images in total.	50 types of Japanese foods	Colour, texture, SIFT and BoF	trained MKLSVM	Achieved 61.34% classification into 50 different types of food .
Image Recognition of 85 Food Categories by Feature Fusion (Hoashi et al., 2010)	35 types of food added to previous study. The total is 8500 images	The total number of food categories is 85.	Colour, texture, SIFT and BoF. Gradient histogram added as new image features.	Trained KLSVM	Similar to previous paper 62.52% for 85 types of foods.
Leveraging Context to Support Automated Food Recognition in Restaurants (Bettadapura et al., 2015)	Collect data from particular restaurant menus using websites such as Allmenus.com, Google Places and Yelp.	5 food categories American ,Indian , Italian, Mexican and Thai	System uses 2 colour descriptors and 4 SIFT descriptors. <ul style="list-style-type: none"> <li>• CMI: Colour Moment Invariants</li> <li>• C-S: C-SIFT</li> <li>• HH: Hue Histograms</li> <li>• O-S: Opponent SIFT</li> <li>• R-S: RGB SIFT</li> <li>• S: SIFT</li> </ul>	SMO-MKL multi-class SVM classification framework	American=68% Indian = 81% Italian = 68% Mexican = 43% Thai = 51%

<b>Paper</b>	<b>Data used in learning and testing</b>	<b>Number of Food categories</b>	<b>Features used for training and testing</b>	<b>Classifier</b>	<b>Results</b>
DietCam (He et al., 2016)	15262 images for 55 food categories. 50% for training and 50 % for testing.	55 food categories	Shape, texture and location	Multi view mult kernel SVM.	
FoodCam (Kawano & Yanai, 2015)	12,905 images for 100 food types. Each food category has more than 100 images	100 food categories	First method: BoF, colour histogram, kernel feature maps. Second method: HOG, colour patch descriptor	Both methods use linear SVMs	79.2 % classification rate

## 2.6 Research Gaps

Automatic food recognition faces some difficulties such as variation in food appearance and ambient environment. These conditions have made the development of an algorithm for identifying food, a challenging task. In recent years, Machine Learning has been applied to image recognition in research studies. The Bag of Features method has shown itself to be a powerful tool to analyse large volumes of data such as food images (Anthimopoulos et al., 2014).

Success in automatic food recognition depends upon the classes of foods that are encountered. Regional cuisines show marked differences in the ability of automatic systems to recognise food types. Systems that limit the possible classes of food types exhibit more accurate categorisation. For example, when the foods were limited to those on the menu of a single restaurant then better performance are achieved.



When the application is the monitoring of food consumption in an institution such as a hospital, then the types of food on offer at any time are very limited. Furthermore, it has been demonstrated that being able to estimate the amount of food eaten to the nearest 25% yields significant health benefits. Although systems have been developed and tested in other places, a system optimised for UK hospitals would have large impact. This project will focus on the development of such a system.

Chapter Two describes the steps required to apply Machine Learning to food recognition. Each step can be achieved in many different ways. For example, there are many ways to define, detect and record image features. Anthimopoulos et al.(2014) proposed an optimized method using a Bag of Features model to identify food types in digital images. However, the authors did not considered several important techniques used in Machine learning. These more modern methods need to be evaluated and so the Anthimopoulos results may be improved upon. For example, there are many approaches that can be used to detect key points, but Anthimopoulos only investigates three: SIFT interest point detector, random sampling, and dense sampling. Other techniques, such as the grid method, and the super pixel approach, which give good results (Bossard et al., 2014), have not been tested, let alone tested on UK hospital food.

The Anthimopoulos study uses a range of tools to describe image features. Some of these features are sensitive to light variation, such as SIFT<sub>rgb</sub> and RGB colour space. Other colour spaces which are less sensitive to light variation were not tested by the study. This suggests the method could be made more robust.

# Chapter 3 Background on Image Processing and Machine Learning

## 3.1 Introduction

This chapter introduces the most common techniques used in food recognition systems. It defines the terminology and techniques used later in the thesis .

## 3.2 Digital Images

Digital images can be acquired by devices such as a smart phone, camera or a scanner. When a picture or photo is captured, it is converted to an array of picture elements (pixels) where each is assigned a set of numbers which specifies the colour. This process is called digitalisation. Digital images may be stored in a range of formats optimised for different purposes. A digital image may be manipulated in complex ways by a computer to display, store and process the photo.

Two common storage formats are PNG and JPEG. These differ in the way digital information is compressed and the way colours are coded. JPEG (Joint Photographic Experts Group) format uses hierarchical loss compression and was designed for storing

high resolution digital camera images in the minimum amount of memory. The compression makes it difficult to edit the image without fully decompressing the file. PNG (Portable Network Graphics) uses a non-loss compression and is an upgrade from the early GIF format that also introduced transparency and more colour definition. For these reasons, it is commonly used for web images.

### **3.3 Colour Spaces**

There are many ways of specifying colour in digital images. Full colour specification requires a power spectral density across the full visible spectrum. However, human eyes cannot distinguish all colours due to the biological limitations introduced by the rod and cone cells in the eye. Colours are usually represented by vectors of three numbers in formats such as HSV, LAB, and RGB (red green blue). Most computer monitors use sRGB (standard RGB) to represent image colours. The Matlab Image Toolbox supports various colour spaces to represent images in formats that ease specific calculations

The system used to represent different colours is called a colour model. A colour model is a mathematical method to define colours. Each colour model uses different primary colours. For example, the RGB model uses red, green and blue to describe the colours. The CMYK model uses cyan, magenta, yellow and key (or black) to define colours. CMYK is usually used in printing. HSL or HSV stand for hue, saturation, and lightness or value. Colour models are independent of device.

#### **3.3.1 Colour Model, Space, CIE 1931**

In computers, colours are specified by a vector of typically three numbers that form a colour space. Each of the numbers is stored as a binary number with a specific number of bits. If  $nb$  bits are used for each colour in a three dimensional colour space then the

total number of colours that can be specified is  $2^{3nb}$ . The colour space maps a set of colours to the visual perception of the human eye. In 1931 the International Commission on Illumination (CIE) defined the first colour space known as CIE 1931. CIE 1931 is used as a standard for human colour perception.

A colour space is a group of colours that are supported by a device to print, save or display. For example, the number of colours that can be displayed by an Adobe RGB colour space is greater than the number of colours that can be displayed with a sRGB colour space.

### **3.3.2 Adobe RGB, sRGB**

Standard RGB (sRGB) created in 1996 uses 8 bits per colour. It is a relatively small colour space and easy to reproduce and display using computer monitors and other devices. This makes sRGB widely used.

Adobe RGB was defined in 1998 and is larger than sRGB. It was designed to span the colours in the CMYK model used by printers. Mismatch between the sRGB used by computer screens and Adobe RGB used by many printers is the main cause for colour disparity when printing photographs.

### **3.3.3 CIELAB Colour Model**

The CIELAB colour model is a three dimensional model developed in 1978 based on earlier work on visual perception in 1948. The three colour components are L for lightness and A and B for the colour components green–red and blue–yellow. The original motivation was to match human colour perception so that all colours perceived to be the same by the human eye are allocated the same LAB coordinates.

### **3.3.4 HSV colour model**

HSV colour model is based on a cylindrical coordinate representation of points in an RGB colour model. HSV stands for hue, saturation, and lightness or value. In HSV cylinder H which is cylinder angle, S which is cylinder radius, and V which is cylinder height and it is a translation from the RGB colour model. HSV use to generate high quality computer graphics and image analysis.

## **3.4 Digital Image Segmentation**

The purpose of image processing is to analyse an image and extract information from it. In many cases, the image needs to be divided into coherent parts or regions. The purpose of image segmentation is to split the digital image into objects or regions that share a particular feature. Image parts are not equally important. Therefore, the purpose of segmentation is to extract the important parts. This increases the accuracy and speeds up image processing. For example, the important part of a food image is the food. Therefore, the purpose is to segment the food image background and keep the food image. It is a completely different process from image enhancement, which aims to improve the image quality, for example by removing noise to make the image more clear visually.

To segment an image, an algorithm must allocate each pixel to a set of pixels that are similar in properties such as colour, texture or region. This simplifies the image and represents it in ways that help the computer to analyse it.

The success of many image processes is based on segmentation. This makes image segmentation one of the most important and difficult tasks in digital image processing.

The input and output of segmentation are usually images.

The pixel is the smallest single component of an image. Therefore, most image segmentation approaches depend on pixel values. For example, thresholding groups pixels based on their values. Pixel values above the threshold value are considered to be foreground and those below the threshold value are considered to be background. (see next page section 3.5). In general, there are two different approaches to image segmentation. The first method is the discontinuity based approach and the second is the similarity based approach.

### **3.4.1 Similarity Segmentation**

Similarity segmentation focuses on pixels that have similar properties, such as colour or texture. There are many different segmentation techniques that use pixel similarity, such as thresholding and the region growing approach.

### **3.4.2 Discontinuity Segmentation**

Discontinuity segmentation aims to detect changes of pixel features in the image, such as edges, lines and isolated points.

In edge detection segmentation, objects are defined by their boundaries using one of the discontinuity operations. To be able to define or identify objects, the boundaries should be complete. Noise in the image, or the eclipse of one object by another, makes it difficult to identify objects from their boundary shape.

## **3.5 Thresholding**

Thresholding is a process that separates specific objects in an image by dividing the image into two parts: foreground (the object of interest) and background (the rest of the image). To threshold the image, a binary image is created from a greyscale image

using a threshold value. The pixel value may be a colour intensity, texture parameter or a wide range of other measures. A suitable thresholding value can be defined using a histogram of pixel values. Pixel values higher than the threshold value are foreground (object) and are set to a pixel value of one. The pixel values that are lower than the thresholding value are background and these pixels are set to zero, (see Figure 4).

Thresholding can be represented by the following:

$$g_{ij} = \begin{cases} 1 & f_{ij} \geq T \\ 0 & f_{ij} < T \end{cases} \quad (3.1)$$

Where:  $g_{ij}$  is the thresholded image value for pixel  $(i,j)$  and  $f_{ij}$  is the original pixel value. The threshold value is T.

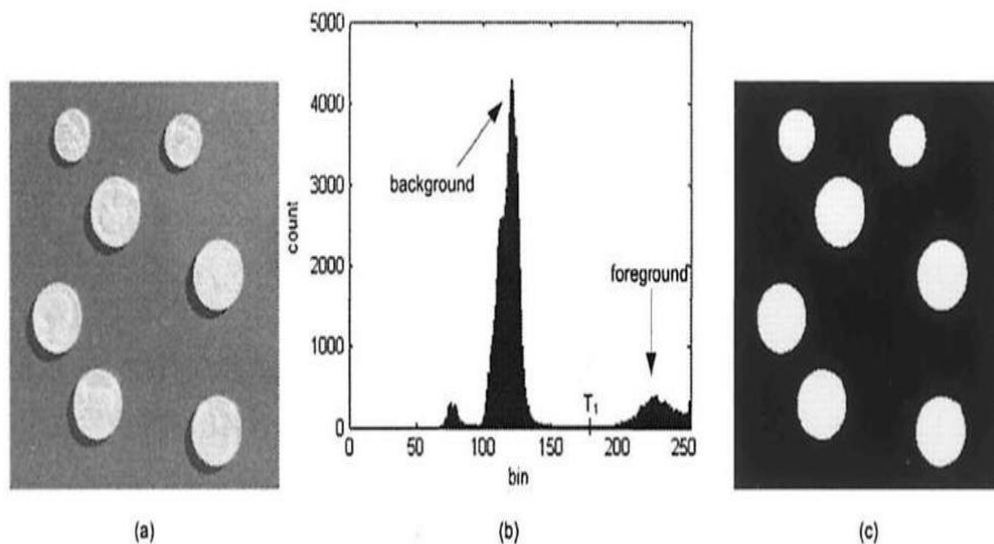


Figure 4: (a) Original image in greyscale format. (b) The histogram. (c) Thresholded image in binary format (Qureshi, 2005).

### 3.5.1 Threshold Different Objects

In the previous thresholding example, the pixel values are concentrated into two intensity regions, one brighter with high pixel values, and the other darker with low pixel values. The two regions create two peaks and one valley in the histogram. However most

images are not as simple as this. Many images have more than one object. In this case more than one  $T$  value is required.

The histogram in Figure 5 suggests that there are three regions, one background and two objects separated by two threshold values,  $t_1$  and  $t_2$ . The objects and background can be separated using the following conditions:

$$g_{ij} = \begin{cases} 0 & f_{ij} \geq T_2 \\ 1 & T_2 > f_{ij} \geq T_1 \\ 2 & T_1 > f_{ij} \end{cases} \quad (3.2)$$

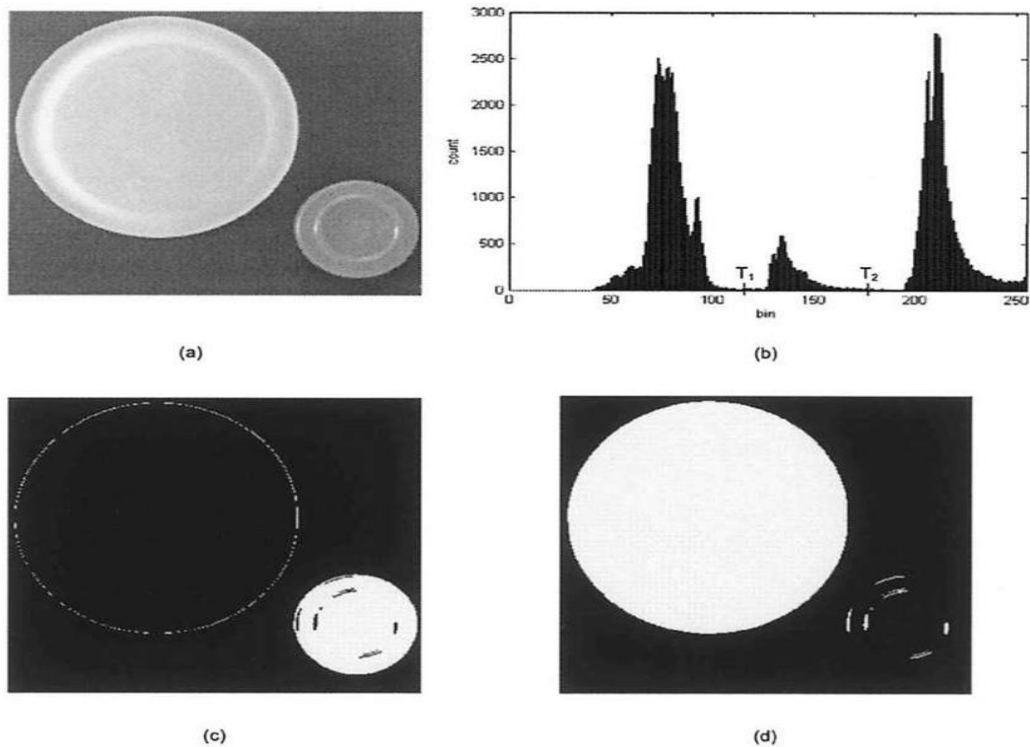


Figure 5: Illustrates the multi-level thresholding. (a) The original grayscale image contains two objects with different brightness level on a dark background. (b) Image histogram with three peaks, the right peak representing the large bright object, the left peak represents the dark background, and the middle peak represents the small gray object. The two threshold values ( $T_1=125$  and  $T_2=175$ ) can be used to split the two objects from the dark background. (c) Segment the small gray object from the original



image (a) using a value of  $T_1=125$  and  $T_2=175$ , All pixels inside the range  $[T_1-T_2]$  set to one and the pixels outside of the range  $[T_1-T_2]$  set to be zero. (d) Segment the large bright object from the original image (a) using a threshold value  $T_2=175$ , All pixels less than  $T_2$  set to zero and the rest pixels set to be one. (Qureshi, 2005)

### 3.5.2 Global and Local Thresholding.

The threshold  $T$  may be a function of position within the image:

If  $T$  changes over the image, then thresholding operation it tests against function  $T$ .

$$T = T [(x, y), p(x, y), f(x, y)] \quad (3.3)$$

Where:

$(x, y)$  = pixel location in the image.

$f(x, y)$  = pixel value at location  $(x, y)$ .

$p(x, y)$  = propriety of the neighborhood around  $(x, y)$  location. Neighborhood propriety can be the average of pixels value around location  $(x, y)$ .

As it's clear from the equation above threshold  $T$  is a function of pixel location, pixel intensity value and local neighborhood property around the pixel location  $(x, y)$ .

Threshold can be a value of any of these three combinations. If the threshold is a function of only pixel value, then this is known as global threshold. When the threshold is a function of pixel value and local neighborhood property, then this is recognized as local threshold. Dynamic or adaptive threshold is a function of all three combinations.

The following equation represents three types of threshold.

$T = T [f(x, y)]$  global threshold,

$T = T [p(x, y), f(x, y)]$  local threshold.

$T = T [(x, y), p(x, y), f(x, y)]$  Dynamic or adaptive threshold.

### **3.5.3 Automatic Thresholding**

An automated system needs to determine a thresholding algorithm valid for a dataset of images, without further human intervention. Automatic thresholding may be used when illumination between objects and background is not clear and when the image has more than one homogeneous region. An iterative algorithm may be used to determine a global threshold to divide an image into two regions. Thresholding is repeated with each new threshold calculated as the unweighted mean of the two region averages. This simplistic approach fails where objects have significant texture or blur into each other.

## **3.6 Machine learning**

Machine Learning (ML) algorithms are demanding in terms of computation and memory. Only over the last decade have computers become powerful enough for ML to be applied to significant problems, initially by large corporations such as Google, Microsoft and Netflix (Gomez-Uribe & Hunt, 2016). Nowadays ML is the most used technique to solve food identification problems. Additionally, ML is useful for big data analysis because it allows a user to analyse a part of it to extract and extrapolate that part from the whole.

Food recognition fits particularly well within ML techniques as a complex image processing method. The ML approach needs a large set of food images to train models. Collecting food images has become easier as digital cameras have been integrated into

phones and image collections become accessible over the internet. Web sites like Google Image allow users to share food images.

### **3.6.1 A brief history of machine learning**

until 50 years ago machine learning was a part of science fiction. Nowadays it has become an important part of our daily life. This section gives a brief history of machine learning.

#### **1950 - The Turing Test**

Alan Turing was a renowned scientist in the field of information technology. He argued that the computers had become more intelligent than humans. He created the Turing Test to find out if a computer has intelligence. The test involves an interrogator staying in one room who receives textual messages from another room. The interrogator objective was to detect whether he was chatting with a human being or a computer. If the interrogator is unable to find out whether he was chatting with a computer or a human beings, this means that the computer was indistinguishable from a human and passed the Turing Test (Machinery, 1950).

#### **1952 - First Computer learning program**

Arthur Samuel was an American pioneer in the field of computer gaming and artificial intelligence. In 1952 he developed the first computer learning program, which helped a computer to improve performance at the game of checkers. The more it played, the more the computer's performance improved (Machinery, 1950).

#### **1956- Birth of Artificial Intelligence (AI)**

in 1956 Martin Minsky and John McCarthy held a conference at Dartmouth during which , AI acquired its name. AI refer to machines that are able to perform duties that are characteristic of human intelligence, like understanding language, pronouncing words, learning, and recognizing objects.

### **1957 -The Perceptron**

Frank Rosenblatt introduce the Perceptron, which is considered as the first artificial neural network. A Perceptron is a supervised learning of binary classifiers. The algorithm helps neurons to learn from the dataset (Rosenblatt, 1957).

### **1967 - Pattern Recognition**

The nearest neighbour algorithm was introduced in 1967. It uses very basic pattern recognition to compare a new object with the existing data. This is considered to be the birth of pattern recognition (Cover & Hart, 1967).

### **1979 -The Stanford Cart**

A student from Stanford University invented a cart that could navigate in a room. A computer program was developed to use images captured from an onboard TV system to drive through cluttered spaces (Moravec,1983).

### **1985 - NETtalk teaches itself to pronounce new words**

Terry Sejnowski and Charles Rosenberg invented a neural network that was able to teach itself how to pronounce new English words. Early outputs were like gibberish. After training NETTALK it could clearly pronounce 1000 words and after a week it could pronounce 20000 words (McCulloch, et al., 1987).

### **The 1990s -Machine Learning Applications**

The 90s are considered as one of the golden eras of machine learning, which became very famous in fields like data mining, web applications, and language learning. The intersection of statistics and computer science moved machine learning further towards data analysis approaches. Large data motivated scientists to create intelligent systems that were able to learn from large data. In addition to the development in software, the hardware improved dramatically. Computers became powerful and able to analyse huge data. In 1997, the IBM computer Deep Blue beat the world chess champion. Thereafter, there have been many more achievements in the field of machine learning (Hsu, 1999).

### **The 2000s - Adaptive programming**

Since the start of the new millennium, many businesses have invested more researches more in machine learning. This led to an explosion of adaptive programming. Referring to programs capable of upgrading themselves, based on the data they receive.

### **2001 - Natural-language understanding (NLU)**

AT&T developed NLU, which could that be able to answer questions posed by humans. The system designed answers to user queries by determining the best information from an electronic database (Hirschman & Gaizauskas, 2001).

### **2012 - GoogleBrain**

Jeff Dean, the leader of Google's artificial intelligence division, created a deep neural network to detect patterns in images and videos. Later, it was developed to be able to detect objects in YouTube videos (Dean, 2017).

### **2014 – DeepFace**

Facebook created a deep learning facial recognition system that can recognize people's faces in images. The system was trained on images uploaded by Facebook users.

### **2015- OpenAI**

OpenAI is a non-profit artificial intelligence company that aims to develop a safe AI that can help humanity (Dean, 2017).

### **2015 - Amazon Machine Learning**


Amazon Machine Learning is a web service that allows companies to easily build smart applications. For example, they can use amazon machine learning to analyze product reviews, recommend corrective actions to their products, and respond to customer feedback (Dean, 2017).


### **2016 - Google DeepMind**


Google DeepMind defeated Go player Lee Sedol, five games to one. The game is one of most sophisticated board games. Professional players assest that the algorithm made moves that had never been seen before.

## **3.7 What is Machine Learning?**

Machine learning (ML) is a subdivision of artificial intelligence (AI) that allows a computer to develop an algorithm to do a specific task. In ML, the computer learns from examining the data, rather than being programmed by a human. The purpose of ML is to make intelligent decisions based on data analysis. ML is a way of modelling data to make predictions and build intelligent applications. An example would be the use of a large dataset of images to train the prediction function  $F$  to recognise features, to get the desired output as shown below.

F (  ) = “apple”

F (  ) = “tomato”

F (  ) = “cow”

## 3.8 Machine Learning Applications

The following are examples of common applications of ML to solve real world problems.

### 3.8.1 Recommendation Systems:

Big corporations like Amazon, Google and Netflix use ML to predict content that will appeal to the user. The Netflix personalized recommender system suggests suitable films based on other movies the user has seen. The recommender is trained by the selection of movies made by all the users. This helps Netflix get a better understanding of the user’s intention and engagement (Gomez-Uribe & Hunt, 2016).

### 3.8.2 Image Analysis:

Web pages like Facebook use algorithms that learn from user photos. For example, when a user uploads a photo of a famous person, like Nelson Mandela, the algorithm extracts a special feature of the photo and saves it. When another user or the same user uploads Nelson Mandela’s photo, the system can recognize and tag it (Becker et al., 2008).

Autonomous vehicles or driverless cars use ML algorithms to identify road edges, road signs and other objects by analysing images taken by a video camera. ML enables the driverless car to make the right decision at the right time. The decision can be turn right, turn left or stop (Berger, 2014).

### **3.8.3 Text Analysis**

Text analysis is used to extract and classify information from text such as emails, documents, and reports (Guzella & Caminhas, 2009). Common ML applications of text analysis include the spam filter algorithm used to classify emails into ham or spam based on the email subject and contents. Another application is automatic language detection and news categorisation into topics such as sport, politics or technology.

## **3.9 How Does ML Work?**

We can summarize ML workflow in four steps also represented in Figure 6 .

Before starting we need to Label food images and categorise them in groups such as apple, pasta.... pizza. This forms a training dataset.

Step 1: Detect interest points (key points) using tools like SIFT, the grid method and random sampling; then extract the feature descriptors (feature extraction).

Step 2: Cluster features to create a dictionary of visual words.

Step 3: Train a classifier to create a model then verify by classifying unknown test images.

Step 4: Evaluate classifier performance using tools like the confusion matrix.

The following sections describe the steps from step 1 to step 5 in more detail.



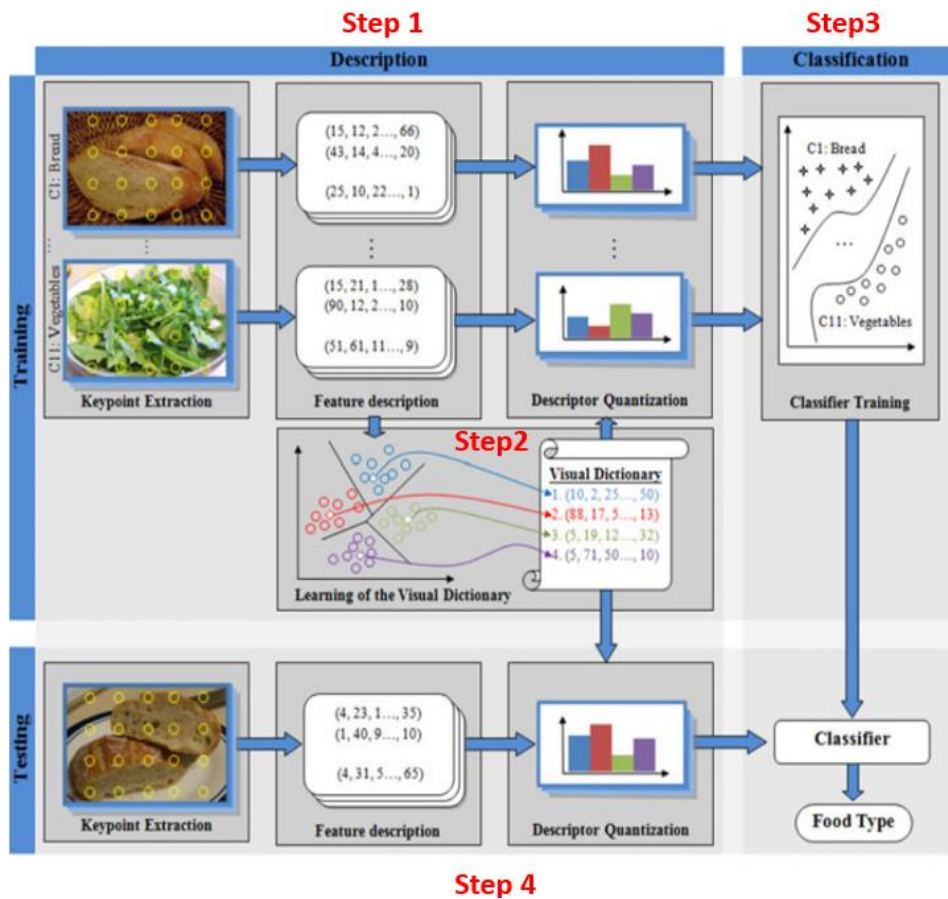


Figure 6: ML workflow Step 1 is to detect, extract, and describe features. Step 2 creates a dictionary of visual words. Step 3 trains the classifier using features extracted from the second step and creates the model. Step 4 tests the classifier using a test image to identify the food type in the image (Anthimopoulos et al., 2014).

### 3.9.1 Form Training Dataset

Data is fundamental to machine learning. Training the model starts with raw data, which can be images. The choice of data depends upon on the application. When identifying food in an image, the data are a large number of food images. The images should represent objects in the range of different conditions expected to be encountered, i.e. a range of illumination, angles and environments. In this example, training data should be labelled, i.e. is food images need to be categorised into groups. Each category

includes images of a specific type of food like fish, rice or orange. These images labelled with their food category form the training dataset.

Normally, training data are large. The amount of training data may be limited by the effort required for human categorisation or computer resources such as memory or computation time.

### **3.9.2 Step 1: Feature Detection and Extraction.**

Feature detection aims to find special points in an image, like corners, edges and regions with special properties. Feature extraction computes descriptors from the pixel value around each point of interest (corners, edges and regions). There are various feature detection techniques, such as Scale-invariant feature transform (SIFT) (Lowe, 2004), and Speeded up Robust Features (SURF) (Bay et al., 2006).

Feature selection depends on the task, for example, food classification or face detection. The specific task helps decide what important features need to be extracted. The final format of features is a numeric representation of the raw data.

The next few pages describe a common features, detector and descriptor, called SIFT. SIFT is a good example to show how to detect and describe features. The input to SIFT is an image and the output is a numeric representation of the image.

#### **SIFT and SURF**

SIFT (Scale Invariant Feature Transform) is a feature descriptor that is able to find local features invariant to transformation. The transformation can be geometric (like scale) and/or photometric (like illumination). This means that the algorithm can detect objects in different conditions, such as various scales, rotations and illuminations. SIFT is a relatively new technique that has had a large impact on image matching (Lowe, 2004).

In 2006, the Speeded-Up Robust Features (SURF) algorithm was published by Bay et al. (2006). SURF is a less computationally expensive version of SIFT.

### How does SIFT work?

SIFT detects the features in different images independently and compares image similarity by comparing features in each image.

SIFT operates in two main stages: feature detection and feature description. The first stage is finding points of interest (key points). The second stage computes feature descriptors. We can summarize SIFT operation in the following steps.

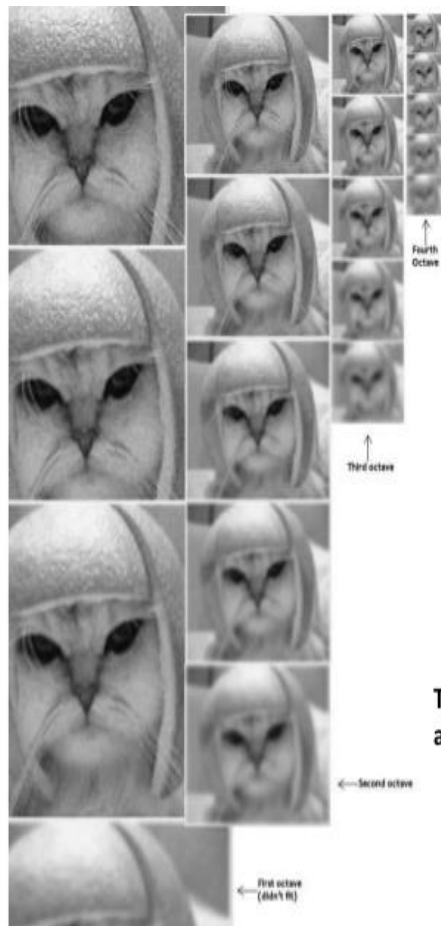
#### 1- Create a scale space:

Take the original image and iteratively reduce to half the number of pixels in each dimension. Also, create blurred images by applying a Gaussian blurring filter. The result is a set of images with a range of scales and resolutions, (see Figure 7). The Gaussian blurred image  $L_\sigma$  is calculated from the original image  $I$  by convolution with a Gaussian kernel with standard deviation  $\sigma$ ,  $G(i, j; \sigma)$ :

$$L_\sigma = I(i, j) * G(i, j; \sigma) \quad (3.4)$$

where the Gaussian kernel is given by:

$$G(i, j; \sigma) = \frac{1}{2\pi \sigma^2} e^{-(i^2 + j^2)/2\sigma^2} \quad (3.5)$$



## Construction of a scale space

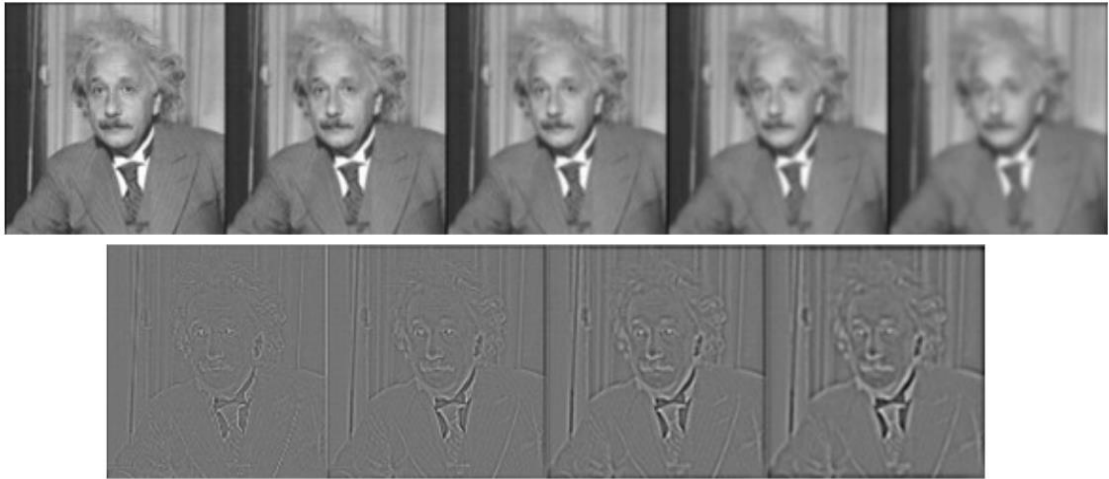
SIFT takes scale spaces to the next level. You take the original image, and generate progressively blurred out images. Then, you resize the original image to half size. And you generate blurred out images again. And you keep repeating.

**The creator of SIFT suggests that 4 octaves and 5 blur levels are ideal for the algorithm**

Figure 7: Create a scale spaces (Utkarsh, 2010).

### 2- Laplacian and Gaussian approximation.

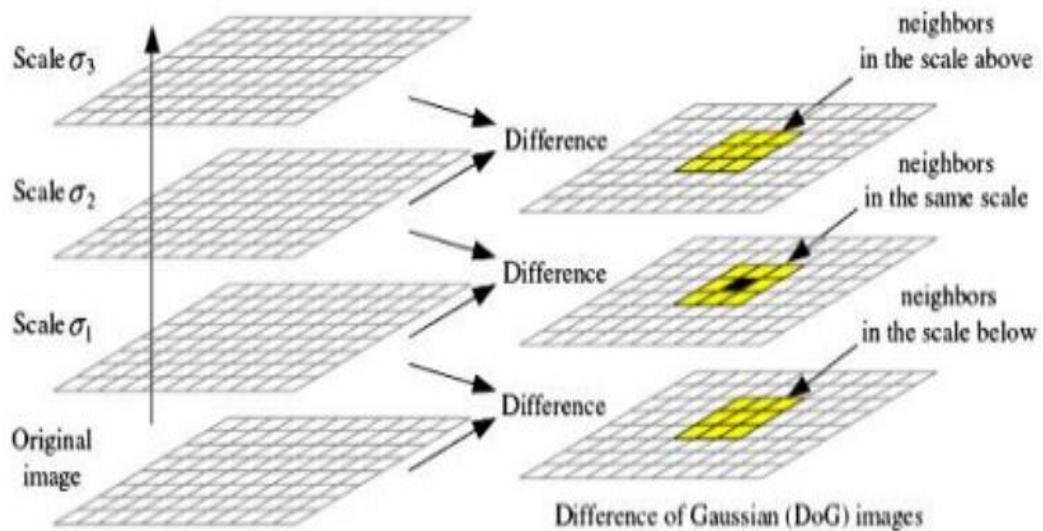
The difference between the original and blurred images highlights the high contrast edges. Figure 8 shows several scales where each scale has five blurring levels. This yields four images formed from the difference between adjacent blurred images, known as Difference-of-Gaussians (DoG) images (see Figure 8).



*Figure 8: Laplacian and Gaussian approximation (Magdalene, 2016) .*

### 3- Finding points of interest (key points)

To find a key point, each pixel is compared with its neighbours in the same image, and also in the images with different blurring, as shown in Figure 9. A pixel may be compared with 8 pixels in the same image and 9 pixels in the image above and also 9 pixels in the lower image. This pixel is a key point when it has the highest or lowest value among neighbours. If an image yields many key points then some may be disregarded, see Figure 10 The next step explains how to eliminate unwanted key points.



## Difference of Gaussian Formation

Figure 9: Finding interest points (key points) (Utkarsh, 2010).

### 4 - Remove unwanted key points (interest points)

The previous step may produce a very large number of key points and it is likely that not all of them are useful. Some key points have low contrast and others are located along an edge. To remove low contrast key points we delete all key points that have DoG (Difference-of-Gaussians) lower than a certain threshold. A corner detector deletes the key points located along an edge. The Harries Corner Detector can distinguish between interior corners and edge artefacts (Estrada et al., 2004). This calculation applies at all scale levels and key points are detected from various scales, (see Figure 10).

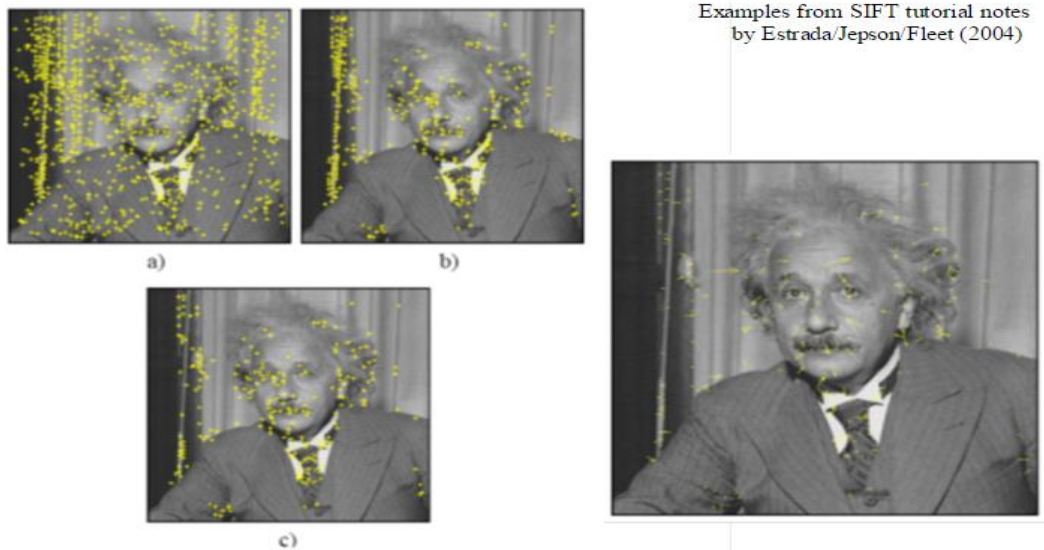


Figure 10: Removes unwanted key points. a) all key points. B) high contrast key points c) after edge artefact removal (Estrada et al., 2004).

#### 5 – Calculate key point orientation

Key point orientation calculations use the region around the pixel.

- Calculate the orientation and magnitude within a specific region around a key point. The region size is determined by the scale size ( $L$ ).
- The following equations are used to calculate orientation magnitude  $m$  and orientation angle  $\theta$  :

$$m_{ij} = \sqrt{(L_{i+1,j} - L_{i-1,j})^2 + (L_{i,j+1} - L_{i,j-1})^2} \quad (3.6)$$

$$\tan(\theta_{ij}) = \frac{L_{i,j+1} - L_{i,j-1}}{L_{i+1,j} - L_{i-1,j}} \quad (3.7)$$

- A histogram is formed from the orientation angles, (see Figure 11), and the highest peak defines the orientation. For example, the highest peak in Figure 11 is bin 3 and so this is the dominant orientation for this key point at this scale.
- Each key point is represented by an arrow with the calculated magnitude and orientation. Arrow size is also determined by the key point scale. For example, the magnitude of a key point magnitude of scale 5 is bigger than that of a key point of scale 1.

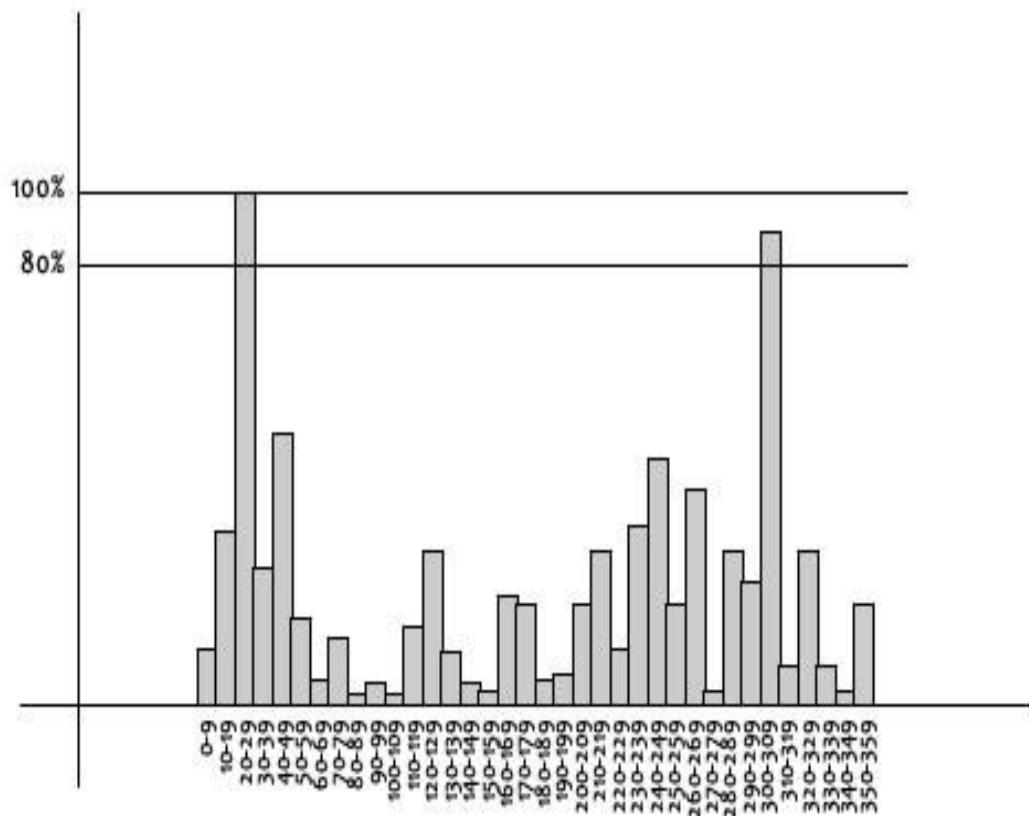


Figure 11: Calculate key point orientation (Utkarsh, 2010)



## 6- Create a SIFT feature

From the previous steps we have features of key points.

- Scale  $\sigma$ .
- Gradient magnitude and orientation.
- Location  $(i,j)$ .

A window is calculated, centred on each key point and with the key point orientation and magnitude. A SIFT feature vector is calculated for each key point. Orientation is calculated for many sub-windows within the key point window. For 16 sub-windows and 8 discrete orientations, a feature vector of length 128 describes each key point, (See Figure 12).

Not all key points are useful and many do not contribute to classification.

### **3.9.3 Step 2 Cluster Features to Create a Dictionary of Visual Words.**

The previous step yields a large number of key point feature vectors, which can be grouped using K-means clustering. The number of clusters (also called vocabulary) is an input parameter. Each cluster represents a single visual word. The number of clusters is the number of visual words (features) in a Bag of Features.

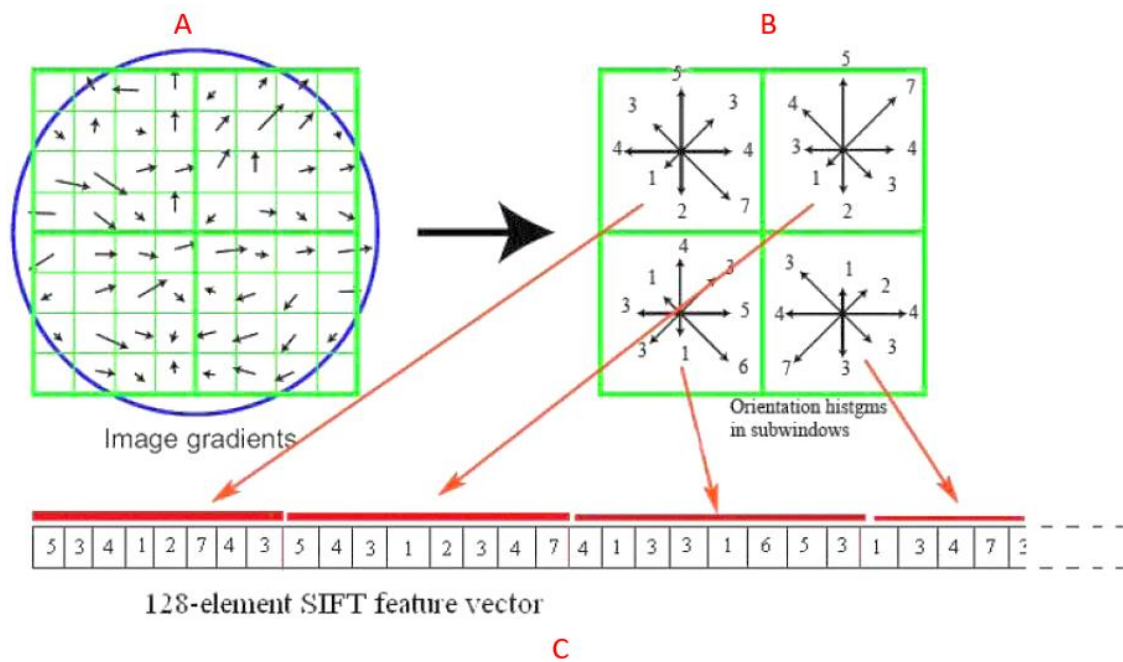


Figure 12: 128-element SIFT feature vector. We looking  $16 \times 16$  neighbourhood of the key point. but in this example, we compute relative orientation and magnitude in an  $8 \times 8$  neighbourhood plus divide it into 4 sub windows. (a) For each pixel the Gradient directions and gradients computed and then weighted. (b) Create a weighted orientation histogram that has 8 bin for each sub window (Step 5). (c) Practically we looking  $16 \times 16$  around the key point also dived it into  $4 \times 4$  sub windows. eventually, we have 16 histograms.  $128 = 16 \text{ histogram} \times 8 \text{ bin for each histogram}$ . (Estrada et al., 2004).

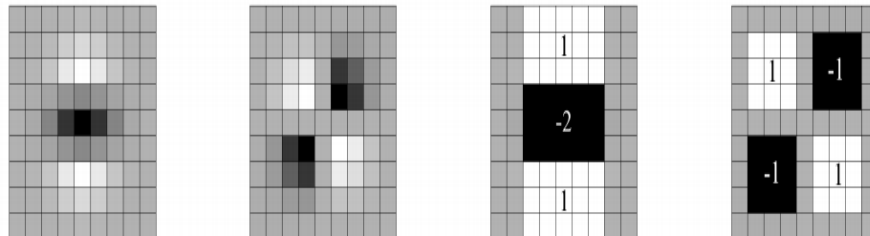
The calculation of Gaussian scale-space and the gradient histograms is intense and leads to SIFT being slow. The SURF algorithms uses an approximation which yields a less computationally intensive and hence faster algorithm, (see Figure 13 ).

The Bag of Features or Histogram of Features represents the distribution of features. Each visual word is represented by one bin. Bin frequency (y axis) represents the occurrence of a visual word in a specific image. Each image may be represented by a Histogram of Features which could be thought of as its fingerprint. A dictionary of visual

words can be created and test images can be compared to the visual words dictionary.

If the intersection is large, this means images are similar and vice versa.

### SURF: Speeded Up Robust Features



Left to right: the (discretised and cropped) Gaussian second order partial derivatives in  $y$ -direction and  $xy$ -direction, and our approximations thereof using box filters. The grey regions are equal to zero.

*Figure 13: SURF approximation.” The  $9 \times 9$  box filters in Fig. 1 are approximations for Gaussian second order derivatives with  $\sigma = 1.2$  and represent our lowest scale (i.e. highest spatial resolution). We denote our approximations by  $D_{xx}$ ,  $D_{yy}$ , and  $D_{xy}$ ” (Bay et al., 2006).*

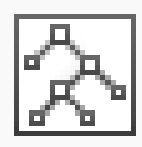
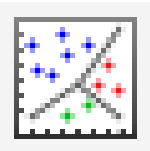
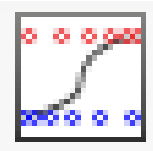
#### 3.9.4 Step 3 Train a Classifier and Verify.

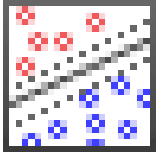
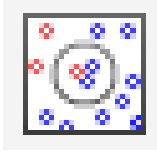
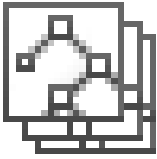
The Bag of Features and labelled data may be used to train the classifier. The classifier associates this set of features with a particular type of food. In classification there are two phases. The first phase is labelling data to known categories of food images. The second phase uses these data to learn to associate features with classes. Several classifier methods may be used to scale data and solve the classification problem. Examples of classifiers include the Support Vector Machine (SVM), decision trees, and Nearest Neighbour Classifier. After building the classifier it can be tested using data from outside the training dataset. For food image classification, the test data is food images

that have not been used to train the classifier. The label should be one of the food categories in the first step.

Table 4 provides some guidance on choosing a suitable classifier, given data characteristics and available software and hardware.

*Table 4: Characteristics of Classifier Types (Mathworks, 2016a).*

Classifier		Prediction Speed	Memory Usage	Interpretability
Decision Trees		Fast	Small	Easy
Discriminant Analysis		Fast	Small for linear, large for quadratic	Easy
Logistic Regression		Fast	Medium	Easy

Classifier	Prediction Speed	Memory Usage	Interpretability
Support Vector Machines 	Medium for linear. Slow for others.	Medium for linear. All others: medium for multiclass, large for binary.	Easy for Linear SVM. Hard for all other kernel types.
Nearest Neighbour Classifiers 	Slow for cubic. Medium for others.	Medium	Hard
Ensemble Classifiers 	Fast to medium depending on choice of algorithm	Low to high depending on choice of algorithm.	Hard

### 3.9.5 Step 4 Evaluate Classifier Performance.

The performance of a classifier can be evaluated using a Confusion Matrix. Figure 14 presents an example Confusion Matrix for a classifier trained to identify images of cars from specific countries. The rows are the actual class while the columns indicate predicted class. Anything on the diagonal (green) is correctly classified and off diagonals (red) are misclassified. In this example, all test images of Swedish and Italian cars are

misclassified while over 90% of American and Japanese cars are classified correctly. To improve classifier performance, the input data can be examined to find out why the model misclassified Swedish cars. Swedish, German cars and French cars are often confused and so more training data highlighting the differences between these makes could be added to the training dataset.

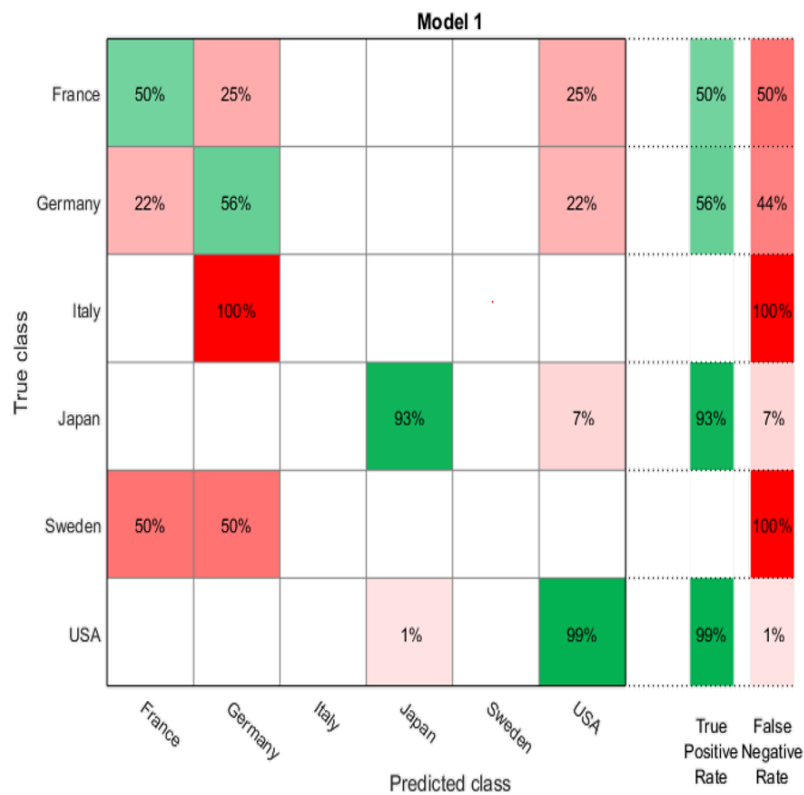


Figure 14: Confusion matrix for car classifier. Green indicates correct classification while red is misclassified (Mathworks, 2016b).

# Chapter 4 Cluster Analysis of collected Results

## 4.1 Overview

Rapid developments in technology makes collecting and sharing data easy and fast. Every day, people create an incredible amount of data such as picture phone calls and internet activities, (see Figure 15). In 2016, 1.1 trillion pictures were taken (Perret, 2016). A challenge today is to transform large quantities of data (known as "big data") into real value. Methods of learning from data (data mining) can be classified into two broad techniques: supervised and unsupervised. This chapter focus on clustering, a specific method of unsupervised learning.



Figure 15: shows the amount of data created online every 60 seconds (Wollaston,2013),  
 b. Shows the intersection of different disciplines (Sayad, 2017).

## 4.2 Supervised learning

Supervised learning involves the use of training data to predict specific classes such as food types. Supervised learning requires a large amount of training data that has already been classified. The algorithms learn from the training data, how to classify new data. The accuracy can be measured by comparing algorithm classifications with those from a trusted method, such as that used to develop the training data. The performance can be summarised and illustrated using a confusion matrix.

## 4.3 Unsupervised learning

Unsupervised learning is the discovery of patterns or structure in the data without pre-knowledge of the data. Unsupervised algorithms do not need training data and evaluate data indirectly or qualitatively. This chapter focuses on clustering as an example of unsupervised learning.



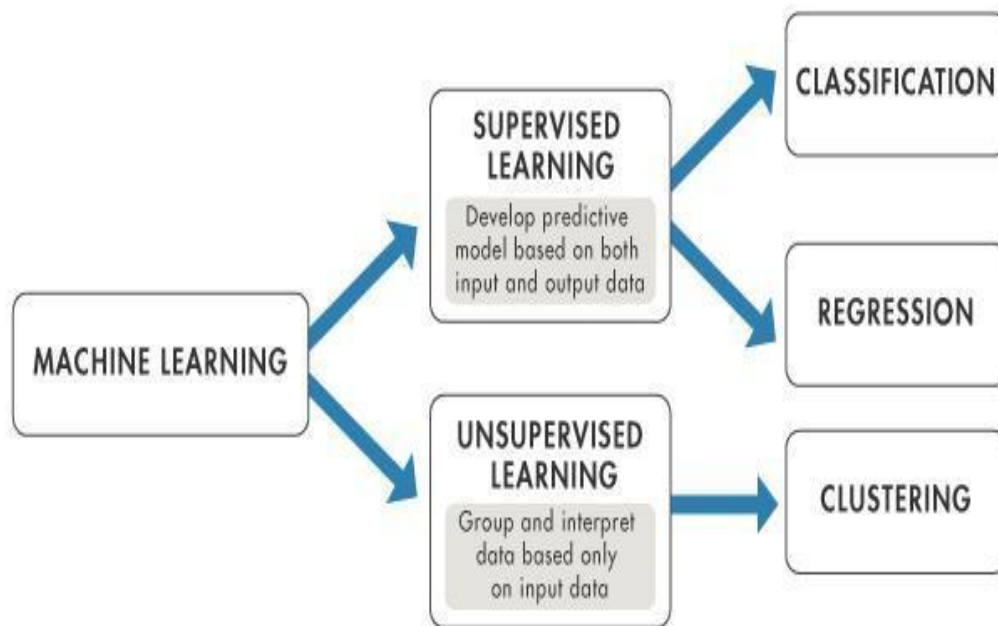


Figure 16: Comparing supervised learning vs unsupervised learning (Mathworks, n.d.).

### 4.3.1 Clustering

The purpose of clustering is to organise data into clusters with similar characteristics. Items within a cluster are more similar than items from different clusters. For example, a clustering algorithm could group pixels from an image that are similar in RGB colour values. Clustering usually does not have a unique solution as there is variation within each cluster and the number of clusters is generally unknown a priori. Increasing the number of clusters will generally reduce the variation within each cluster but ultimately leads to a solution where each item is its own cluster. A compromise needs to be selected.

Clustering is applied in a large number of fields, such as data mining, market research, pattern recognition, search engines and image processing. Clustering techniques can be categorised based on a large number of algorithm features. The paragraphs below describe some of the more important taxonomies.

### **4.3.2 Overlap**

Clustering algorithms can be divided into two classes

- Exclusive clustering, also called hard clustering, divides data into clusters such that every point belongs to a single cluster, i.e. there is no overlap. An example method is K-means clustering (see section 4.4 ).
- Overlapping or smoothing clustering. The data elements can belong to multiple clusters. For example, C-mean and fuzzy clustering are overlapping algorithms. The degree of belonging (strength of association) describes how much each data point is similar to its cluster characteristics.

### **4.3.3 Hierarchical or flat clustering**

Hierarchical clustering builds a Hierarchical Clustering Tree (HCT). At the lowest level, every data point is an individual cluster. Similar clusters are merged at each hierarchical level yielding a Dendrogram.

Flat clustering groups data points into clusters that contain points that are similar to each other and different from other clusters. K-means is an example of a flat clustering algorithm.

### **4.3.4 Goals**

The data features used for clustering can be monothetic or polythetic. In a monothetic cluster, all data points have a specific common feature. For example, in a specific image cluster, the pixels can have different locations, but all are red. Another example: customers in category A could have different ages but all of them buy computer games. In a polythetic cluster, data points are similar to each other in general but there is no

single property that is common to all cluster points. The similarity defines membership, not a particular attribute value.

#### 4.4 K-means

K-means is an iterative unsupervised form of clustering used to group various types of data, including in image analysis. Unlike other algorithms, the K-means algorithm does not estimate the number of clusters. The user should select a cluster number (K). K-means input data points should be numeric (not categorical) and are usually a set of vectors:  $X = \{x_1, x_2, x_3, \dots, x_n\}$ .

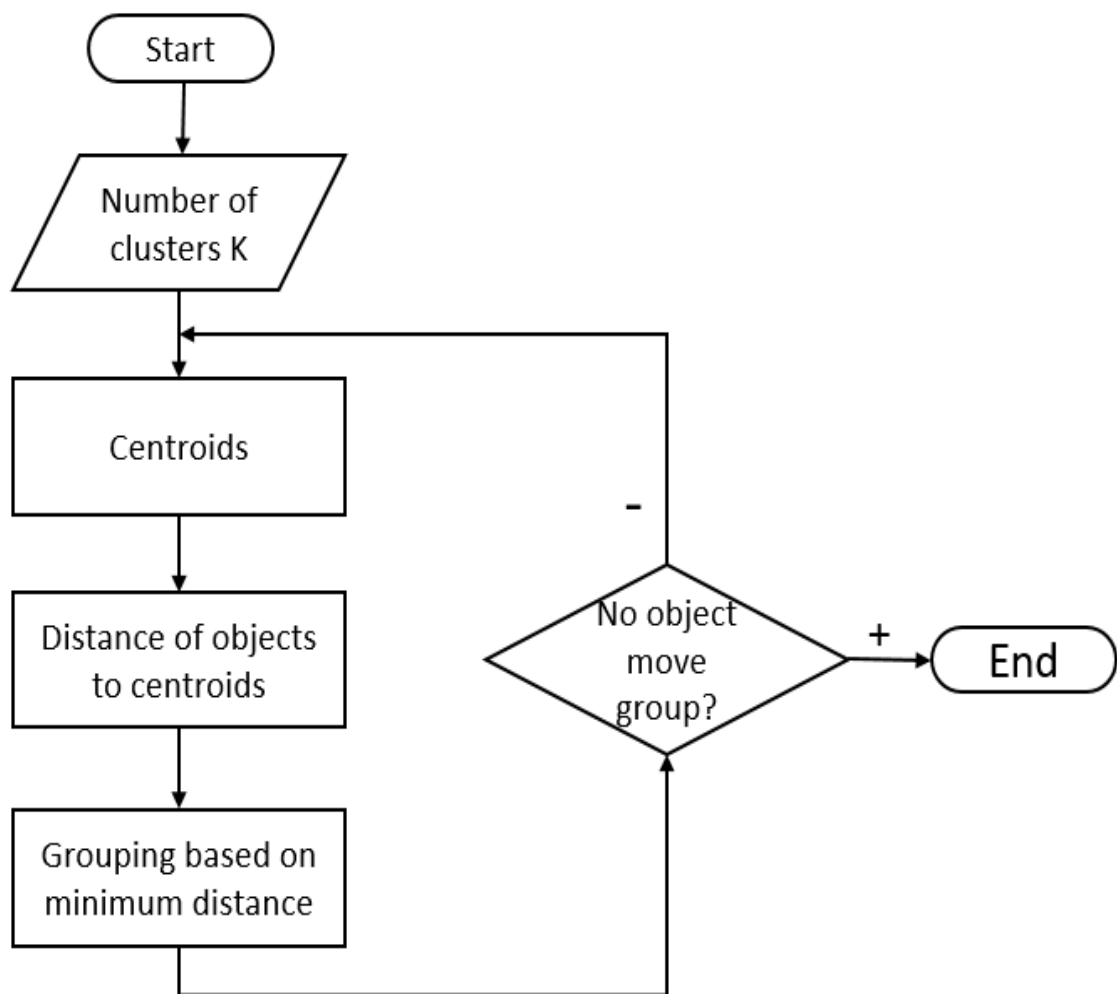


Figure 17: K-means algorithm steps.

### 4.4.1 Steps

#### Step one: initialization of centroids.

Initially, the algorithm chooses  $K$  points as cluster centres (called centroids). Each centroid represents a single cluster. The centroids should be in the same space as the  $n$ -dimensional data points. Centroids can be selected randomly or smartly.

Different seeding leads to different clustering results for the same data. "The careful seeding method of K-means++ avoids this problem altogether, and it almost always attains the optimal results on synthetic datasets. The difference between K-means and K-means++ on real-world datasets is also quite substantial. On the Cloud dataset, K-means++ terminates almost twice as fast while achieving potential function values about 20% better" (Arthur & Vassilvitskii, 2007). The authors claim that the K-means++ improves accuracy and speed. K-means++ follows the steps below to select centroids.

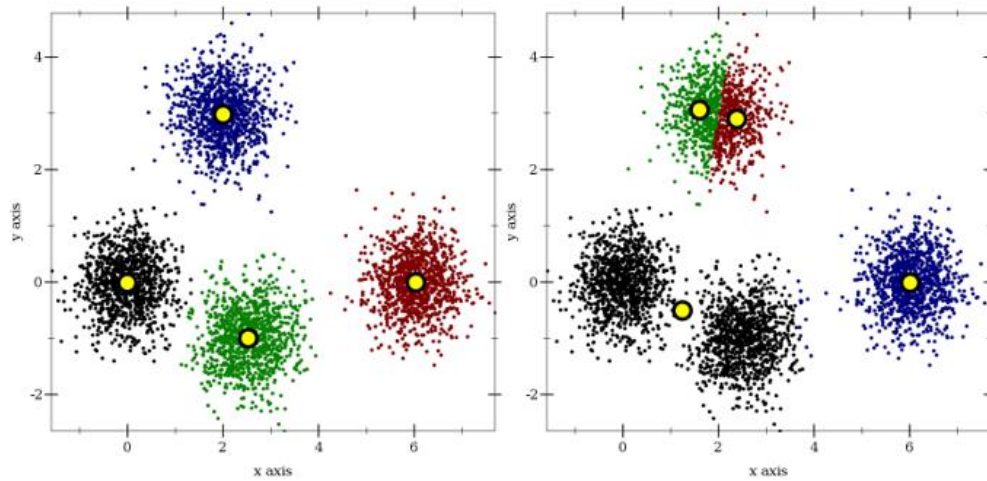
1-choose the first centroid randomly.

2- Compute distances  $D(\mathbf{x})$  from each data point  $\mathbf{x}$  to the first cluster centre (C1) which is the only centroid i.e. the distance between the first centroid and all data points.

3-Choose the next centroid to be one of the data points with a probability of  $\mathbf{x}$  being chosen proportional to  $D(\mathbf{x})^2$ . The cluster centre chosen is likely to be distant from the current centroid. Figure 18 shows how to select the next cluster centre using K-means++ and compares the results with random seeding.

4- Re compute distances between each data point  $\mathbf{x}$  and the nearest cluster centre (the first and second centres). Choose the next centroid as in the previous step.

5- Repeat steps 2 to 4 until all  $k$  centroids have been selected  $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \dots, \mathbf{c}_k\}$



Initializing by K-means++

Initializing by random choice

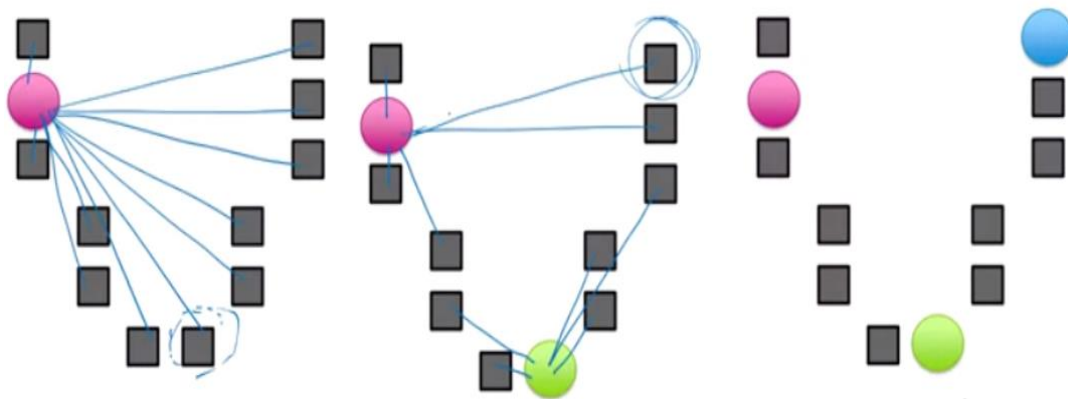


Figure 18: Top figure: Comparison of random initialisation and K-means. Random initialisation may give low-quality results. Bottom figure: Shows seeding progress using K-means++ ( steps 1to 4)[ Rosettacode,2017].

**Step two: Calculate distances.**

K-means clustering is based on similarity, which is measured by a distance between elements. Each datum is assigned to a cluster based on the closest centroid. Distance and similarity reflect the degree of closeness or separation between data points, and a wide variety of distance norms are used with the most common being the Squared Euclidean norm. This is computationally efficient and most appropriate when all datum

elements have the same units. If the dimensions are distance then the Squared Euclidean norm is the squared distance between the datum  $\mathbf{X}$  and the centroid  $\mathbf{C}$ .

$$D(\mathbf{X}, \mathbf{C}) \equiv \|\mathbf{X} - \mathbf{C}\|_2^2 = (\mathbf{X} - \mathbf{C})^T (\mathbf{X} - \mathbf{C}) \quad (4.1)$$

The city block or Manhattan norm is the distance between two points assuming it is only possible to move along variable directions:

$$d(\mathbf{X}, \mathbf{C}) \equiv \|\mathbf{X} - \mathbf{C}\|_1 = \sum_{i=1}^n |\mathbf{X}_i - \mathbf{C}_i| \quad (4.2)$$

The cosine distance is one minus the cosine of the angle subtended by two vectors at the origin.

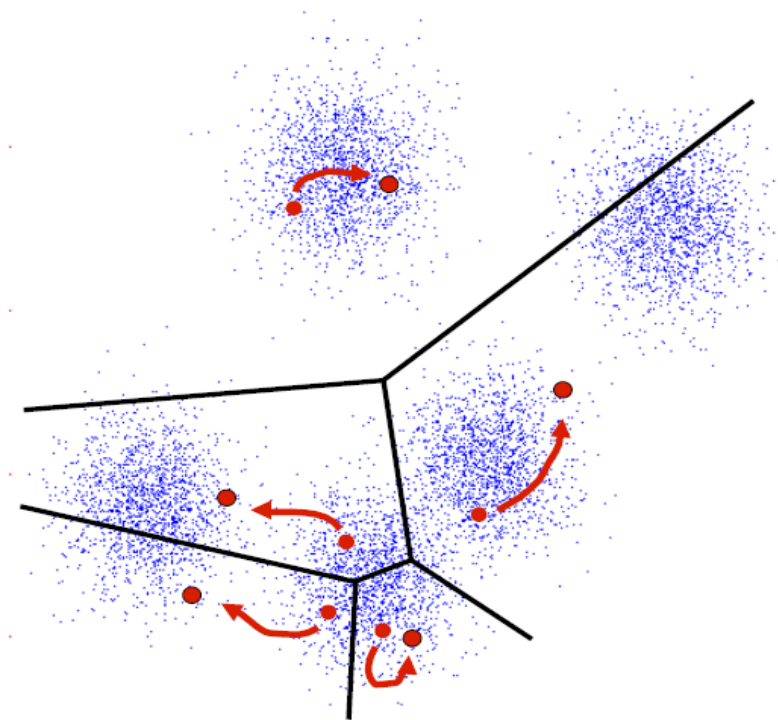
$$D(\mathbf{X}, \mathbf{C}) \equiv 1 - \frac{\mathbf{X} \cdot \mathbf{C}}{\sqrt{(\mathbf{X} \cdot \mathbf{X})(\mathbf{C} \cdot \mathbf{C})}} \quad (4.3)$$

Step three: Compute new centroids.

The algorithm iteratively calculates cluster centroids and assigns data to clusters. The new centroid is the average of the points assigned to a cluster:

$$\mathbf{C}_i = \frac{1}{|S_i|} \sum_{\mathbf{x}_j \in S_i} \mathbf{x}_j \quad (4.4)$$

where  $S_i$  is the set of data assigned to the  $i$ th cluster.



*Figure 19: Compute new centroids (Padhraic, 2015).*

#### **Step four: Update Centroids.**

Data points are re-assigned to the nearest new centroid cluster. The algorithm iteratively updates centroids, by repeating steps 2 to 4, until centroid values become stable and do not change.

K-means clustering is less computationally expensive than other clustering algorithms. Several factors affect the computation required: the number and dimension of data, the similarity norm used, the number of clusters and the number of iterations.

#### **4.4.2 Clusters number (K value):**

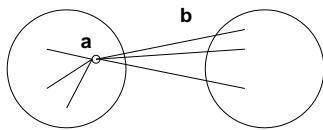
One disadvantage of K-means is that the number of clusters is selected manually by the user before starting clustering. This section explains three approaches to automatically

estimate the number of clusters (K value). The approaches are Silhouette, Gap and Davies Bouldin.

### Silhouette

Silhouette is a method used to measure cohesion and separation for a data point in its own cluster and neighbouring cluster, as shown in the following equation. "Each cluster is represented by a so-called silhouette, which is based on the comparison of its tightness and separation. This silhouette shows which objects lie well within their cluster, and which ones are merely somewhere in between clusters" (Rousseeuw, 1987).

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (4.5)$$



where:

For an individual point,  $i$

$s_i$  = silhouette coefficient.

$a_i$  = the average distance of  $i$  to the points in its cluster.

$b_i$  = min (average distance of  $i$  to points in another cluster).

The silhouette coefficient yields values ranging from -1 to 1. A high silhouette value (1) indicates that the data point is well connected to the cluster and at the same time not well connected to other clusters. If the silhouette coefficient is negative it means that



the data point does not clearly belong to the cluster. The optimal k value is the one that yields the maximum element silhouette value.

### The Gap method

The gap method is used to estimate the cluster number by plotting an error measurement versus suggested k value, (see Figure 20). The curve looks like an arm and the point where the distortion (gap value) changes dramatically is called the elbow. As is clear in Figure 20, a dramatic change occurs when the k value is two. Therefore, the optimal k value is two. The following formula is used to calculate gap value.

$$Gapn(k) = E_n^*\{\log(Wk)\} - \log(Wk) \quad (4.6)$$

where

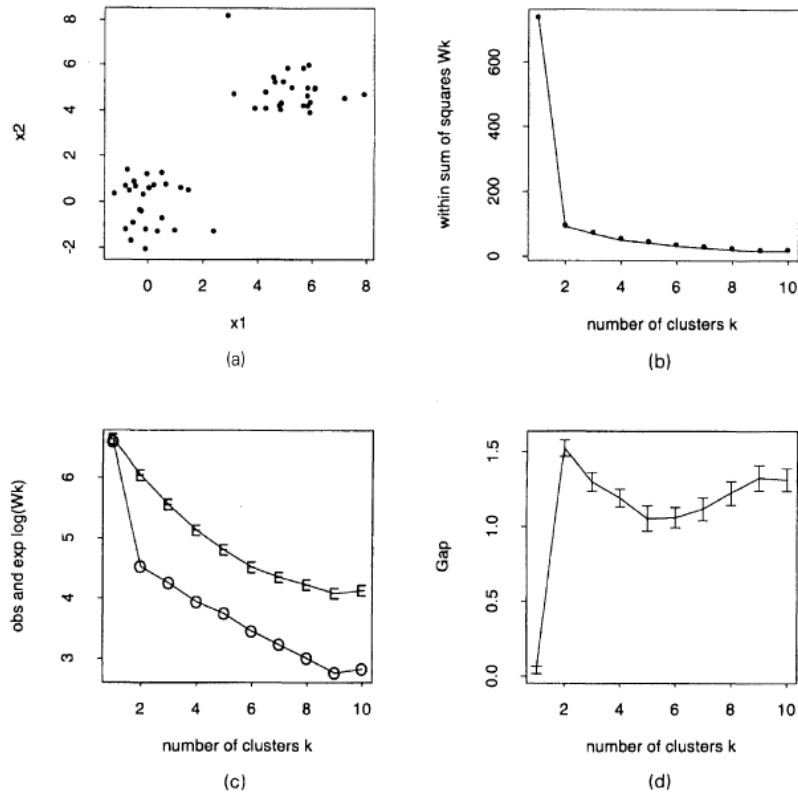
$n$  is the sample size.

$k$  is the suggested k value numbers.

$Wk$  is the total within-cluster variation.

$$Wk = \sum_{r=1}^k \frac{1}{2n_r} Dr \quad (4.7)$$

where  $n_r$  is the number of data points in cluster  $r$ , and  $Dr$  is the sum of the pairwise distances for all points in cluster  $r$ . "The expected value  $E_n^*\{\log(Wk)\}$  is determined by Monte Carlo sampling from a reference distribution, and  $\log(Wk)$  is computed from the sample data" (Mathworks, 2017)



Results for the two-cluster example: (a) data; (b) within sum of squares function  $W_k$ ; (c) functions  $\log(W_k)$  (O) and  $E_n^*(\log(W_k))$  (E); (d) gap curve

Figure 20: Gap method example

Figure 21 illustrates average gap values versus proposed numbers of clusters for a number of images collected from Thwaite Hall (student accommodation). For these images, there are no clear elbows, but the best  $k$  value would need to be selected manually. However, this is neither an accurate or feasible method to select cluster number for food images.

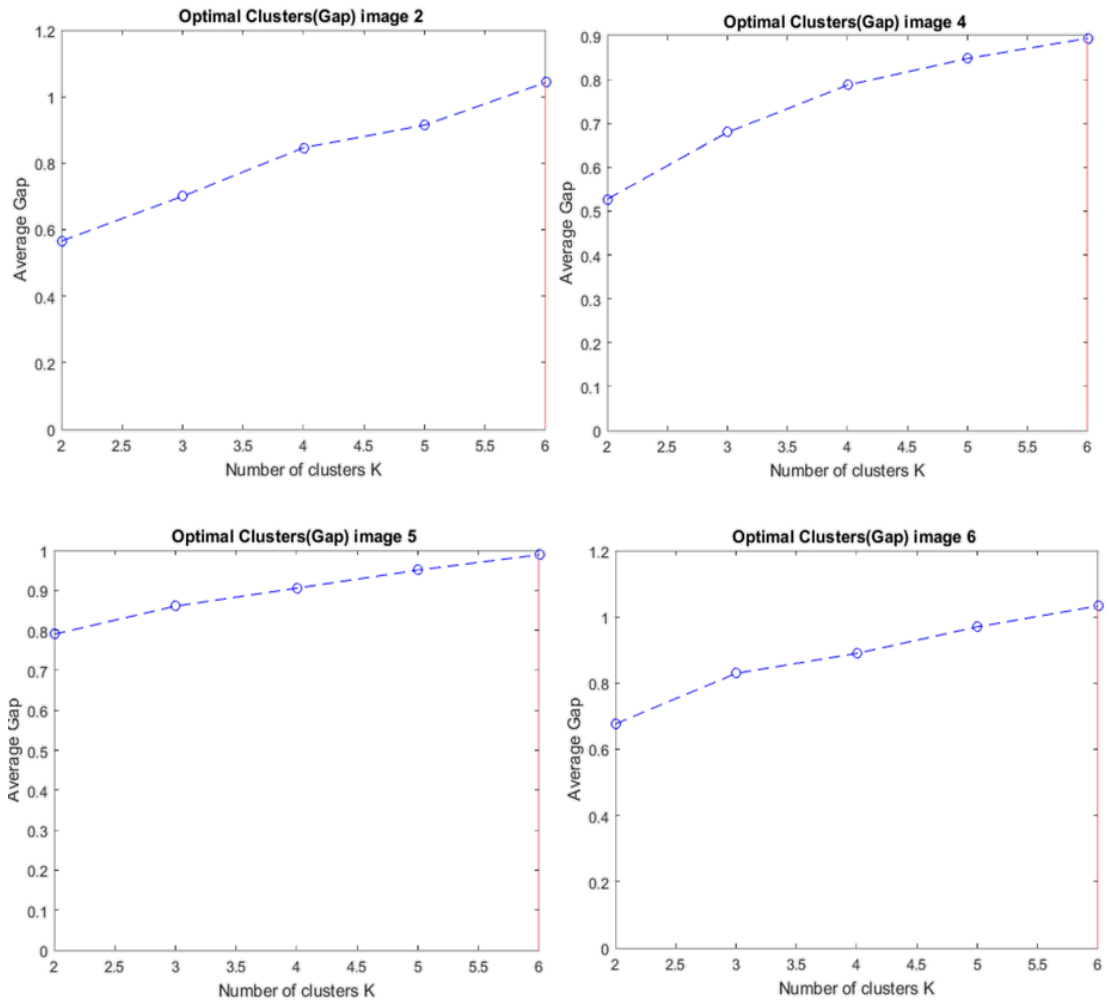


Figure 21: Example of gap method failing to estimate K value.

### Davies Bouldin Criterion

This is a mathematical measure used to evaluate the optimal k value by finding the similarity between clusters with an expected data density. The measure is independent of cluster number and method of partition.

The Davies Bouldin measure compares the spread within clusters to the distance between cluster centres:

$$DB = \frac{1}{K} \sum_{i=1}^k \max_{j \neq i} \{D_{i,j}\} \quad (4.8)$$

where

DB is Davies-Bouldin Criterion.

$D_{i,j}$  is defined:

$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}} \quad (4.9)$$

where

$\bar{d}_i$  is the average distance of data points for the  $i$ th cluster.

$\bar{d}_j$  is the average distance of data points for the  $j$ th cluster.

$d_{i,j}$  represents the Euclidean distance between the central points of the  $i$ th and  $j$ th clusters.

Figure 22 illustrates the variation of the DB coefficient and shows that  $k=3$  yields the smallest value.

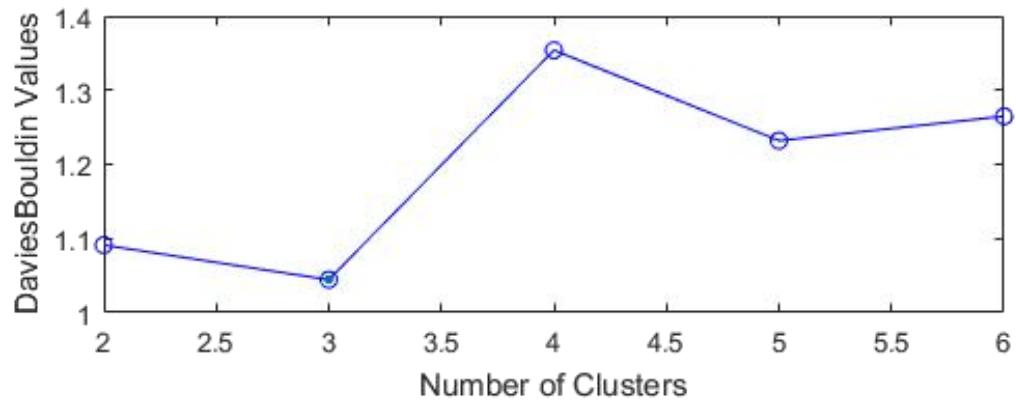


Figure 22: Optimal cluster number selection using the Davies Bouldin algorithm.

## 4.5 Conclusions

Chapter 4 has reviewed the taxonomy of clustering techniques and looked at K-means clustering in detail. In the next chapter, these algorithms will be applied to the problem of segmenting images of plates of food into individual food items.

# Chapter 5 Results of initial experiments on food images

## 5.1 Introduction

Chapter Five describes two numerical experiments using different approaches to segment and identify food in the image. The first experiment tests a method based on thresholding measures of colour and texture. The second experiment uses Machine Learning to identify the food meal in the image without segmenting food types. The conclusions drawn from these experiments have informed the development of the final proposed algorithm and system; presented in Chapters six and seven.

### 5.1.1 Datasets used in this project

The initial plans of the project was to collect data from one of the hull hospital institutions. Unfortunately, we did not succeed to have the authority to collect data from hospitals (patient privacy constraints and official agreements with university), therefore we decided to collect data for initial experiments from institution providing food services similar to hospitals.

Three databases of food images have been used in this project at various stages. These were collected from two student halls of residence in Hull: Thwaite Hall and Portland Street. Thwaite Hall is operated by the University of Hull and Portland St is privately run. A third dataset was acquired within the School of Engineering, University of Hull. The foods considered were chosen to be similar (as much as possible) to hospital or UK institutional food.

#### **5-1.1.1 The Portland database (2016 mid year)**

During the Thesis Advisory Panel (TAP) meeting, a machine learning (ML) approach was proposed by the researcher to tackle the project challenges. Initially the project used various images from public databases or acquired locally in the Engineering department. For the next stage, it was agreed to collect data (food images) from the a private restaurant institution to test the suitability of ML approaches to identify different meals. Private student accommodation called “the Portland” was suggested as the location to collect the data, as it has a large and suitable dining area to provide food for the students. In particular, this dataset was principally used to test the sensitivity of the ML classifier to food illumination and different scenarios of food arrangement.

Hospitals and care homes are expected to have a list of fixed meals, and that the user or patient chooses their preferred meal from a meal list. Each meal in the list contains fixed contents and the patient has no opportunity to change it. For example, Meal 1 contains salad, chicken and peas, so each time the patient chooses this meal, it contains the same food types.

The panel recommended collecting images of 40 different meals. The meals were labelled from 1 to 40, and these labels were visible in each image. For each meal (from

1 to 40), images were collected in four different light conditions: near to the window, in the middle of the room, in the corner, or directly under a lamp. This was done to test if the machine learning could identify the same meal in different light conditions.

Additionally, for each light condition (from 1 to 4), the food was placed in five different arrangements, in order to test if the ML algorithm performance over different food arrangements. For example, the images were collected in Light Condition 1, the food rearranged, and another image captured. This was repeated to give five images for each food, giving a total number of 800 images. The images were collected using a 24-megapixel digital camera and a 13-megapixel camera in a mobile phone

The results showed that the ML algorithm successfully identified the 40 meals in various light conditions and various arrangements. The experiment details are explained in section 5.4.1. However, The results also showed that food identification was inadequate as a means of calculating food nutrition values, because each type of food has different nutritional values, and each patient eats only their preferred food type and leaves the rest. For example, if the meal contains salad, chicken and rice, and a patient does not like rice, he will not eat the rice. In this case, it is difficult to identify the food eaten without segmenting and extracting each type of food in a separate image. Therefore, the next development step was to build and test segmentation approaches before identification; this method was later developed and tested, as shown in Chapter 6. The segmentation stage allowed calculating the different nutritional values for each type of food separately.



### **5-1.1.2 The Thwaite Hall database (Year end 2016 and mid year 2017)**

As the types of food provided in the Portland are not very similar to hospital food or UK institutional food, other locations were discussed, to collect a new dataset. Additionally, image of food individually taken was needed. These locations were Castle Hill Hospital in Hull, the Staff House dining area in the university, and a number of student accommodations in Hull.

However, access to Castle Hill Hospital was not available. The Staff House dining area in the university was closed for renovation, and there was no clear expected date for its reopening. The third location options were student accommodations under university management. We started to contact the student accommodations, and Thwaite Hall was the only response received. The location was visited and images collected.

The data details and the results of the experiment are shown in section 5.4.2. Two types of images were collected: the first type were images of each type of food, to train the classifier; for example fish, chips and peas individually taken. The total number of images of this type was 974 ( with the same protocol of multiple luminosity conditions) . The second type of images were meal images, used to test the late algorithm stages; these images contained more than one type of food. Meal images were used to test food segmentation and identification, but not to train the classifier; the total number was 105. The total number of images collected from this location was 1097. The images were collected at a various viewing angles and distances, using again a 24-megapixel digital camera and a 13-megapixel camera in a mobile phone. Results showed that the algorithm was able to successfully segment, identify and estimate food nutritional values for each food type in the meal images (for more details, see section 5.4.2 ).

### **5-1.1.3 The School of Engineering database ( 2018 earlier year )**

At this of the project, we had working algorithm to identify food present in meals and wanted to compare after and before eating food. The last part of the algorithm was intended to estimate the food eaten by comparing food images before and after eating. The algorithm needed to be tested when patients had eaten for instance a quarter or a half of the food. Unfortunately, we found that the students (in Thwaite Hall) tended to eat all the food they bought. Therefore, the food images after eating usually had no food remaining. In addition, the food list in Thwaite Hall was changed to a fast-food list, to meet student demand, and the previous food list was no longer available. The issue was discussed with the supervisors, and the decision was made to collect food images in the School of Engineering, in order to, at least, test the last part of the algorithm ( which was to estimate food eaten by comparing food images before and after eating).

The data were collected to test the algorithm's ability to identify different scenarios of remaining food: if the patient ate a quarter, a half or all of the food. The data were collected by the researcher in the similar way as previously in Thwaite Hall, and with a decision to choose the same food types as before. There were two types of images: a single food type (1775 images) see Table 8, and meal type (189 images). Single food images were intended to improve the classifier performance by increasing the of the dataset size used for the training. The meal images were images used to test the food eaten by comparing food images before and after eating. The images before eating try to represent food provided by the hospital to the patient, and the images after eating try to represent food images after the patient had finished eating. The total number of images was 1964.

The collection method and the list of food types were the same in Thwaite Hall and the School of Engineering ( changes on luminosity and food permutations). Note that the use of machine learning is consistent with mixing images from different source as it is common in the literature. "We start with collecting images localized to a particular restaurant R. Once we know R, we can use the web as a knowledge-base and search for R's menu. This task is greatly simplified thanks to online data-sources like Yelp, Google Places, Allmenus.com and Openmenu.com, which provides comprehensive databases of restaurant menus. Let the menu for R be denoted by  $M_R$  and let the items on the menu be  $m_i$ . For each  $m_i \in M_R$ , the top 50 images of  $m_i$  are downloaded using search engines like Google Image search. This comprises the weakly-labeled training data" (Bettadapura et al., 2015: 6). Food-101 is a food recognition algorithm created using a dataset from various clients (Bossard et al., 2014). The dataset (101000 images) used to train the classifier build from a website called OpenTable ([www.foodspotting.com](http://www.foodspotting.com)). The website allows users to upload their images to form where they are using their cameras (see section 2.5 ). The food log is an online service allow clients uploading their food image to estimate food balance (Aizawa et al., 2013). The uploaded images use to update and train the classifier. The dataset consists of images collected in different locations and conditions (See section 2.5).

Therefore, the two sets of single food images were combined in order to increase the training data and improve the classifier performance. The total number of images resulting from combining the two sets was 2749 (974 Thwaite Hall + 1775 School of Engineering) (see Table 8). The second type of collected images were meal images to test the performance of the classifier in estimating food eaten, by comparing food

images before and after eating. More details about how the data were used to train the classifier and test the algorithm are explained in Chapter 6.

## 5.2 The First Experiment

The experiment aimed to develop a method to segment and identify food by applying thresholding techniques to measures of colour and texture. If two objects are similar in colour (like apple and fine beans) then the colour mask is not enough to segment the two objects correctly. Therefore, three different masks were developed: a texture mask, a colour mask and an intensity mask (see Figure 23). To reduce the number of variables considered, the food images were acquired in a systematic way that ensured images were taken from the same distance, at the same viewing angle and with the same lighting. A stand was used to hold the webcam and the light source 50 cm above the plate. The three steps in the algorithms may be summarised:

**Step one:** Image preparation by converting the original image to grayscale, HSV and LAB colour spaces.

**Step two:** Segmenting the image using colour, texture, and intensity masks.

**Step three:** Display results and extract region features such as object's area, see Figure

23

### 5.2.1 Step one: Image preparation.

The system receives the image from the webcam in RGB format, then converts it to the LAB, HSV, and grayscale. The colour and light intensity values are separated in the LAB and HSV colour spaces, allowing variation in light intensity to be removed from consideration. Further analysis uses the channels (A, B, H, and S) excluding light channels

(L and V). Furthermore, the image is converted from RGB to greyscale, which is used to extract texture features, such as entropy, and to create the intensity mask.

The human eye is not equally sensitive to all colours. Green, red and blue at the same intensity are perceived to have decreasing brightness. When converting from RGB to grayscale image, each channel has a specific weight. The standard NTSC conversion formula, presented below, was used to assign specific weights to each channel.

$$\text{intensity (grayscale image)} = 0.2989 * \text{red} + 0.5870 * \text{green} + 0.1140 * \text{blue} \quad (5.1)$$

(Mathworks, 2010)

The binary mask is created by adding the extracted features from the three masks. It extracts the targeted object and deletes the rest of the image. For example, the apple mask extracts apple only and deletes the rest of the image.

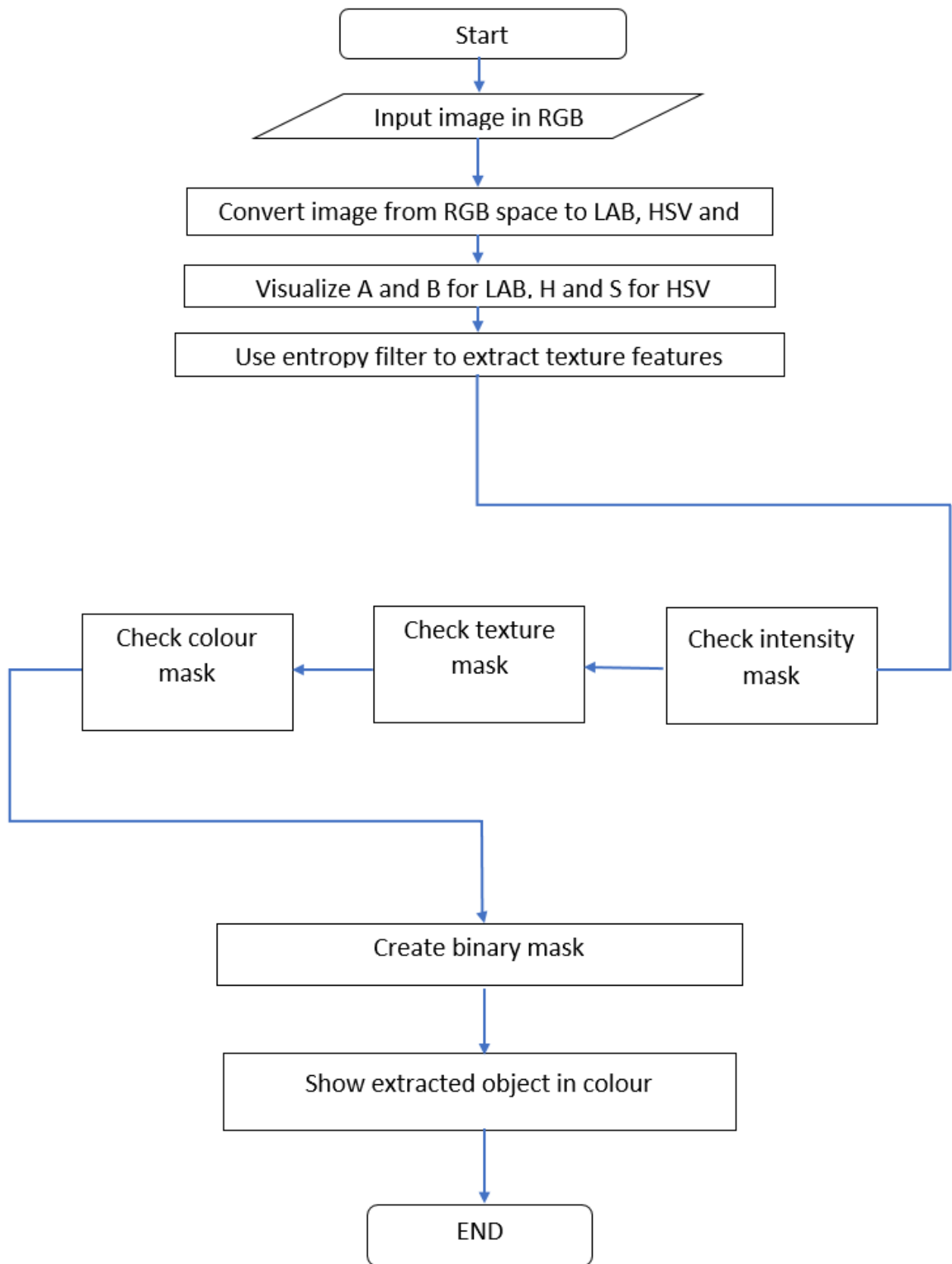


Figure 23: Flowchart describing the algorithm steps

### 5.2.2 Step two: Segment objects using the three masks.

To segment and identify objects, the system uses a multi-mask technique. There are three masks used together to filter and segment objects. Using three masks improves the accuracy and system performance.

#### Mask One: Intensity Mask.

The first level of information comes from the intensity in the grayscale image. Each food type yields a range of greyscale intensities, from very white foods, such as eggs which yield greyscale values around 250, through to foods that are reflective, such as apples which yield values between 160 to 250, to dark objects like cookies with intensities around 70. Typically, many foods yield similar intensity values and this characteristic alone is not adequate for distinguishing all food types.

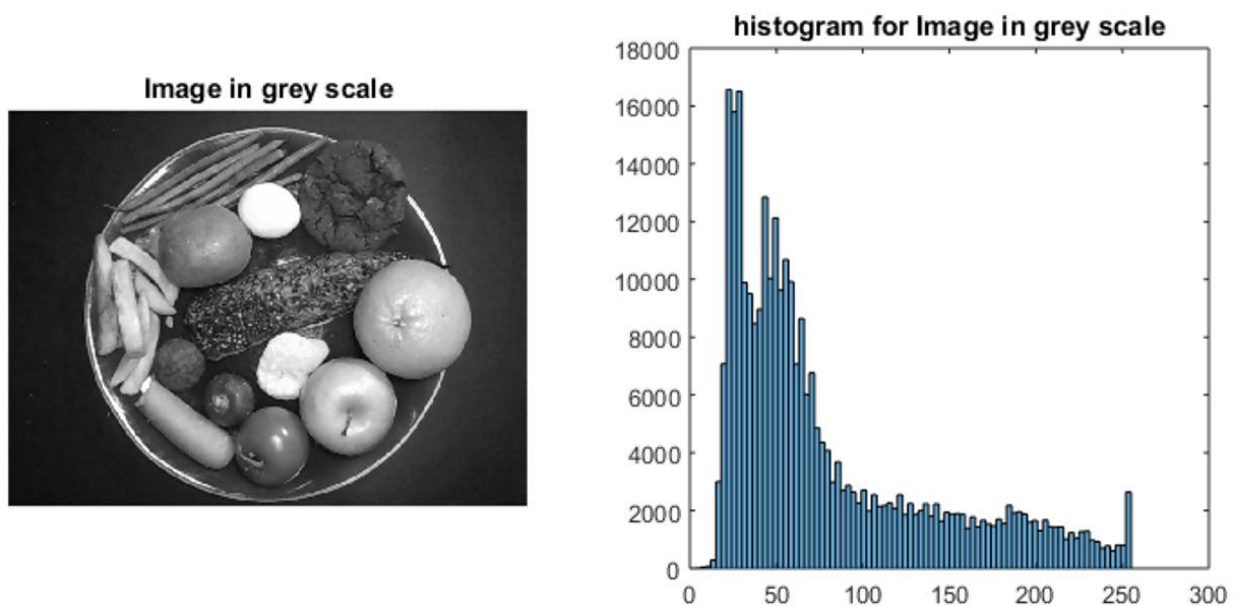


Figure 24: Original image in greyscale and histogram. The image has 13 different types of food.

## **Mask Two: Texture Mask.**

The texture of an image fragment describes the local variation in pixel colour intensity values. It is a complex property that cannot be summarised in a single number. Texture is determined by the distribution of pixel values, but also the covariance of values on different pixels and higher order statistics. For a region of an image to have an identifiable texture, these statistics need to be uniform over an area larger than the correlation length. Human perception is very sensitive to texture, but it is difficult to automatically classify all the potential variation.

Texture features help to detect objects and segment them from the image. There are many measures of texture, including the entropy, range and standard deviation of pixel colour intensity values. The range and standard deviation help distinguish rough objects from those that are smooth.

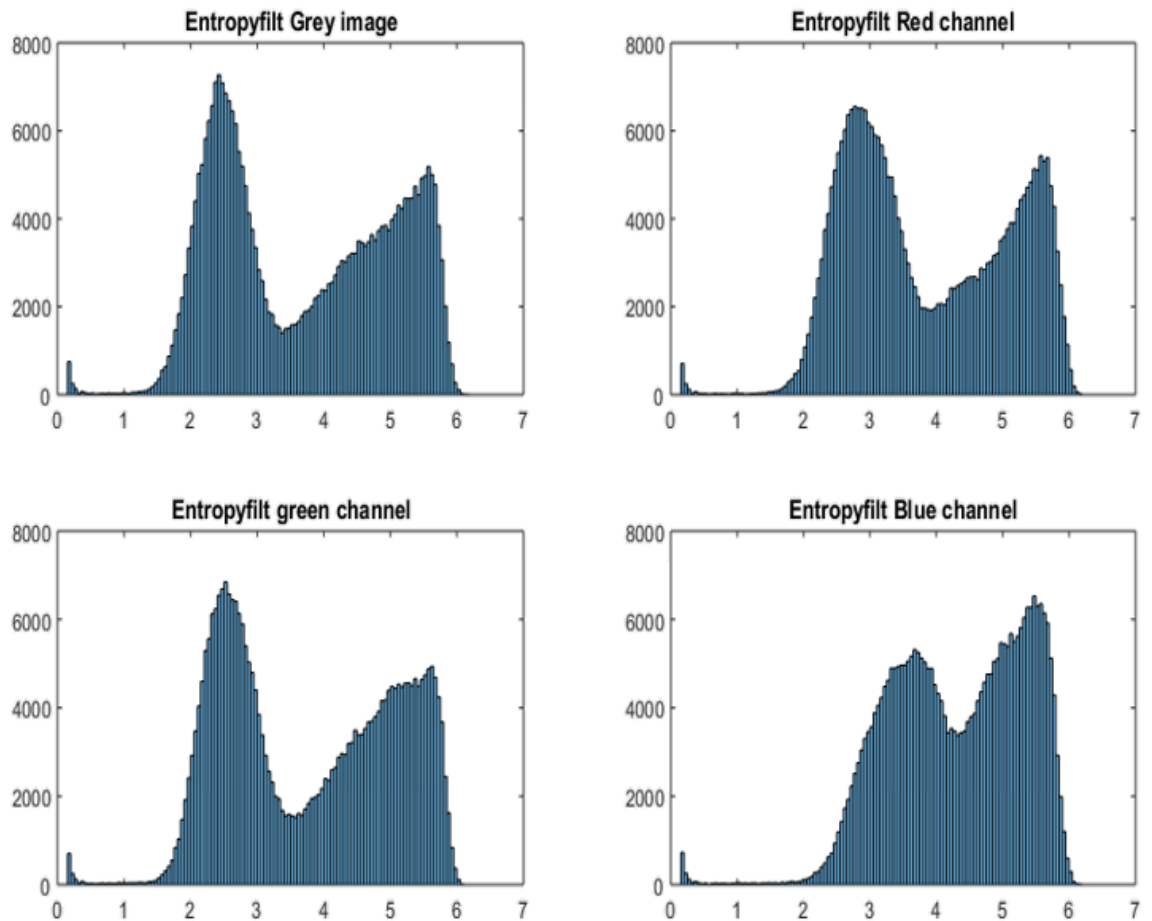
The following sections provide details and examples of texture filtering applied to grayscale images or single channel colour images.

### **Entropy filter:**

Entropy is a term from statistical physics that describes the amount of disorder in a system. A highly ordered system, like a sugar crystal, has low entropy, while a disordered arrangement, such as sugar dissolved in a liquid, has high entropy. For an image, the entropy is calculated from the histogram of pixel values in a region. If only a few pixel values occur, then the image is highly ordered; but if all pixel values are equally likely then the image is disordered and the entropy is high. Typically, entropy is calculated over a small window that is moved over the image producing a new image of the texture measure. This can be used in image segmentation based on texture.



The entropy filter can be applied to grayscale, or single colour channel. Figure 25 shows the entropy histograms of different channels in the food image in Figure 24. The entropy values, range from 0 to 7. The wide range of the entropy histograms helps to segment objects.



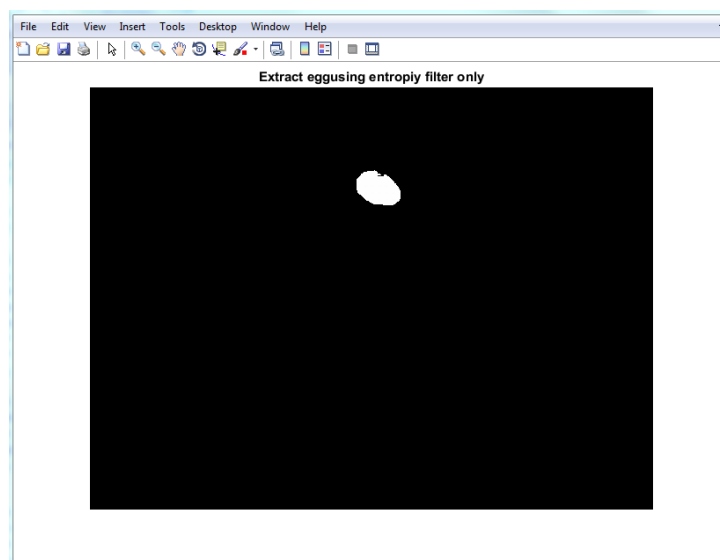
*Figure 25: Entropy histograms for four channels. Top left for grayscale; top right the red channel; bottom left; the green channel; bottom right; the blue channel.*

### Example 1: Identifying an egg using an entropy filter

Different foods have different textures and hence entropy values. Objects with regular surfaces and uniform colours have low entropy, for example a boiled egg. The entropy value for an egg ranges from a minimum of 0 to a maximum of 3. Another objects, such as fish skin, are rough and may have a colour pattern. These objects have a high entropy value. For example, the skin of smoked mackerel yields an entropy value between 5.4 and 7. Ideally, the texture statistic should be independent of lighting variations.

It is clear from the histograms in Figure 25 that there is an object that has a low entropy value in the range 0 to 1. This means that the object is smooth and has a surface with few details. The object is an egg and can be extracted based on texture segmentation only, as shown in Figure 26. The Boolean equation below yields true (1) when the entropy equals 0 or 1 and false (0) otherwise. Figure 26 is formed using the Boolean entropy to mask the original colour image.

```
egg_mask= (E >= 0) & (E <=1);
```

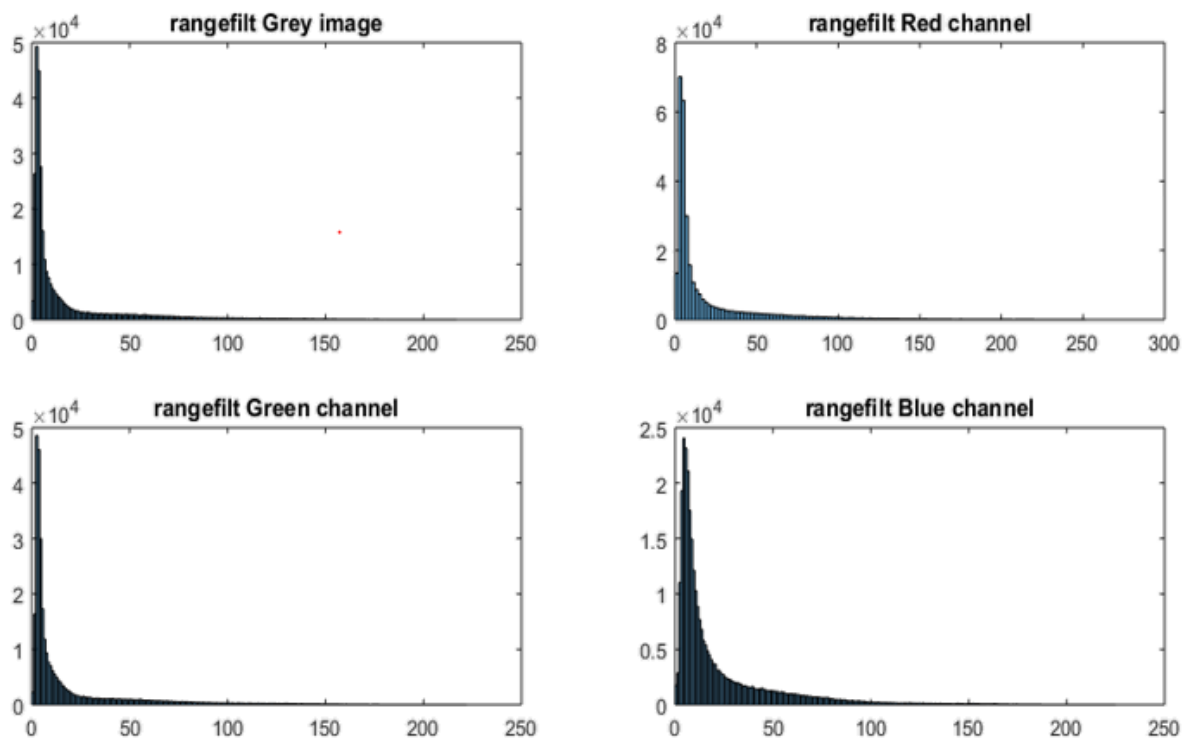


*Figure 26: Egg identified using entropy filter.*

**Range and standard deviation filters:**

Range and standard deviation filters are also used for image texture analysis. The filters measure the amount of variation in pixel intensity over small regions. The range and standard deviation both measure the variation of pixel values around the mean value.

Figure 27 shows histograms of local range and Figure 28 shows the standard deviation for the original image in Figure 24.



*Figure 27: Histograms of local ranges for grayscale and RGB channels.*

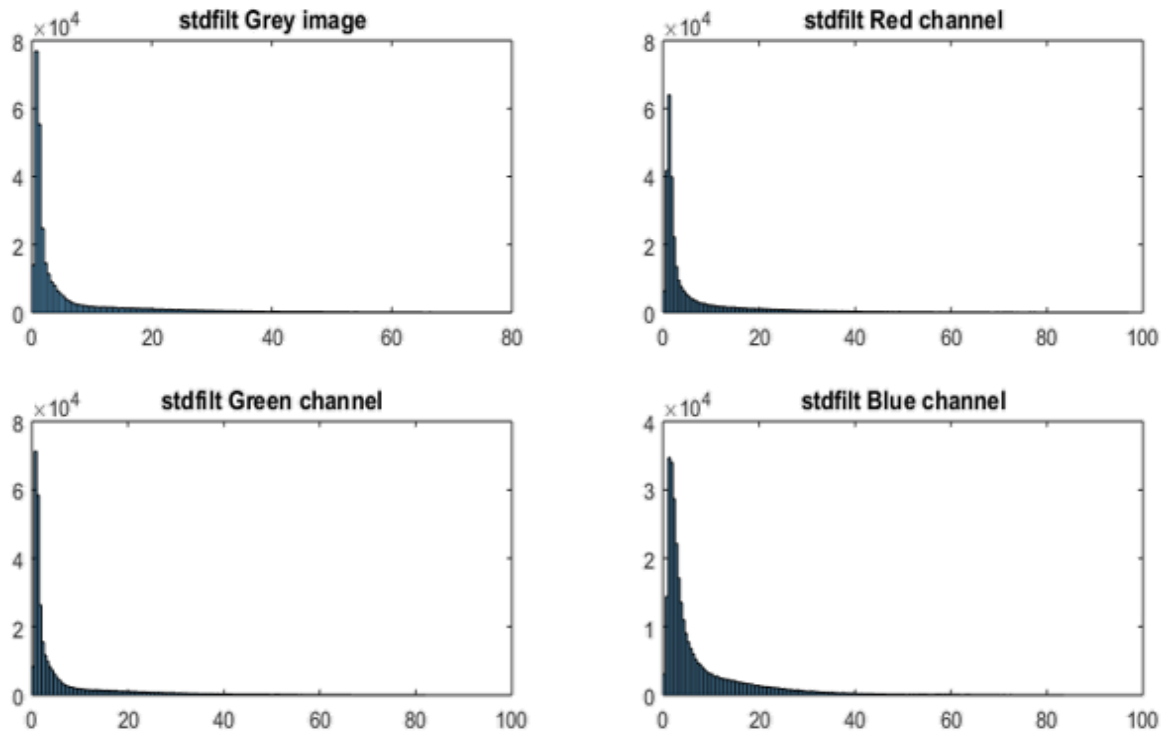


Figure 28: Histograms of local standard deviations for grayscale and RGB channels.

As shown in Figure 27 and Figure 28, the histograms for range and the standard deviation have only low values, suggesting that the entropy measure in Figure 25 is better at distinguishing food objects.

### **Mast Three: Colour Mask.**

The four channels  $A$ ,  $B$ ,  $H$ , and  $S$ ; are theoretically independent of light variation. Figure 29 shows the histogram for  $A$  and  $B$  values for the original image in Figure 24. For the LAB colour space, the  $A$  channel values are from -40 to 50 and the  $B$  values are from -70 to 80.

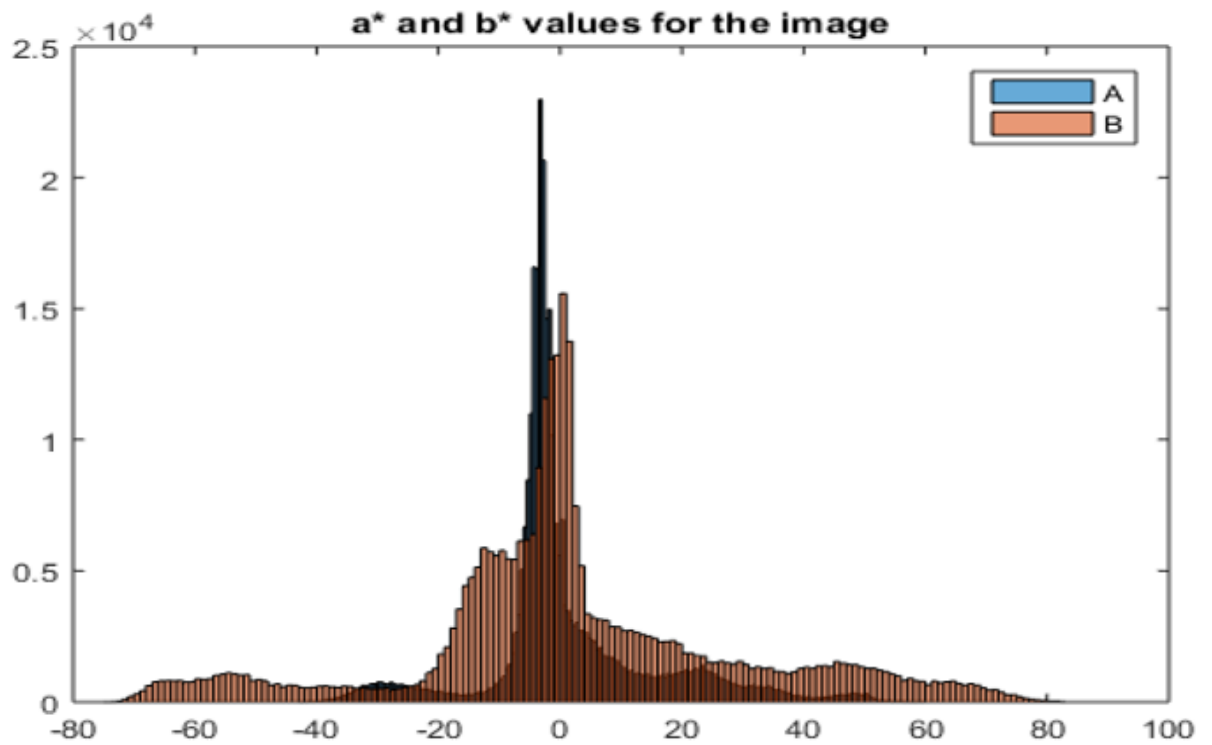


Figure 29: Histogram for A and B values for the image in figure (Figure 24)

To extract a specific type of food, the range of A and B values for the food concerned is used to produce a colour mask largely independent of illumination intensity.

The following example defines a LAB colour mask for tomato. A region of interest around the tomato is defined and the A and B values extracted. A histogram of A and B values for the tomato is shown in Figure 30.

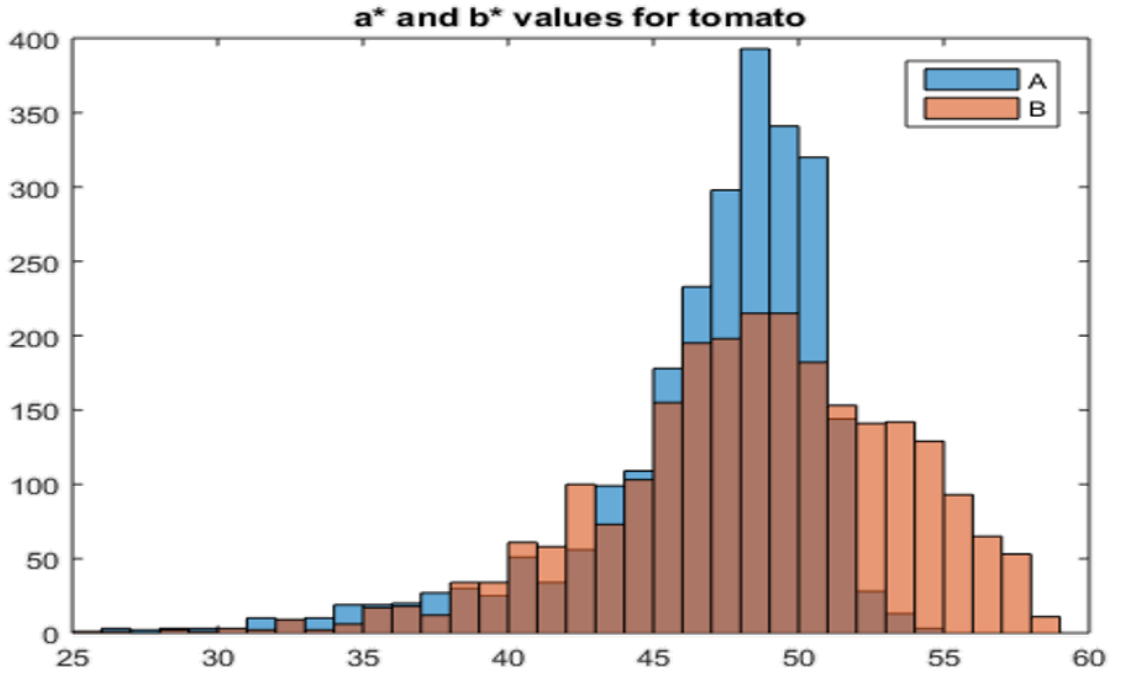


Figure 30: Histogram of the A and B values for tomato.

The A values for the tomato mask range between 35 and 55, while the B values range between 38 and 58. The tomato LAB colour mask is as shown below.

$$\text{Tomato\_mask} = (A \geq 35) \ \& \ (A \leq 53) \ \& \ (B \geq 38) \ \& \ (B \leq 58) \quad (5.2)$$

The A and B values from all pixels forming an item x in an image I, are likely to exhibit some level of correlation and be well approximated around the 2D histogram peak by a Gaussian. Therefore, the (A,B) distribution is approximated by a 2D Normal distribution with probability density function:

$$f_x(A, B) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\left(\begin{pmatrix} A \\ B \end{pmatrix} - \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}\right)^T \Sigma^{-1} \left(\begin{pmatrix} A \\ B \end{pmatrix} - \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}\right)\right) \quad (5.3)$$

where the means and co variances are calculated from all pixels forming item x:

$$\mu_A = E(A), \ \mu_B = E(B), \ \Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix} \quad (5.4)$$

$$\text{and } \Sigma_{AA} = E\left((A - \mu_A)^2\right), \Sigma_{AB} = E\left((A - \mu_A)(B - \mu_B)\right) \text{ etc.} \quad (5.5)$$

The object mask uses the Mahalanobis distance, effectively selecting pixels where the probability density is greater than a target value:  $f(A, B) > f_t$ . This allows the probability of erroneously mislabelling a pixel to be set. The binary mask for item x may be written:

$$M_x(I) = \begin{cases} 1 & f_x(A_{ij}, B_{ij}) > f_t \\ 0 & f_x(A_{ij}, B_{ij}) \leq f_t \end{cases} \quad (5.6)$$

The HSV colour mask is based on hue and saturation values only and may be used in the same way as A and B to produce a colour mask. For a kiwi fruit the H and S colour mask is:

$$\text{Kiwi Fruit} = (H \geq 0.085) \& (H \leq 0.15) \& (S \geq 0.15) \& (S \leq 0.55)$$

The proposed algorithm combines three binary masks added together: each pixel that appears in the final mask must be true in all masks. For example, the radish mask combines four masks, as follows:

$$\text{Radish mask} = (D \geq 50) \& (D \leq 130) \& \dots$$

$$(E \geq 3) \& (E \leq 8) \& \dots$$

$$(A \geq 30) \& (A \leq 65) \& \dots$$

$$(B \geq 5) \& (B \leq 30);$$

where

$D$  is the image in grayscale.

$E$  is the texture result from the entropy filter.

$A$  is the colour channel.

$B$  is the colour channel

The final binary mask is shown in Figure 32.

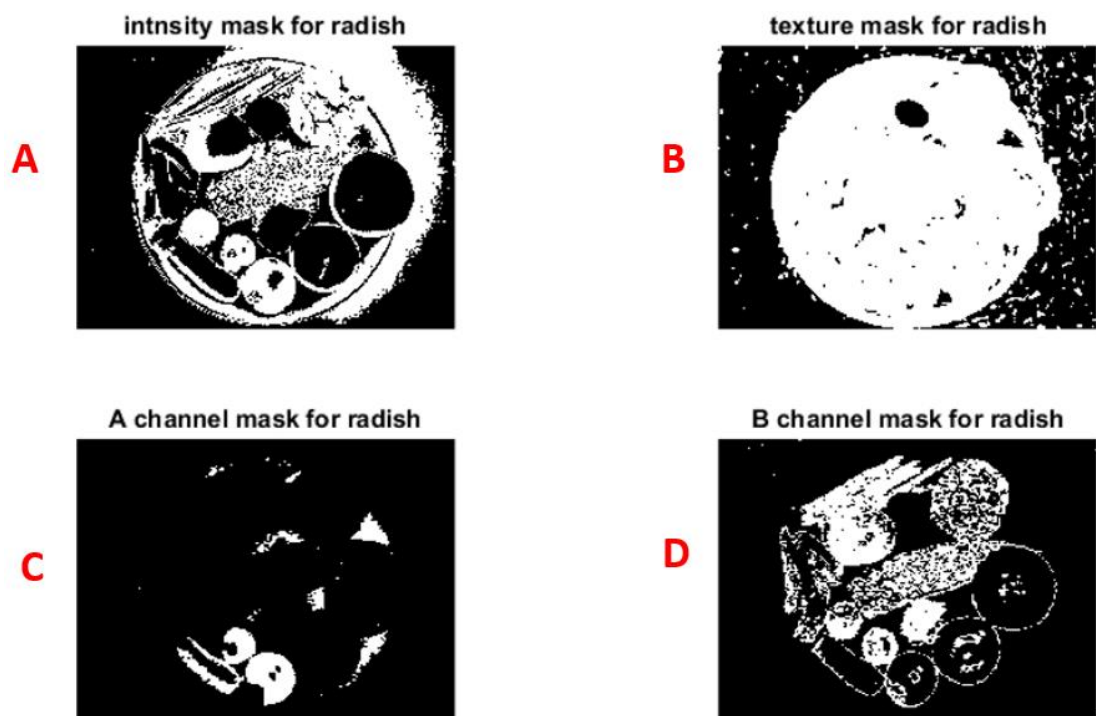


Figure 31: Four radish masks. The Radish Mask consists of four masks. (a) is the image output of the intensity mask for radish. (b) is the output of the texture mask for radish. (c) is the image output for the A channel for radish. (d) is the image output for the B channel for radish. The four images are combined to create the final mask, as shown in Figure 32.



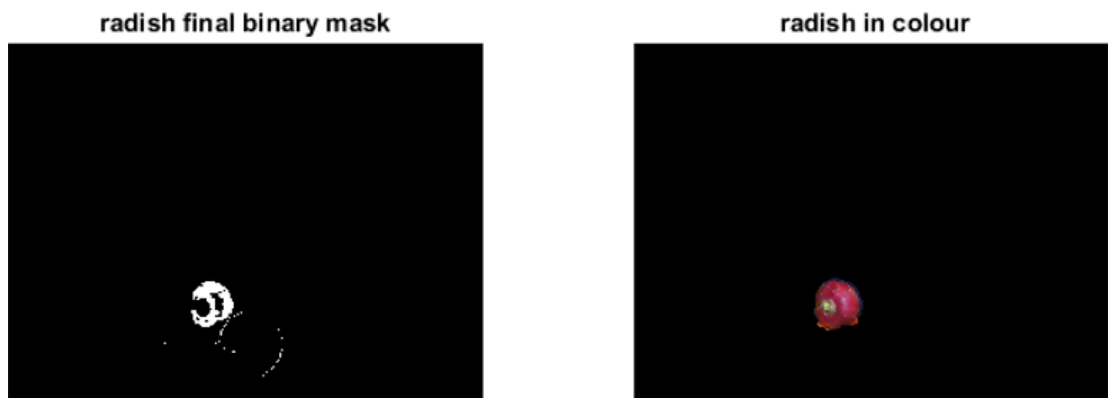


Figure 32: Final mask and extracted radish in colour.

Step three: display results and extract features, such as the object area.

In step three, the algorithm uses the binary mask from the previous step to extract the targeted food and display it. Also, the algorithm calculates the areas for all extracted foods. Figure 33 shows examples of extracted foods.



Figure 33: Examples of extracted foods.

### 5.3 Challenges and limitations.

The system has to overcome a number of challenges to be able to perform automatic and accurate recognition. The original image is in RGB format, which is sensitive to light variation. To reduce variation due to illumination, i.e. direct daylight, artificial lights etc., the algorithm uses colour space components that are less sensitive to light variations

such as H and S from HSV or A and B from LAB. Furthermore, the algorithm uses texture information along with colour. The algorithm successfully segmented food types illustrated in Figure 33, and failed to segment chicken nuggets, chips and egg. On the other hand, the algorithm correctly segmented and identified radish and carrot. The second example shows that nuggets and apple were correctly classified, while carrot, tomato, and avocado were misclassified. There appear to be two reasons for the misclassifications. Firstly, some items are very similar in the information used for classification: colour and texture. For example, the plate reflections and eggs, nuggets and chips, tomato and carrot; are pairs of objects for which these characteristics are similar. In most cases, a human observed could classify these objects correctly. However, the initial algorithm only used information derived from small regions and so the shape of the object is not detected. It also appears as though differences in illumination have not been completely removed by the choice of colour parameterisation. Although the over-all intensities are normalised, this does not remove shadows of objects. Unless the webcam and illumination are collocated, food items can cast shadows on the plate or on other items. This leads to misidentification of the number and types of items in the image. Even complicated, multi-source illumination is unlikely to solve this problem. Another problem is the natural variation in the colour and texture of foods. Even chips can be under or over cooked and range from pale yellow to black, and vary in shape from French fries to wedges. Selecting variable ranges to cover all variation is likely to lead to significant mask overlap and significant misclassification.

*Figure 34* shows an image of a plate of food captured using a webcam. The algorithm has correctly identified and segmented three types of food, which are avocado, croissant

and radish. It has misclassified reflections off the plate as egg because the plate surface reflects white and has similar texture to egg.



Image above is an input original image.



Above algorithm identifies part of the plate as egg.



Image above there is no object detected.

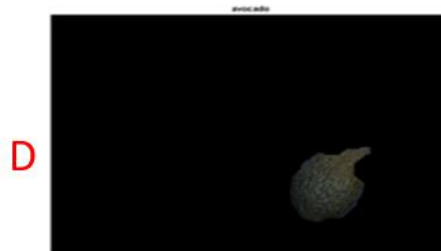


Image above, avocado detected.

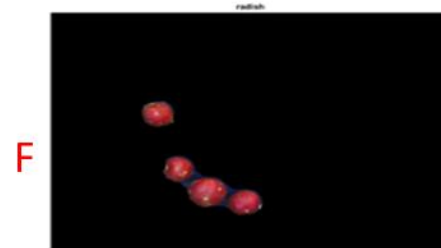


Figure 34: Example 1 shows challenges and limitations of the multi mask method. (a) The original image is a plate with four types of food. (b) The light reflection is identified as egg. (c) The carrot mask is not able to identify the carrot. (d) The avocado mask identifies part of it. (e) The croissant mask identifies it correctly. (f) The radish mask identifies it correctly.

#### Example 2

Figure 35 presents a second example. On a plate of five different types of food, only apple was identified correctly. Parts of the tomato and carrot were confused, because

of the similarity in colour and texture. Nuggets were identified as chips, or fish, once again because colour and texture are similar.

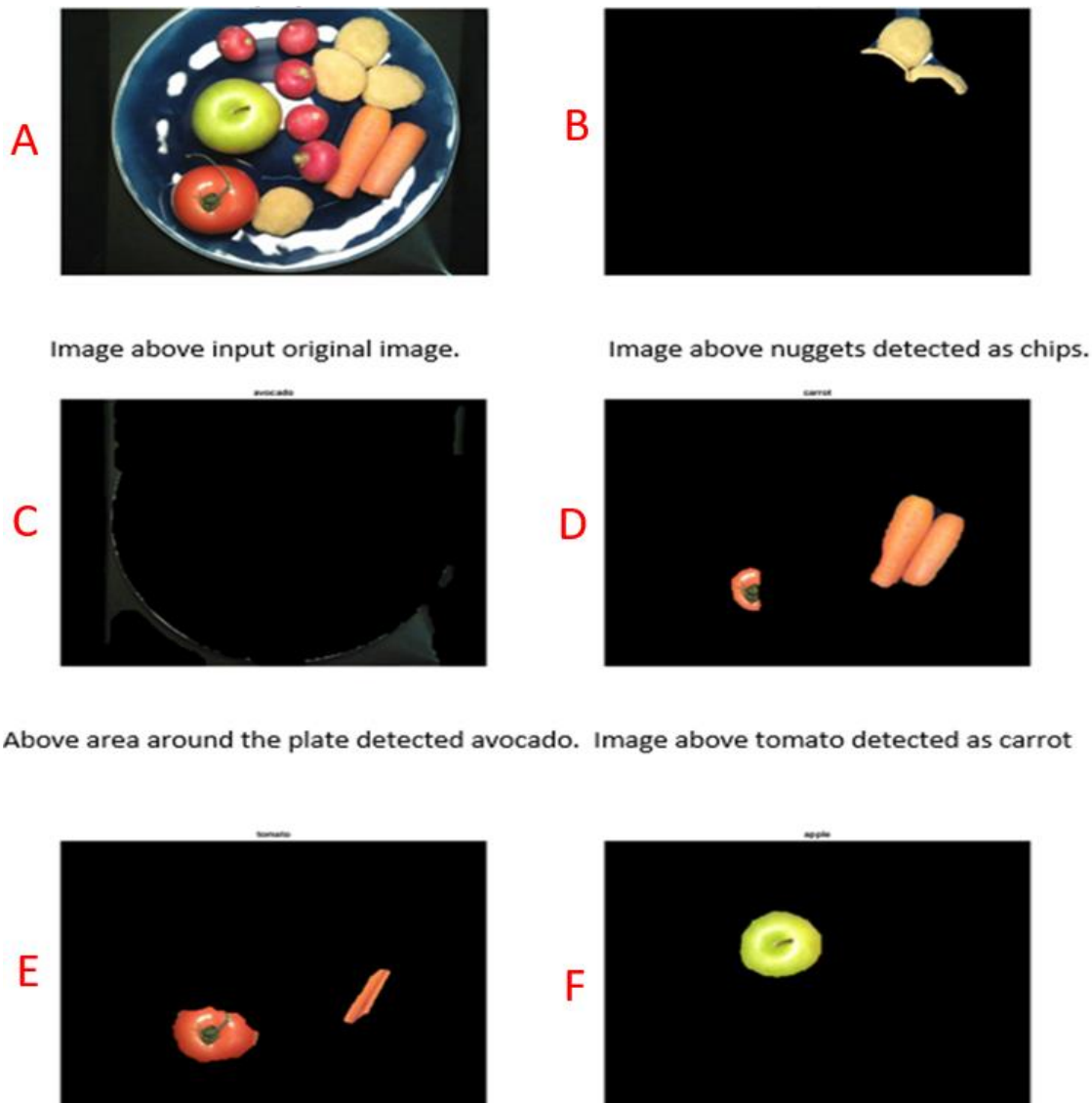


Image above input original image.

Image above nuggets detected as chips.

Above area around the plate detected avocado. Image above tomato detected as carrot

Figure 35: Example 2 shows challenges and limitations of the multi mask method. (a) The original image is a plate with five types of food. (b): The nuggets mask identifies part of it. (c) The area around the plate is identified as avocado. (d): The carrot mask identifies it partly and identifies part of tomato as carrot. (e) The tomato mask identifies part of it and identifies the carrot as tomato. (f) The apple mask identifies it correctly.

## 5.4 The second experiment

The second experiment was to evaluate the ability of Machine Learning algorithms (ML) to identify a range of fixed meals, i.e. a known selection of foods on a plate. Data for the experiment was collected in two locations: Portland private student accommodation and the refectory in Thwaite Hall student accommodation, both in Hull. This experiment is less ambitious than the first one as there are more visual clues to distinguish meals than individual foods. However, the experiment focus on institutional food similar to that served the expected major application area, i.e. hospitals and nursing homes.

### 5.4.1 The first location:

For the first location, images were taken in the large eating area, approximately 12 m by 15 m, at the Portland student accommodation in Hull city centre. Images were collected for 40 different meals using a 13 megapixel mobilephone camera. The images were taken in four different light conditions and for each condition, food was arranged in four different ways. The first light condition was near to the large north facing windows, the second in the middle of the eating area, the third just below a fluorescent lamp, and the last in the corner of the eating area. The total number of images of each plate was 20, and the total number of images of the 40 plates was 800. Each image included a white sticker to record plate number, e.g. p1, p22 and the light condition, C1, C2, C3, and C4.

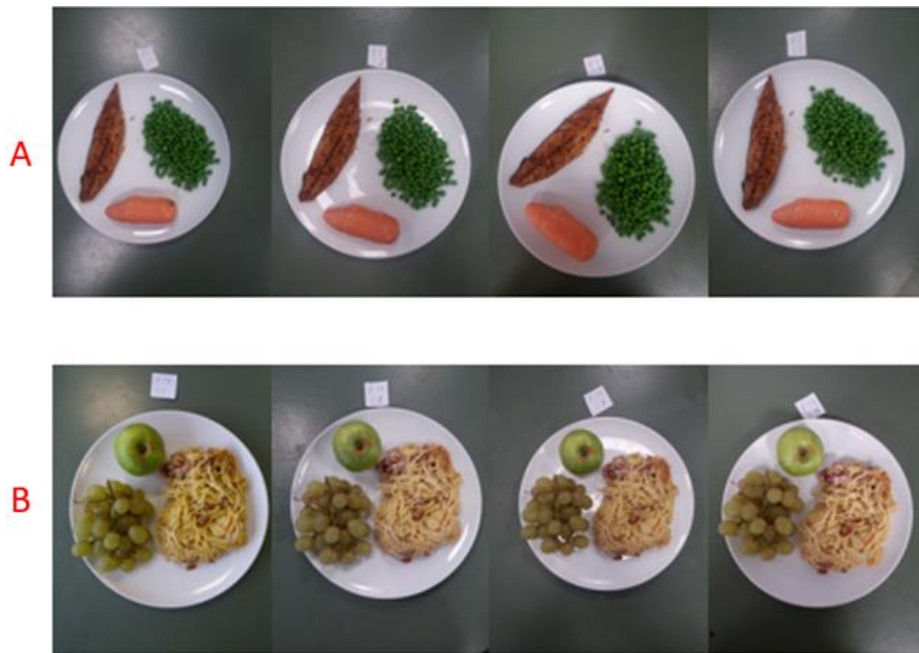


Figure 36: The Portland data collected in different light conditions. The figure shows two examples, A and B. For both examples the plate image uses captured in different locations in the dining area.

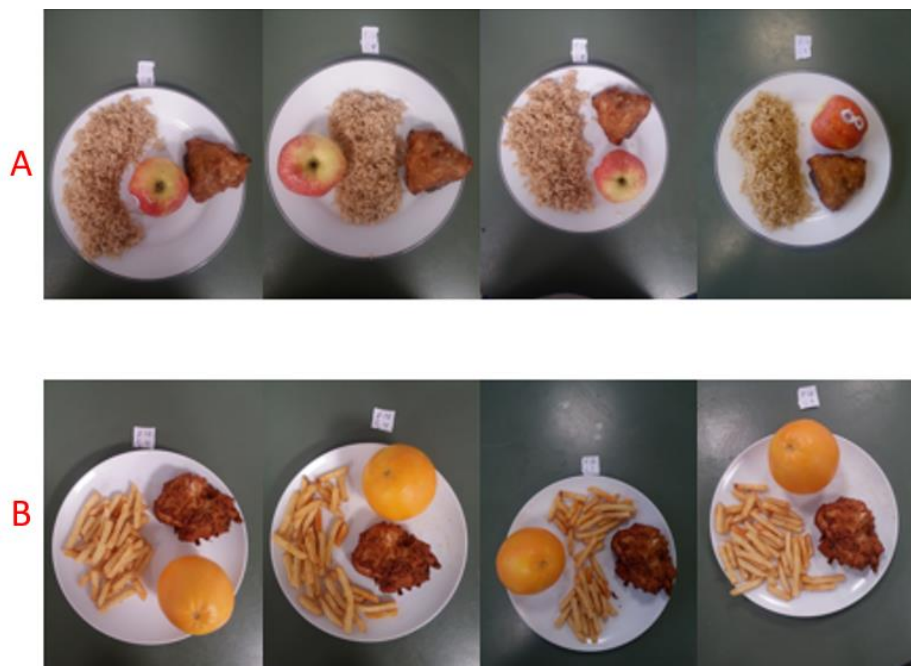


Figure 37: The Portland data collected in different food arrangements on the same plate. The figure shows two examples, A and B. For both examples the same food was rearranged for each image.

## Data processing

In preparation for data analysis, the images were categorised into 40 folders, each folder containing 20 images of one plate. The Bag of Features algorithm was used to extract image features. The total number of features was 17960. K-means clustering was then used to cluster features and create 100 of visual words. 80% of the data was used to train the SVM classifier and 20% used to evaluate the model performance. Figure 38 shows the confusion matrix illustrating SVM performance.

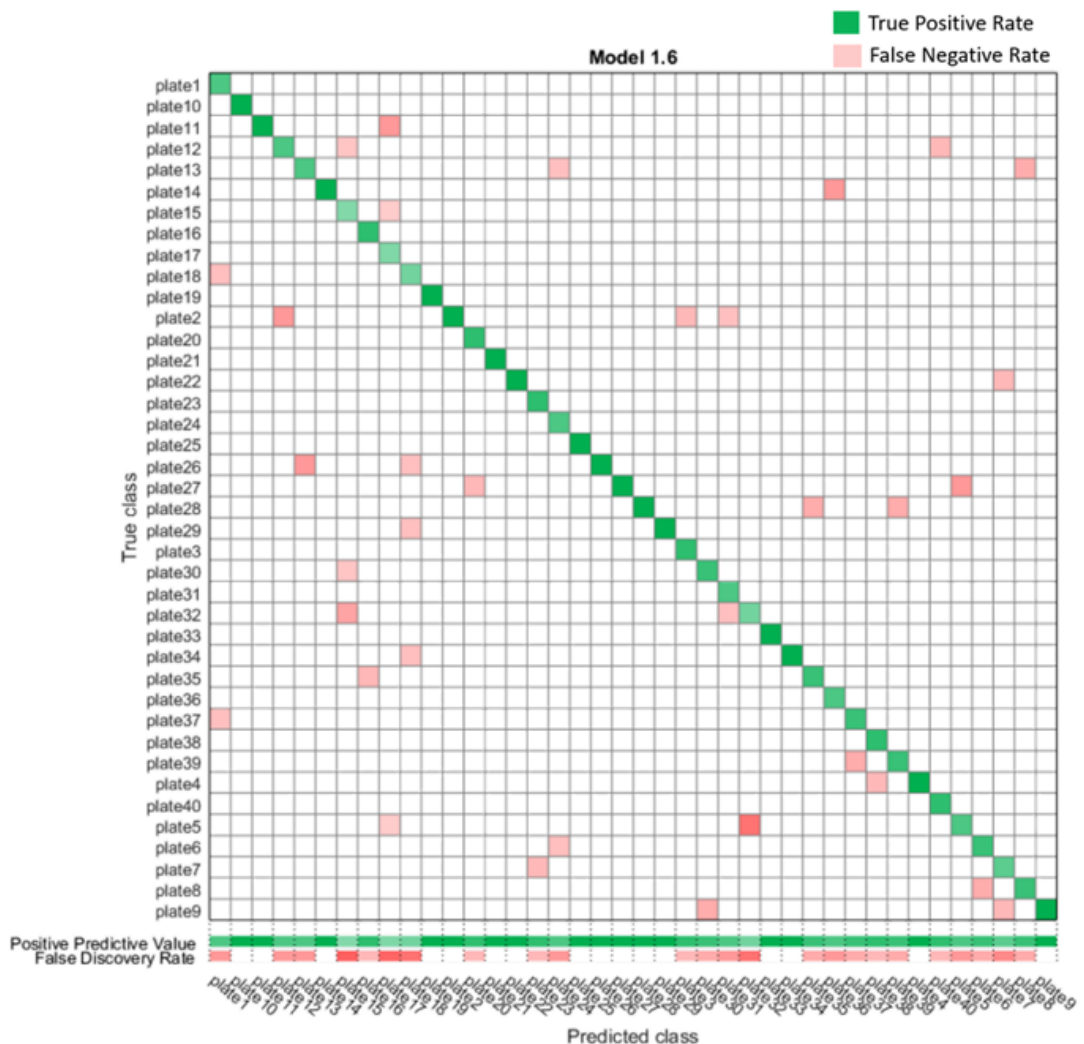


Figure 38: The SVM classifier identified most of the data correctly, with 76% accuracy.

The green (diagonal) indicates correct classification and the red means incorrect classification.

#### **5.4.2 The second location:**

On the 10th and 17th of February 2017, food images were collected from Thwaite Hall student accommodation in Cottingham, near Hull. More than 600 images collected for six different types of food, as follows.

- Option A day one: battered cod fillet, chips, and garden peas;
- Option B day one: beef stroganoff, rice, and green salad;
- Option C day one: mushroom and pepper stroganoff, jacket potatoes, and mushy peas;
- Option A day two: Thai cod and prawn fish cake, garden peas, and rice;
- Option B day two: penne pasta bolognaise bake, chips, and mushy peas;
- Option C day two: vegetable curry, naan bread, and baked beans.

As it is clear from Figure 39, there are similarities between food types. For example, there are similarities in colour and texture between beef stroganoff (Option B day one) and mushroom and pepper stroganoff (Option C day one). In addition, there is similarity in colour between rice and pasta. Texture is a similar between chips and pasta. The food background is white in all options.





Option A day one



Option B day one



Option C day one.



Option A day two



Option B day two



Option C day two

*Figure 39: Shows images for food meals from Thwaite Hall. For day one and day two the user can choose one of three options either A, B, or C. Or he can choose any other types of foods.*

Machine learning algorithm used to train and evaluate more than 20 classifiers. each classifier trained and evaluated to identify the six food categories. The Ensemble Classifier consistently yielded the highest classification accuracy of (94.9%) of correctly identified plates of food. The trained classifier (the model) performance evaluated using confusion matrix. The confusion matrix showed that option c day two had high similarity with option c day one(8%). this refers to the similarity in colour between two options.

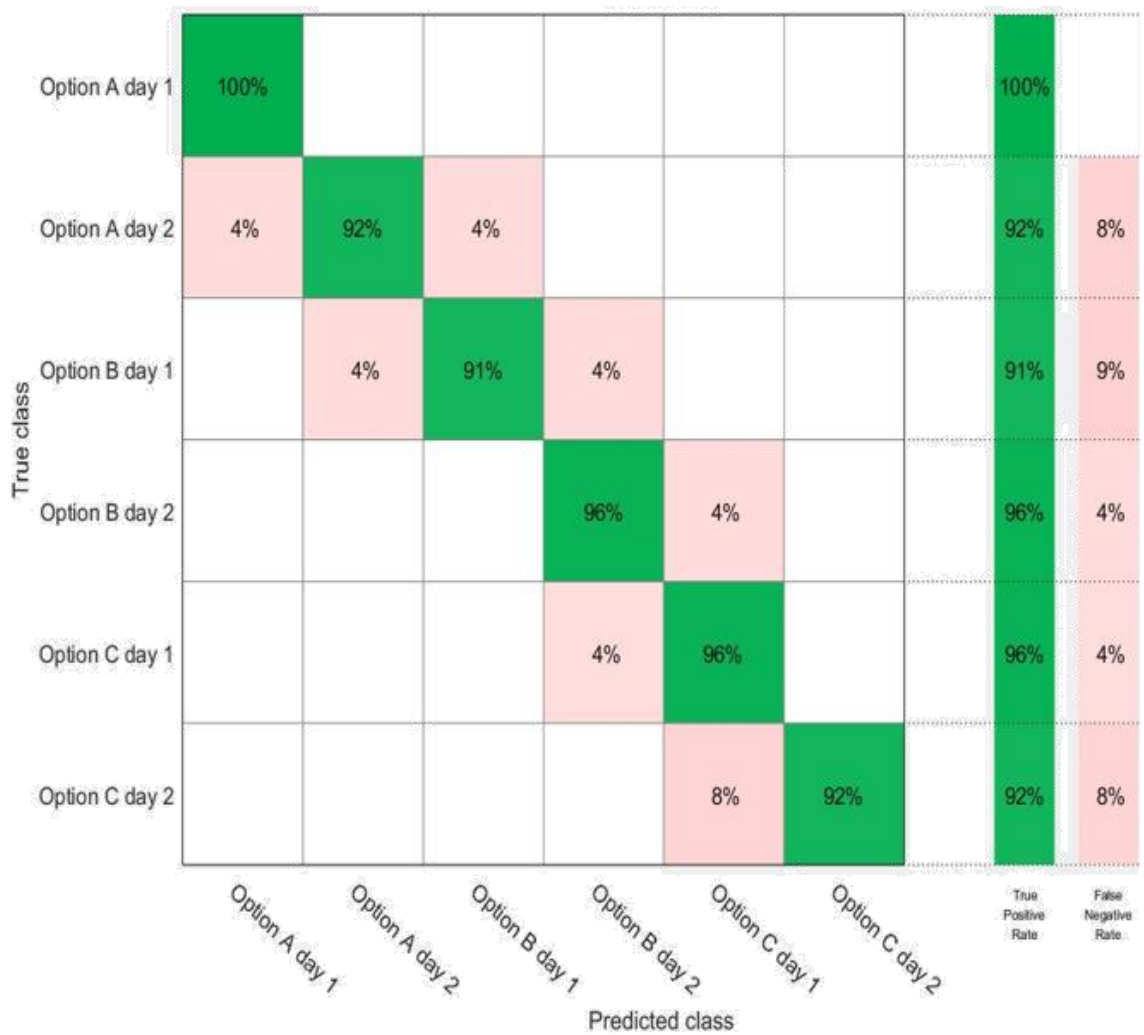


Figure 40: Confusion matrix for Thwaite Hall experiment.

## 5.5 Conclusions

The two experiments help to understand and explore Machine learning and thresholding techniques. The first experiment uses a thresholding algorithm to segment food types. A colour thresholding algorithm separates foreground and background. Multiple masks based on colour and texture are required to identify regions of a particular food type. However, in practice defining these masks would be time consuming and expensive. The second experiment tested the ability of Machine Learning algorithms to identify plates of food. This is an easier task and the ML

algorithms performed well enough to be used. This suggests that a two-stage approach may be required where, the plate of food is identified from those on offer on a particular day, then the individual food types can be identified and measured from the small set of foods known to be present. These conclusions inform the specification of the proposed algorithm to estimate the amount of food eaten, presented in subsequent chapters.

# Chapter 6 The Proposed Algorithm

## 6.1 Introduction

Chapter Six describes the complete method, which starts with a true colour image of a plate of food and yields a food recognition report. The major part of the algorithm aims to identify and separate the foods on the plate. The type and quantity of food are used to estimate nutritional values. In order to achieve this, the method progresses in three stages. In the first stage, the algorithm divides the image into two parts: foreground and background. The foreground represents the food and the background is any object that is not food. During the second stage, the foreground image (which contains the food only) is segmented into images, each of which represents a single type of food. Finally, in the third stage, each type of food image is classified. The food size is estimated in order to calculate the nutritional value.

## 6.2 Identify Foreground and Background

The purpose of this stage is to remove the image background to eliminate any object that is not food and keep all parts of the food. The background is non-food objects such as the table, placemats, cutlery etc., as shown in Figure 41. Removal of the background

is a pre-processing stage before the food segmentation stage. Accurate removal of the background improves the accuracy of food segmentation.

Backgrounds can be as diverse as foods, and so it is likely that each institution will need to train the background identifier to local conditions, e.g. wooden or Formica tables. Three methods of background removal were tested: circle finder, texture thresholding, and colour thresholding.

### **6.2.1 Circle Hough Transform (CHT) technique:**

The Circle Hough Transform (CHT) algorithm detects round objects in images, such as a plate. In portrait photography, it is used to identify irises in an image. We assumed that plates are circular, and so the CHT algorithm was tested to detect and segment a plate in the image (Atherton & Kerbyson, 1999).

The CHT algorithm may be applied to colour images and has five parameters: Object Polarity, Method, Radius Range, Sensitivity, and Edge Threshold.

Object Polarity switches between searches for a light circle on a dark background and vice-versa. The Sensitivity parameter determines if partial circles will be detected. Higher sensitivity helps the algorithm to detect weak and partial circles. The Edge Grading Threshold parameter determines whether sharp edges or blurry edges are detected. Radius Range specifies the minimum and maximum radius of the circles to be found. Two different CHT methods exist, and this parameter determines which is used.

Two experiments were conducted to illustrate how the CHT algorithm may be used to segment a plate from the image. The first experiment used a blue plate to provide a strong colour contrast between foreground and background. The results are shown in

Figure 41. When a white plate was used, the contrast was much less, and the CHT algorithm often failed to identify the plate, as shown in Figure 41, right.



Figure 41: shows CHT performance in detecting the plate in the image. (a) CHT algorithm successfully detects the plate when the background is different in colour from the foreground. The red circle defines the plate edges. (b) The algorithm deletes all parts of images out of the red circle(background) and keeps all objects in the red circle (the plate). (c) The CHT algorithm is not able to detect the plate when the background and the foreground are similar in colour. Therefore, the red circle cannot detect the plate edges.

As white plates are very common and CHT did not detect the plate accurately, other methods of background segmentation were explored. The methods are texture and colour threshold techniques.

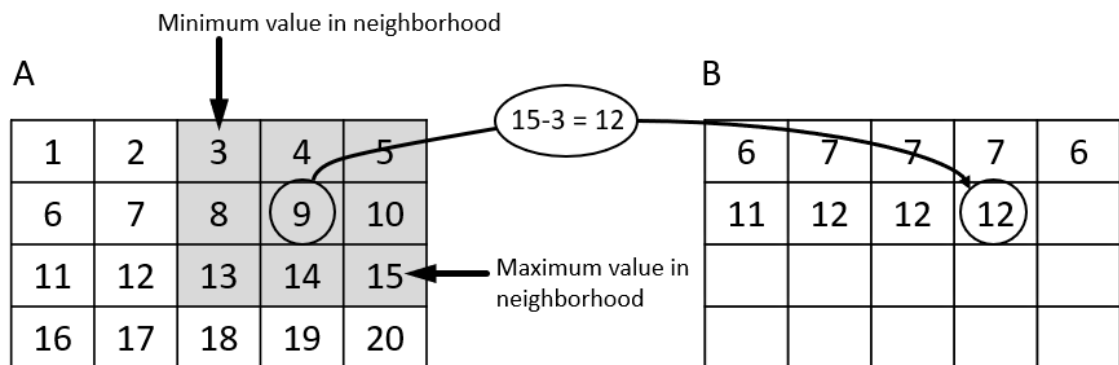
### **6.2.2 Texture technique:**

The texture technique looks for texture variation between food (rough) and plate (smooth). The visual texture of a greyscale image can be measured using standard statistical parameters such as entropy, standard deviation and range of pixel values over some region. For example, entropy is a statistical measure of pixel variability. The Entropy values range from 0 to 8 and when the entropy value is low, this means that the

surface is smooth with less detail and when the entropy value is high, it means that the surface is rough with more detail.

The statistical measures can also be presented as an image where the pixel value is the statistical measure derived from the region around that pixel. For example, the range value for each pixel is the difference between the minimum and maximum values of pixels in the neighbourhood. The pixel's neighbourhood is usually a square array of pixels centred on the target pixel, e.g. a 3-by-3 array, (see Figure 42). The standard deviation and entropy filters work in a similar way.

Texture segmentation is powerful when the texture of the object of interest is different from that of the background. Furthermore, texture measures are often less sensitive to light variation than other measures.



Determining Pixel Values in Range Filtered Output Image

Figure 42: Illustration of 3x3-range calculation (Mathworks, 2018).

The texture segmentation of a greyscale image uses the following steps:

Step 1: Apply one of the statistical filters (range, standard deviation or entropy) to characterize image texture;

Step 2: Create a binary mask by thresholding the texture image;

Step 3: Use morphological operations for removing of small objects, morphological closing, and filling holes.

Step 4: The binary mask may be used to segment the greyscale or original RGB image.

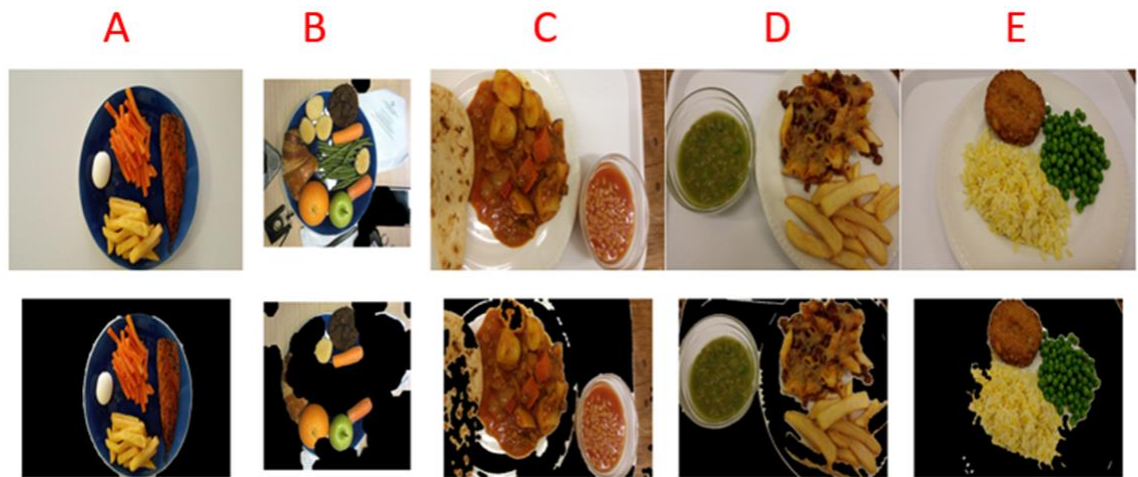


Figure 43: Illustration of texture segmentation applied to test images. (a) The image background is segmented successfully because the image background has fewer details (soft). ( b and c) The image background has more details and objects(rough), therefore the algorithm fails to segment the background. (d and e) The backgrounds of the images have fewer details, therefore they are partly segmented.

### 6.2.3 Colour Technique

In addition to texture, colour provides information that can be used to distinguish foreground from background. Colour is usually specified by three vectors of intensity. The most common three vectors are RGB, LAB and HSV. Figure 44 illustrates the same image of a plate of food, using grayscale images for each of the three different intensities. In this example, the HSV saturation variable appears to provide the most contrast between food and background.



With the saturation, the distinction between the white background and foreground (food) is clear. The technique has been tested on almost 3000 images from the combined Thwaite Hall and School of Engineering datasets, and performs better than adequately when the table surface has a uniform light colour, even over a range of viewing angles and distances. The most common reason for poor performance is when colour variation between the table surface and table cloth are visible at the edges of the image.

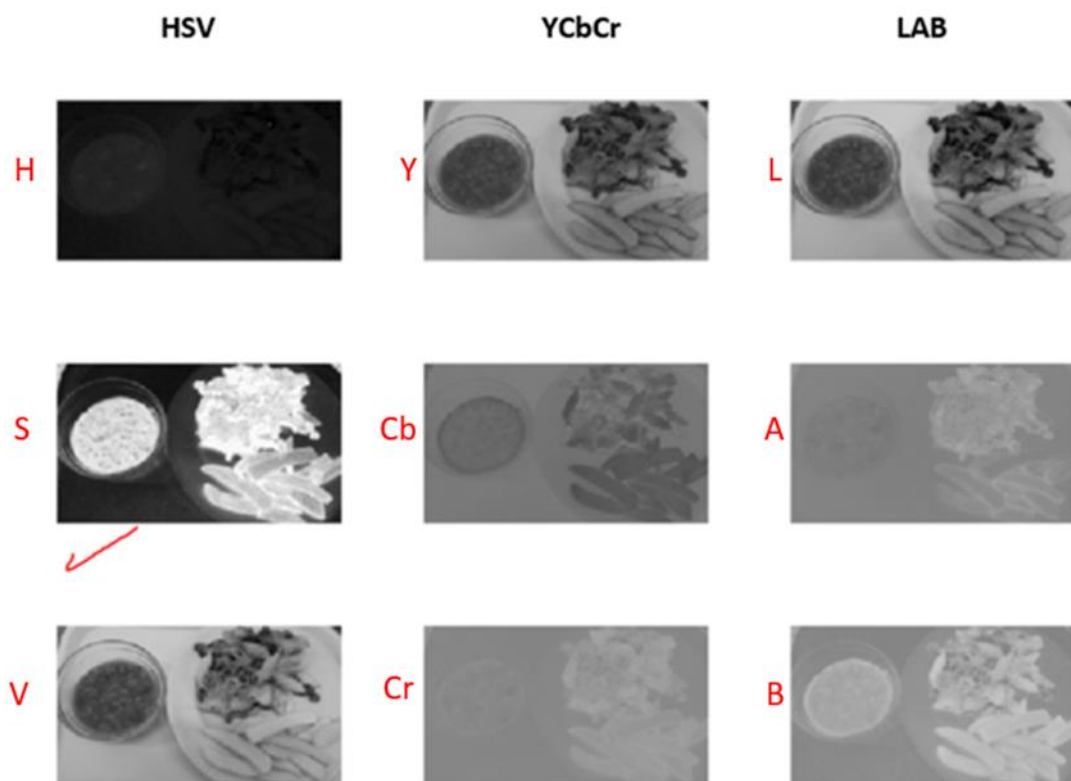


Figure 44: Illustration of the food plate represented in nine images for three colour spaces. The first column represents HSV, the second column represents YCbCr, and the third column represents LAB. As it is clear in channel S, the difference between the background and foreground is clear. The foreground is white and the background is black. The tick in red means that the channel S is used in the algorithm to segment the foreground from the background. In the other images, there is a similarity between the background and the foreground, which means some parts of the food segments are segmented as background.

The colour segmentation approach uses the following steps:

- Step 1: Convert RGB image to the HSV colour space and extract the S channel;
- Step 2: Choose minimum and maximum values based on the S histogram;
- Step 3: Create binary mask by thresholding;
- Step 4: Use morphological operations for removing small objects, morphological closing, and filling holes.
- Step 5: Use the binary mask to display the segmented image in RGB format.

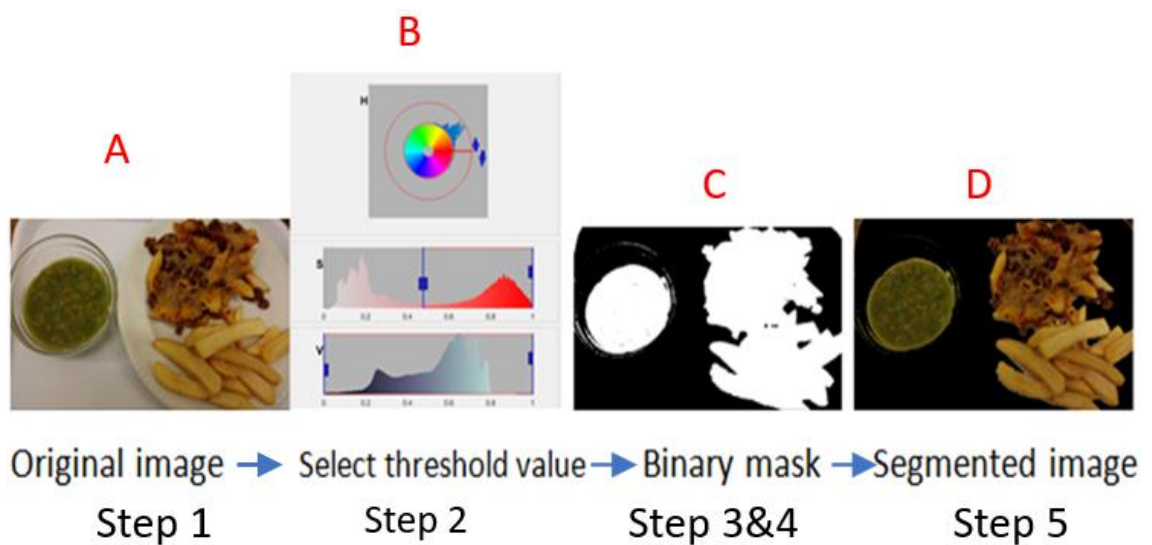


Figure 45: Illustration of the steps in applying the colour segmentation method. (a) The original image in RGB format. (b) Set minimum and maximum of thresholding values. (c) Threshold values are used to create the binary mask. All pixel values in the background are zero (black) and all pixel values in the foreground are one (white). (d) All the white part of the image appears in the final image and the black part is deleted. The result is a foreground segmented in RGB format.

Test images were collected from Thwaite Hall student accommodation. The image background (plate and tray) is white, therefore colour thresholding will be used to segment the background.



Figure 46: Eight examples of automated foreground identification. (a and c) The original images before segmentation. (b) Segmented images below the original images. As is clear, in all images the foreground is segmented successfully. (d) Segmented images below the original images. For images 5 and 6, the food is segmented successfully. Image 7 shows that the algorithm is able to eliminate cutlery. Image 8 shows that the algorithm fails to eliminate the highly coloured image background. The algorithm is designed to segment the white plate and cutlery.

The colour technique was tested using almost 3000 images and yielded near 98% accuracy. Figure 46 shows examples of test data. The algorithm is also able to eliminate bright objects like cutlery as shown in Figure 46 images 7 and 8. Image 8 is an example with a highly coloured Christmas table covering where the algorithm incorrectly identified some of the pattern as foreground.

## 6.3 Foreground Segmentation

The first stage aims to eliminate any object that is not food from the image. The K-means algorithm is used to segment the image into coherent parts. There are three initial choices that determine the success rate of clustering which are, selecting the initial cluster centres (seeding), choosing a distance measure, and finally, selecting the number of clusters.

### 6.3.1 Selecting Cluster Centres (Seeds)

Originally, MacQueen (1967) initialised centroids randomly. In 2007, Arthur and Vassilvikii proposed a smart approach to select centroids known as K-means++ (Arthur &Vassilvikii, 2007). The authors claim that the K-means++ improves both accuracy and speed (see section 4.4). Random seeding yield inconsistent clustering from the same original data. This is a problem addressed by K-means++.

"The careful seeding method of K-means++ avoids this problem altogether, and it almost always attains the optimal results on synthetic datasets. The difference between K-means and K-means++ on real-world datasets is also quite substantial. On the Cloud dataset, K-means++ terminates almost twice as fast while achieving potential function values about 20% better" (Arthur &Vassilvitskii, 2007).

### 6.3.2 Distance measure

Section 4-4 introduced a range of methods to compute the distance between data points. Where data vector elements have the same units and scaling, there is no driver for selecting an asymmetric distance metric or one more complex than the Euclidian or squared Euclidian distance:

"K-means is typically used with the Euclidean metric for computing the distance between points and cluster centres" (Jain, 2010,p 654).

This is the case for image data where vector elements are intensity distances in the colour space selected.

### **6.3.3 Cluster Number**

Section 4-4 introduced three methods to estimate number of clusters using an input parameter to clustering algorithms known as the k value. In this section, an entirely new method is developed that is particularly relevant to food images. A SVM classifier has been trained to estimate the k value for food images. Results show that the trained classifier successfully estimates the k value in food images with 97 % accuracy. The dataset of 2227 images was created by combining the Thwaite Hall and School of Engineering datasets, by selecting images with 1, 2 or 3 types of food present. A subset of 1782 images were used to train the classifier and 445 images were used to test. The trained SVM classifier categorised images into one of the categories in the following table.

*Table 5: Dataset sizes and performance of k value classifier.*

	Category (K value)	Total images	Image tested	Accuracy
One type of food and background	2	1445	289	99.7%
Two types of food and background	3	252	50	82%
Three types of food and background	4	530	106	97.2 %
Total average				93 %

The classifier counts the background and number of food types, so that  $k=4$  indicates background and three foods types. Previous Figure 46 shows example images where the classifier selects one food type (images 1 and 2), two types of food (images 3 and 4) and three types (images 5 to 7). The images 1 to 8 are part of the test set used to test the four methods in Table 6 .

Table 6 compares four different methods to estimate the k value. 50 food images collected from Thwaite Hall were used to test the Silhouette, Gap, and Davies Bouldin Criterion method. The Classifier method tested using dataset in previous Table 5. The Classifier method yielded the best performance on this dataset, correctly estimating the k value in 431 images out of 445 i.e. 97% accuracy. The Silhouette method correctly estimating the k value in 38 images out of 50 i.e. 76% accuracy. The Gap Method was 52% Accuracy. And The Davies Bouldin Criterion method was 8% Accuracy.

*Table 6: Comparison of methods to estimate the k value.*

Compare different methods to estimate the k value.			
Method	Image tested	Image corrected	Accuracy
Classifier method	445	431	97%
Silhouette method	50	38	76%
The Gap method	50	26	52%
Davies Bouldin Criterion method	50	4	8%

## 6.4 Food segmentation

After selecting the clusters number (k value) K-means is used to cluster images into coherent parts, generally with each part representing a single food. Figure 47 illustrates food segmentation using a Squared Euclidean distance metric and K-means++ centroid selection.

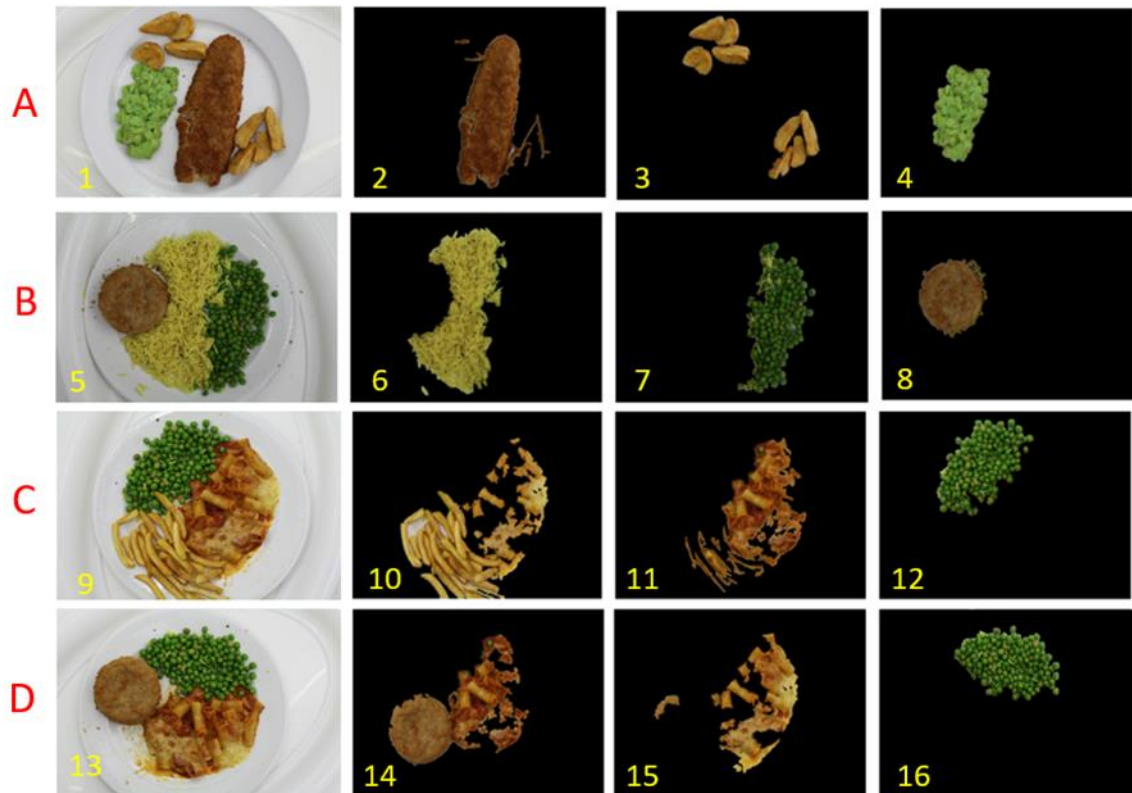


Figure 47: Four different examples of the final results of food segmentation. (a) In images 1&3, food is segmented successfully. In image 3, although potatoes is separated into two groups, the algorithm segments it as one type of food. (b) Images 6 to 8 show that all types of food are segmented successfully. (c) Images 10 and 11 show a part of the pasta segmented as chips. This is due to the colour similarity between the pasta and the chips. In image 12, peas are segmented successfully. (d) Images 14 and 15 show a part of the pasta segmented as Thai fish cake. This is due to the colour similarity between the pasta and the Thai fish cake. In image 16 peas are segmented successfully.

Figure 47 shows examples of successful and unsuccessful segmentation. For image 10, part of the pasta is unsuccessfully segmented into the chips cluster, probably because of the similarity in colour. In image 14, some pasta is unsuccessfully segmented into the Thai Fish Cake cluster. Images 3 and 4 are examples of successful segmentation.

Table 7 shows the data used to evaluate the food segmentation stage. The dataset of 1633 images was created by combining the Thwaite Hall and School of Engineering datasets. The average accuracy for all types of food was 94% accuracy

*Table 7: Segmentation accuracy.*

	Number of images tested	Number of Images successfully segmented	Accuracy
Chips	162	150	92.6%
Fish	157	137	87.3%
Mushy	245	245	100%
Pasta	291	274	94.2%
Pea	309	309	100%
Potatoes	125	99	79.2%
Rice	275	269	97.8%
Thai Fish cake	69	53	76.8
Total	1633	1536	94%

## 6.5 Food Type Identification

At this stage, segmented images are available containing a single type of food. An algorithm is developed to identify the food type within each sub-image. Supervised classification is used to train a model which will be used to identify food types. The classification steps are summarised as follows:



### **6.5.1 Step one: Training**

The classifier is trained with food images collected from the Thwaite Hall and School of Engineering. The dataset consists of 2749 Images, labelled into 8 food categories. 80 per cent of dataset (2199 ) used to train the classifier and 20 per cent (550)to teste the classifier

*Table 8* gives details about the dataset and *Figure 48* shows samples of the data.

### **6.5.2 Step two: Feature extraction.**

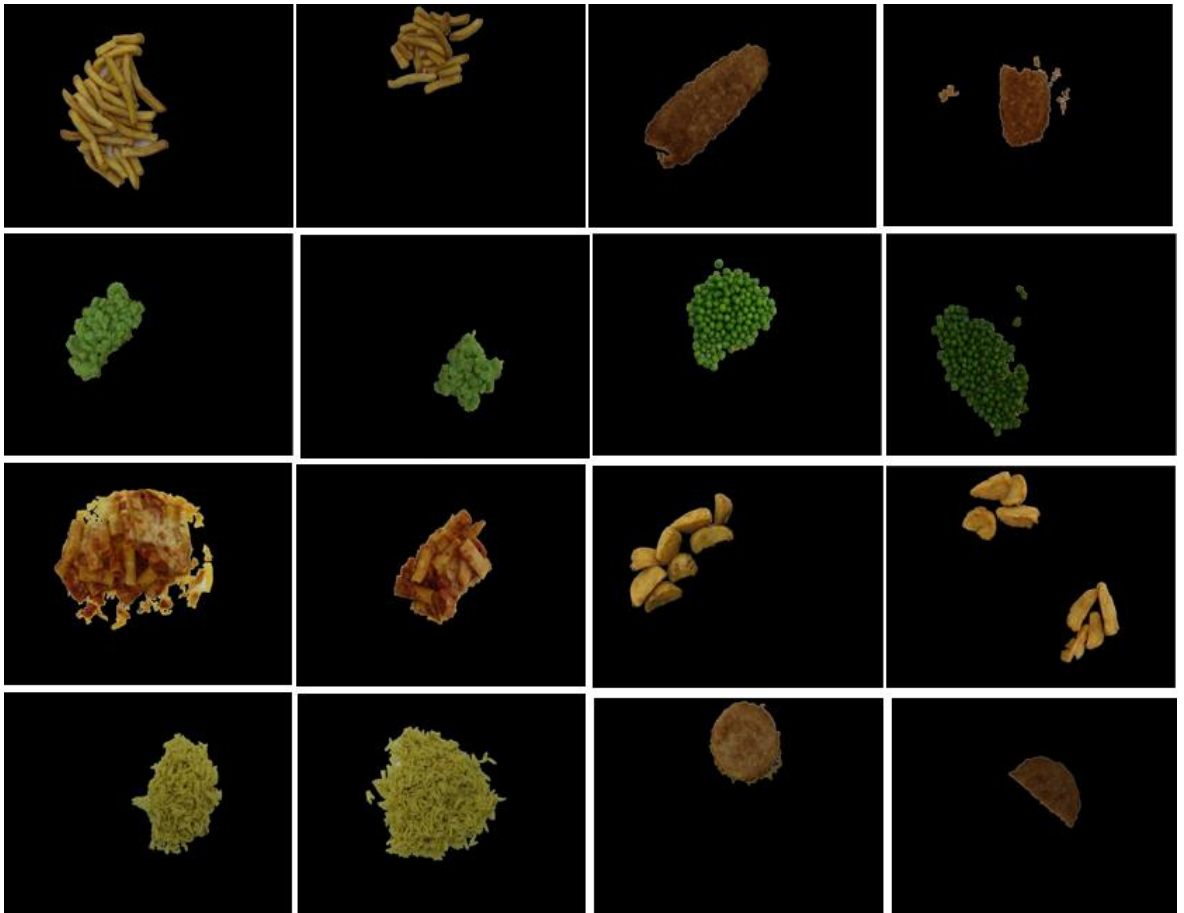
Bag of Features was used to detect and extract the visual words. The process starts by selecting key points and ends by creating a histogram of visual word occurrences. We can summarise the bag of the visual words in the following steps. The algorithm extracts features from the training images. Each category has a specific number of extracted features. For example, 308 images of Pasta yield 215046 extracted features.

Keeping 80 percent of the strongest features from each food category are retained. Then the number of features is balanced across all food categories to improve clustering. Image category 8 (Thai Fish Cake) has the least number of features 58239. The 80% of Thai Fish Cake is 46591. The total number of features is the number of food categories times the least number of strongest features i.e.  $8 \times 46591 = 372,728$  features.

Keeping 80 percent of the strongest 46591 features from each of the other image categories are used.

Table 8: Dataset used to train the classifier.

<b>Dataset used to train the classifier</b>			
Food category	Thwaite Hall student accommodation	School of Engineering	Total number of images by combining the two sets
Chips	106	197	303
Fish	143	224	367
Mushy peas	116	238	354
Pasta	105	203	308
Peas	111	330	441
Potatoes	102	236	338
Rice	163	229	392
Thai fishcake	128	118	246
Total	974	1775	2749



*Figure 48 Samples images from the dataset used to train the classifier. shows a sample of images used to train the classifier. The images include food in full portion and a part of the food. The images help to train the classifier to identify the food types before and after eating.*

### **6.5.3 Step Three:** Cluster features to create a dictionary of visual words.

K-means is used to cluster the total number of features (372,728) into 800 clusters to create 800 visual words. The following figure shows the visual word histogram for three types of food: Chips, Peas, and Thai Fish Cake.

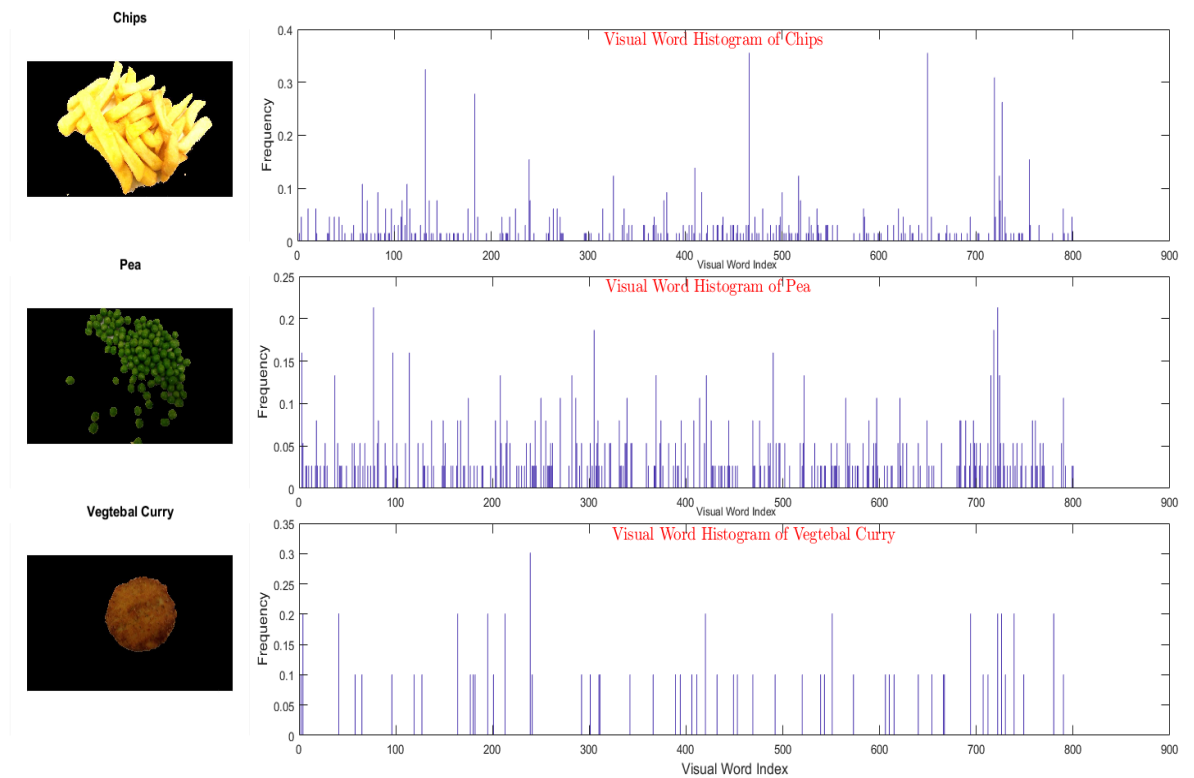


Figure 49: The visual word histogram of chips, Pea, and Thai Fish Cake. The x-axis represents visual word index which consists of 800 words and y-axis represents frequency of occurrence.

#### 6.5.4 Step Four: Train a Classifier to identify food types.

The visual word vocabulary and label data are used to train and validate the classifier. This informs the classifier that a particular set of features describes a type of food. Images of single food types have been manually classified. These are used during the learning phase to build a model or classifier. After training 21 different classifiers, it was observed that the Ensemble Subspace KNN classifier performed best on the training data.

Table 9: Summary comparison between 21 trained classifier (model)

	Classifier	Accuracy
1	Complex Tree	71.6%
2	Medium Tree	63.4 %
3	Simple Tree	46.1%
4	Linear Discriminate	97.6%
5	SVM Linear	98.4%
6	SVM Quadratic	98.4%
7	SVM Cubic	97.8%
8	SVM Fine Gaussian	25.3%
9	SVM Medium Gaussian	95.8%
10	SVM Coarse Gaussian	92.5%
11	KNN Fine	93.1%
12	KNN Medium	74.0%
13	KNN Coarse	54.1%
14	KNN Cosine	88.3%
15	KNN Cubic	82.0%
16	KNN Weighted	75.8%
17	Ensemble Boosted Tree	77.6%
18	Ensemble Bagged Tree	90.7%
19	Ensemble Subspace Discriminate	98.4%
20	Ensemble Subspace KNN	98.7%
21	Ensemble RUSBoosted	70.9%

Table 9: summarizes the comparison between 21 models. The third column shows the classification accuracy measured using 20% of the dataset (550 images) as training data.

#### **6.5.5 Step Five: Evaluate classifier performance using a confusion matrix.**

The Ensemble Subspace KNN classifier is evaluated using the 550 food images not in the training dataset and yields 98.7% accuracy. A confusion matrix is a diagnostic tool that provides information on classifier performance. Elements on the diagonal count correctly classified images (green) while off-diagonal cells represent misclassifications (red). A good classifier would have 100% on the diagonal and 0% everywhere else. In

Figure 50 , the highest accuracy is for five categories, which are Mushy Peas, Pasta, Pea, and Thai Fish Cake. The lowest accuracy is 97% for Chips, Fish, and Rice.

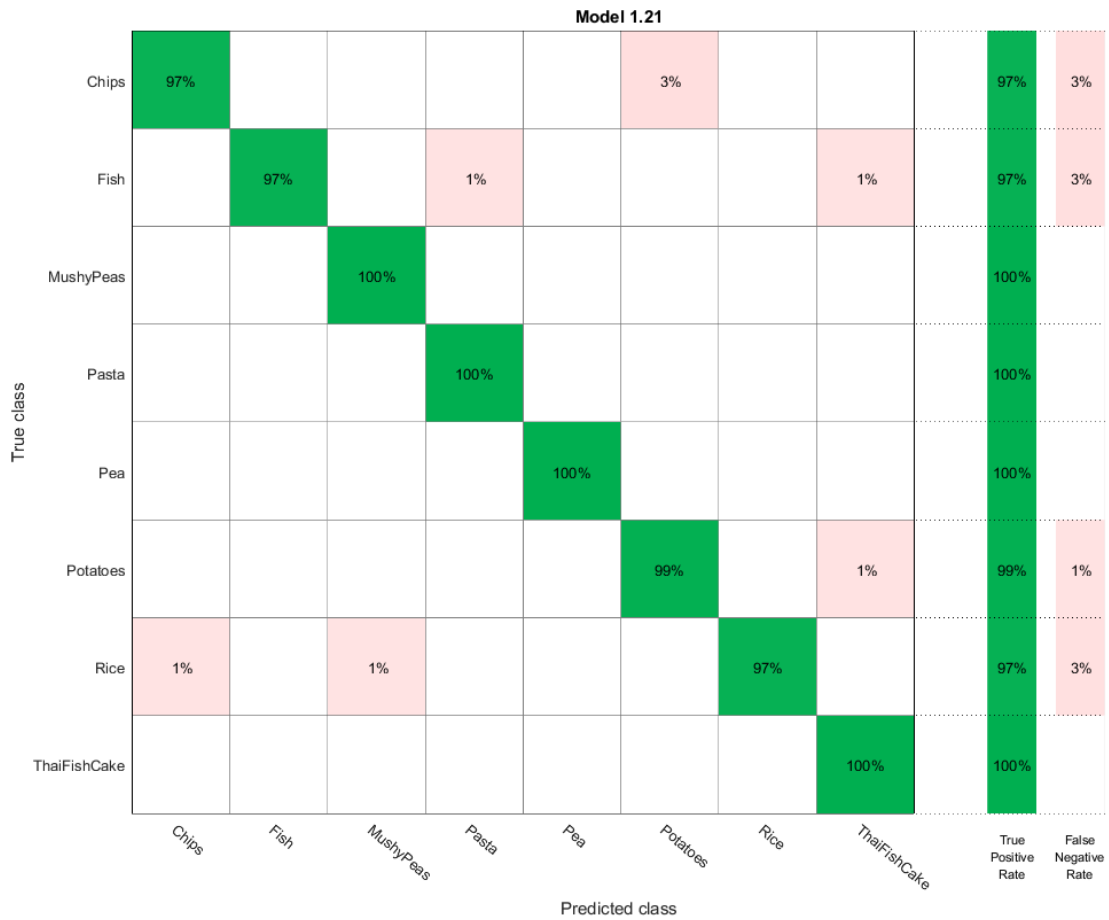


Figure 50: The confusion matrix shows the classifier performance in identifying food data. The diagonal (green) indicates the good performance of the model. The off-diagonals have been misclassified.

### 6.5.6 Step Six: Prediction step and results.

The following two examples illustrate the complete method. The first shows correct segmentation and identification. The table consists of six measures, which are: food type weight, fat, saturated fat, salt and sugar (all in grams) and energy in Kcal.

Food weight was estimated from the food area in pixels. For example, one piece of Thai Fish Cake weighed 140 g and the surface area was 736308 pixels. Therefore, 1 g of Thai

Fish Cake is approximately 5259 pixels. The nutritional content of food was taken from the packaging in which it was supplied to the catering services.

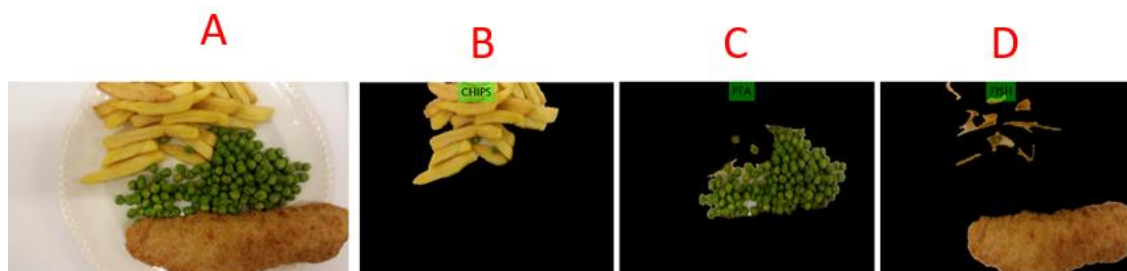


Figure 51: A example of food segmentation and identification. (a) The original image.

Notice that the chips overlap with the peas. (b and c) Despite the overlap, the algorithm segmented and identified the food correctly, as shown in the green boxes. (d) Image to show the algorithm limitation. The fish is identified correctly, and part of the chips is identified as fish.

Table 10: The final results of food nutritional values estimation summarized as an Excel table. A shows food types. B weight estimated. C to G illustrates the nutritional values.

	A	B	C	D	E	F	G	H
1	Food	Weight_gm	Energy_Kcal	Fat_gm	Saturates_gm	Sugars_gm	Salt_gm	
2	Chips	201.96	424.11	19.19	2.02	30.29	1.41	
3	Pea	173.72	72.96	0.97	0.19	1.74	0.35	
4	Rice	170.19	280.81	4.49	1.36	2.45	0.41	
5								

The second example illustrates the estimation of the amount of food eaten by analysis of images taken before and after eating.

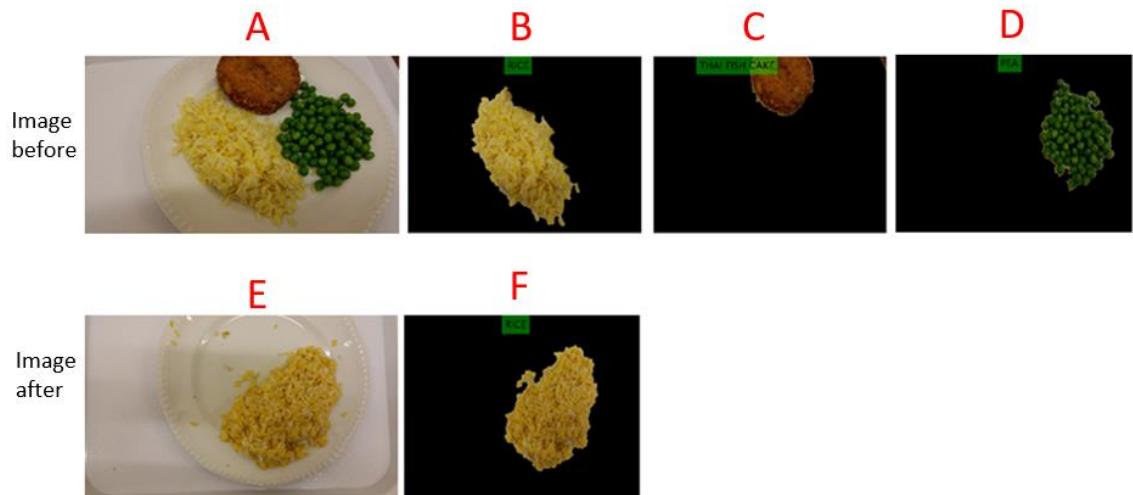


Figure 52: (a) Shows the image of the plate before eating. The plate contains rice, Thai fish cake, and peas. (b to d) Are the segmented images for the image before eating. (e) Shows the image of the plate after eating. The plate contains the rice only, which means that the patient ate the Thai fish cake and the peas. The algorithm estimates the food eaten by comparing the food images before and after eating. The results in Excel format as shown in Table 11.

Table 11: illustrates the estimation of food intake. The quantity of food eaten is estimated from the difference between the areas spanned by food types in images before and after eating.

	A	B	C	D	E	F	G	H	I	J
1	Food	WeightBefore_gm	WeightAfter_gm	FoodEaten_gm	Energy_Kcal	Fat_gm	Saturates_gm	Sugars_gm	Salt_gm	
2	Pea	100.00	0.00	100.00	42.00	0.56	0.11	1.00	0.20	
3	Rice	150.00	138.91	11.09	18.30	0.29	0.09	0.16	0.03	
4	Thai Fish Cake	140.01	0.00	140.01	238.02	10.50	2.38	1.40	1.54	
5										



### **6.5.7 Estimating the proportion of food eaten**

This section presents results that address the primary research question: can the proportion of food eaten be estimated from images of the food before and after eating. Although the problem is very difficult, the required accuracy is low. The clinically useful pen-and-paper system currently in use at Castle Hill Hospital only estimates the proportion of food eaten into five categories i.e. nil, 25%, 50%, 75% and all. The type of food eaten is not recorded. Consequently, a record of 50% meal consumption could be associated with very different nutritional intake, depending upon whether the starch or protein was consumed. The system that has been developed through this report appears to have the capability to do better than this.

A database of 200 images was collected in the School of Engineering. The plates of food were chosen to be representative of the types of Yorkshire institutional food available in hospitals. The images are grouped into pairs showing the plate of food before and after eating. Each pair of images is manually assessed by the author and allocated to one of the five proportion categories. Figure 53 shows examples from the dataset with odd numbered images being before eating and even images after eating. Each pair of images is passed through the system i.e. the food types are segmented, identified and the areal coverage estimated. The estimation of the percent of eaten food is done by using the area ratio of each food identified in the two images (before and after). For example, for a plate containing initially chips among other food, the ratio between the area occupied by the chips in the after plate and the area of chips in the before plate image gives the remaining chips ratio. The ratio of eaten chips is 1 minus the remaining chips ratio. Note that there is a case where no chips is detected in the after plate and in this case the area occupied by chips is set to 0. The table gives an example of csv file generated by the food

recognition ran over a set of two before/after plate images. The difference between before and after eating is used to estimate the proportion of food eaten:

$$food\_eaten\% = \frac{(area\_before)-(area\_after)}{(area\_before)} \times 100\% \quad (6.1)$$

There is a tendency for food to be spread-out on the plate during eating. Food with sauces, particularly sauces with strong colours, tend to leave residue over a large area of the plate. Other food, such as Thai Fish cakes, do not exhibit this at all. The spreading-out feature sometimes leads to estimates of food after eating being larger than before, and in this case the *food\_eaten%* is set to zero. To address the problem of food spreading, different food density values i.e. grams of food per pixel; could be used before and after eating. This has not been done for this particular trial.

Over the full dataset, the average difference between the manual estimate of the proportion of food eaten and the automated estimate, was 13.6%. Given that the manual categorisation yields an uncertainty of  $\pm 12.5\%$ , this is close to the best that could be expected.

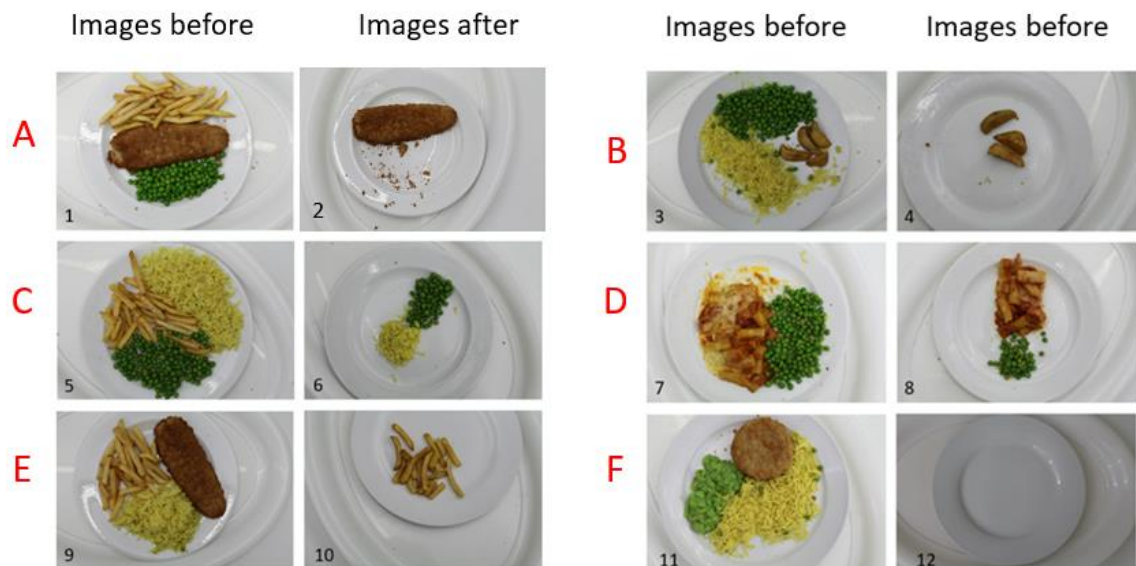


Figure 53 shows six examples(a to f ) of data set used to test the system accuracy to estimate the proportion of food eaten.

Table 12 food eaten report where 1 means all of food is eaten and 0 means no food eaten.

	A	B	C	D	E	F
1	Food	Areabefore	AreaAfter	FoodEaten_P		
2	Chips	2864086	0	1		
3	Fish	2771592	2775154	0		
4	Rice	3089162	0	1		
5						
6						

The automated system can also state what part of the meal was eaten. As can be seen in Table 12, the system will report not just the proportion of the meal that was eaten, but state that the Chips and Rice categories were eaten but the Fish Category was not.

It is then known that the participant has consumed significant amounts of starch but very little protein.

## **6.6 Conclusions**

An automated, image processing based system has been developed to estimate the proportion of food eaten from images of the plate of food before and after eating. In a trial of 200 plates of food, the automated system has yielded an estimate of the proportion of food eaten that is close to the uncertainty in a manual estimate. The system provides more information than the system currently in clinical use as it also identifies the food items that have been eaten.

# Chapter 7 Discussion and Future Work

7

## 7.1 Introduction

Chapter Seven describes the specification of the proposed system. Description of the hardware of the system will help to implement the proposed future work. In addition to that, the chapter includes discussion, conclusion, and future work

## 7.2 Specification of the Proposed System

To build an accurate food recording system for care homes or hospitals, it is necessary to overcome the difficulties such as variations in environment and food appearance.

Globally, the number of food types is vast, making general food recognition inaccurate. Restricting the training and testing of food types to UK hospital or care home food is likely to greatly increase accuracy. Restriction to the food types available at a particular institution on a particular day decreases the number of food types even more and should also increase food recognition accuracy. Additionally, the variety of food appearance will be limited. To develop an algorithm able to recognize buffet meals, where the mixture of foods on a plate is not prescribed, the algorithm will be trained to recognise more than one object in the image.

To address environmental variation, this project will look at both a fixed central system and a portable system. In the first scenario, the camera will be installed in a fixed position in the patient's room or in a dining area. The system consists of a professional camera, a wireless communication system and a central computer with software. The camera is mounted on the ceiling above the patient's bed or dining table. The camera light sensor will be able to compensate for general intensity variation, but not be able to adjust for side lighting leading to shadows. The camera could be connected to a central computer wirelessly as shown in Figure 54: Trolley system (camranger, 2017). The fixed camera to plate distance allows the camera focus can be fixed. The food image may be captured automatically and sent to a central computer for archiving, to analyse the image and display the results.

The portable system consists of four major parts, which are the professional camera, light source, trolley, and computer including software. The camera and light source are mounted about 50 cm above the plate on the trolley. The computer analyses data and displays results. There are several advantages to this method, such as controlling the environmental conditions (light variation), distance and camera angle. A fixed standard light helps to get a clear image, to avoid shadows and background variation. The shorter distance between the camera and food should yield improved resolution images. The fixed distance between camera and food helps to calculate food volume accurately. The camera angle should be directly above to get a clear image and to avoid tall food eclipsing short food. This position is the best to show all food items and to avoid shadows.

The portable system could be integrated into the food distribution trolley. Images would be collected when food is supplied to a user and when the leftovers are cleared away.

This also addresses the problem of linking plates of food with individuals as this is routinely done in hospitals as food is often linked to medical condition. For example, the patient wristband or bar code identifies the individual. Only the staff distributing and collecting food need be trained in the system and no patient cooperation is required. The system will be able to recognize multiple foods on a single plate.

The algorithm will be able to estimate the proportion of food eaten by the patient. The results can be presented in a similar way to the food chart used in Castle Hill hospital, for example: nil, 1/4, 1/2, 3/4, and all.

The system will be a portable unit that includes a trolley, professional camera, computer, and software, see Figure 54: Trolley system (camranger, 2017).



Figure 54: Trolley system (camranger, 2017).

### 7.3 Discussion

In this section, I will compare the study results with those of previous studies, in order to identify their relative strengths and limitations. I will begin with the current system

used in Yorkshire hospitals, which is the manual method, using paper and pen (Green & McDougall, 2002).

The manual approach to recording food intake does not provide information about the energy, protein and nutrients consumed, and it is not accurate. The manual system only estimates the amount of food eaten (such as 1/4, 1/2 or nil), but it does not calculate the nutrition and energy consumed. There are also concerns about human error when writing the reports manually.

On the other hand, this study proposes a system that is able to produce and save the reports with less effort from the medical staff. As we know, modern hospitals need to save patients' data in digital format. Therefore, the results are saved in Excel format, which can be used in statistical operations and for other possible operations. Patient data can be used to improve service quality, for example, such as reducing food waste.

As an alternative system, "Hospitalfoodie is an interprofessional case study of the redesign of the nutritional management and monitoring system for vulnerable older hospital patients" (Moynihan et al., 2012). In Hospitalfoodie, each patient had a bedside touch screen for nutrition management. The bedside touch screen presented an image of the food provided and prompted the user to rub away the food consumed; calories and nutrients were then automatically calculated (Moynihan et al., 2012).

Hospitalfoodie needs a touch screen to be installed beside each bed, which would increase the cost of the system and make it impractical. By contrast, the trolley system needs only one unit to cover one or two wards. The low cost of the trolley system makes it practicable and more efficient. For example, a 100-bed hospital needs 100 touch screens. whereas with the trolley system it requires only four units.



The bedside touch screen presents an image of the food provided and prompts the user to rub away the food consumed, then makes an automatic calculation. However, this is not an accurate method of calculating the food eaten, and the probability of human error is high.

The trolley system captures food images before and after eating, then saves them in the patient's file as part of the report. According to our knowledge, the trolley system is the first system to capture and save an image after the patient has finished their food. The image after eating is an important image for estimating consumed food, and it is saved in the patient's file as part of the patient's report.

Hospitalfoodie is still a proposed design; therefore the study could not supply any details about the system's accuracy.

The third study proposed a personal mobile app to record food intake. "The authors propose a personal software instrument to measure calorie and nutrient intake using a smartphone or any other mobile device equipped with a camera" (Pouladzadeh et al., 2014). They described a one-time calibration process for the thumb, which is used as a size reference to measure the real-life size of food portions in the picture (Pouladzadeh et al., 2014). That being the case, the algorithm is designed to be a personal record system.

However, it can be suggested that using the thumb is not comfortable for the patient, especially for older people. In addition, the results show that the method is not accurate when the plate contains food similar to the thumb in colour. Moreover, the colour and the shape of the nail is different from that of the thumb, which leads to the thumb being segmented without the nail.

The study assumes that the user eats all the food on the plate. For this reason, the study does not show any results or images after eating. As we mentioned before, the image after eating is important data for the patent file and to estimate the consumed food.

FoodLog is a popular online food recording system. The authors describe “FoodLog” as multimedia food-recording system which enables users to record their meals easily. Users upload photographs of their daily meals to FoodLog (<http://www.foodlog.jp/>), and it constructs their food diary automatically (Aizawa et al., 2013). The website uses image processing to classify food into five categories: grains, vegetables, meat/fish/beans, fruit, and dairy products. It is clear that the algorithm does not estimate the amount of consumed food and nutritional values, but it analyses food images to estimate the food balance between the five categories.

## **7.4 Conclusion**

The study has designed an automated system to monitor food intake, with an emphasis on the image processing part of the system. Tests indicate that the system could replace the paper and pen approach used in Hull and East Yorkshire Hospitals, and yield similar or better nutritional metrics.

The food image processing method has been developed in stages with the constraints on each stage considered, a range of potential solutions evaluated and the final solution validated. The first stage is to remove unwanted objects in the image background. Three methods were tested to identify and eliminate the image background: the Circle Finder (CHT), Colour Thresholding, and Texture techniques. The success of these algorithms depends upon the background and plate colours and textures. These are largely in the control of a particular institution and could be chosen to enable automatic separation

of foreground and background. Ideally, contrasting, uniform, matt fixed colours would be used for the table surface and the plate. White plates are very common but could lead to problems with the identification of white foods such as boiled rice and eggs. The first two techniques showed good results. For this project, it is assumed that the image background is one colour and that the Colour Thresholding technique will be used for foreground.

The second stage uses a K-means clustering algorithm to group parts of the image into coherent regions, each assumed to be a food type. In many cases, a human would struggle to distinguish a chicken nugget from a deep fried mushroom. A major disadvantage of K-means is that the number of clusters (foods) in the image needs to be set before segmentation. Three different approaches to automatically estimate the number of foods were tested. None were considered sufficiently accurate across the possible range of food images. A new method, based on training a model that can categorise the image into clusters of the same K, has been developed. The method has been tested with 445 images and yielded 97% accuracy. The K-means++ algorithm was found to yield the best results on databases of food images.

The third stage was to identify the foods within each segment of the image. Machine Learning was used to achieve this. A subset of a food image dataset was used to train the classifier and the rest was used to validate its performance. The results show very good performance in stages one and two. The third stage has also performed well, but is dependent upon the range and types of foods available within a particular institution.

A new method to estimate food weight by calculating food surface area has been proposed. The developed algorithm can calculate food surface area and estimate food weight using a specific table, which matches food surface to food weight. In order to

estimate food intake, two images of a plate of food are captured, before and after eating. The difference between the estimated food amounts is the food intake estimate.

The very large costs associated with manual monitoring of food intake make it practical for an institution to invest significant amounts in training a classifier with the selection of foods that it offers.

## **7.5 Future Work**

The development of the food recognition algorithm was a significant task due to the enormous range of foods available and the fundamental limitations imposed by the restriction to image data. Future work could focus on the hardware aspects of the project, i.e. the smart cart food delivery system. The cart needs to incorporate a food image acquisition system, designed in conjunction with the food recognition system, to acquire the food images before and after eating, and to assign these to a particular individual. The data networks present in institutions, such as hospitals and care homes, need to be considered when design the data flow. If WiFi is ubiquitous then the data processing can be performed remotely and reports of individual nutritional intake can be stored centrally. If networks are not present then the carts may need to store images for upload later or perform more analysis on the cart.

Once the hardware and software have been validated in conjunction, then a trial in an institution will be required to fully identify and quantify the costs and the benefits. Many of these benefits will come from individualised meals, so that people are served just the foods they eat, that benefit them, and potentially include functional foods/medicines. Such an integrated system would greatly reduce food waste and catering costs, but will also increase the health and quality of life of people within institutions.

## References

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), 2274-2282.
- Aizawa, K., Maruyama, Y., Li, H. & Morikawa, C. (2013) Food balance estimation by using personal dietary tendencies in a multimedia food log. *IEEE transactions on Multimedia*, 15 (8), 2176-2185.
- Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P. & Mouggiakakou, S. G. (2014) A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE Journal of Biomedical and Health Informatics*, 18 (4), 1261-1271.
- Arthur, D. & Vassilvitskii, S. (2007) K-means: The advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics.
- Atherton, T. J., & Kerbyson, D. J. (1999). Size invariant circle detection. *Image and Vision computing*, 17(11), 795-803.
- Bay, H., Tuytelaars, T. & Van Gool, L. (2006) SURF: Speeded up robust features. European conference on computer vision. Springer. Graz, AUSTRIA.
- Becker, B. C., & Ortiz, E. G. (2008, September). Evaluation of face recognition techniques for application to facebook. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on* (pp. 1-6). IEEE.
- Berger, C. (2014) From a competition for self-driving miniature cars to a standardized experimental platform: concept, models, architecture, and evaluation. arXiv preprint arXiv:1406.7768,.
- Bettadapura, V., Thomaz, E., Parnami, A., Abowd, G. D. & Essa, I. (2015) Leveraging context to support automated food recognition in restaurants. 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Beach, Hawaii, January 6-9, IEEE.
- Bossard, L., Guillaumin, M. & Van Gool, L. (2014) Food-101—mining discriminative components with random forests. In *Computer Vision—ECCV 2014*. Springer, 446-461.
- camranger (2017) Wireless camera control; Available online: <http://camranger.com/>.
- Cover, T., & Hart, P. (1967) Nearest neighbour pattern classification. *IEEE transactions on information theory*, 13(1), 21-27.
- Crankshaft's staff (2016) In Depth Tutorials and Information. Available online: <http://what-when-how.com/embedded-image-processing-on-the-tms320c6000-dsp/segmentation-image-processing-2016/>.

- David, J., Bajaj, S. & Jazra, C. (n.d) A Facebook profile-based TV recommender System. *Vectors*, 1 u2.
- Dean, J. (2017) Machine learning for systems and systems for machine learning. In Proc. of NIPS Workshop on ML Systems.
- Estrada, C. F., Jepson, A. & Fleet, D. (2004) Local features tutorial 2. - Citeseer
- Flegal, K. M., Carroll, M. D., Kit, B. K. & Ogden, C. L. (2012) Prevalence of obesity and trends in the distribution of body mass index among US adults, 1999-2010. *Jama*, 307 (5), 491-497.
- Fontana, J. M., Farooq, M. & Sazonov, E. (2014) Automatic ingestion monitor: A novel wearable device for monitoring of ingestive behavior. *IEEE Transactions on Biomedical Engineering*, 61 (6), 1772-1779.
- Graeme, W. (2010) 185,000 leave hospitals starved, *The Irish Sun*, London. Available online: <http://www.thesun.ie/irishsol/homepage/health/2819372/185000-leave-NHS-hospitals-starved.html> [Accessed 8/8/2016].
- Gariballa, S. E. & Forster, S. J. (2008) Dietary intake of older patients in hospital and at home: the validity of patient kept food diaries. *The Journal of Nutrition, Health and Aging*, 12 (2), 102-106.
- Gomez-Uribe, C. A. & Hunt, N. (2016) The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6 (4), p13.
- Green, S. & McDougall, T. (2002) Screening and assessing the nutritional status of older people: Malnutrition among older patients is a major problem both in hospitals and community settings. Sue Green and Tina McDougall outline methods of screening and assessment commonly used by nurses. *Nursing Older People*, 14 (6), 31-32.
- Griffiths, S. (2014) The 'microwave' that counts CALORIES: Device uses waves travelling through food to calculate its nutritional value, *dailymail newspaper*, Nov 23rd [Online]. Available at: <http://www.dailymail.co.uk/sciencetech/article-2684476/The-microwave-counts-CALORIES-Device-uses-waves-travelling-food-calculate-nutritional-value.html> (Accessed November 2017)
- Guzella, T. S. & Caminhas, W. M. (2009) A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 36 (7), 10206-10222.
- Han, J., Pei, J. & Kamber, M. (2011) *Data mining: concepts and techniques* [eBook] Elsevier.
- He, H., Kong, F. & Tan, J. (2016) DietCam: multiview food recognition using a multiKernel SVM. *IEEE Journal of Biomedical and health informatics*, 20 (3), 848-855.
- Hirschman, L., & Gaizauskas, R. (2001) Natural language question answering: the view from here. *Natural Language Engineering*, 7(4), 275-300.

- Hoashi, H., Joutou, T. & Yanai, K. (2010) Image recognition of 85 food categories by feature fusion. *2010 IEEE International Symposium on Multimedia (ISM)*, IEEE.
- Hsu, F. H. (1999) IBM's deep blue chess grandmaster chips. *IEEE Micro*, 19(2), 70-81.
- Jain, A. K. (2010) Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31 (8), 651-666.
- Joutou, T. & Yanai, K. (2009) A food image recognition system with multiple kernel learning. *2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE.
- Kawano, Y. & Yanai, K. (2015) Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, 74 (14), 5263-5287.
- Kitamura, K., Yamasaki, T. & Aizawa, K. (2009) FoodLog: Capture, analysis and retrieval of personal food images via web. *Proceedings of the ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities*. ACM.
- Lowe, D. G. (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2), 91-110.
- Machinery, C. (1950) Computing machinery and intelligence-AM Turing. *Mind*, 59(236), 433.
- MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, Oakland, CA, USA.
- Wilcox M. (2016) Harris Corner Detector; Scale Invariant Feature Transform (SIFT); <https://slideplayer.com/slide/8695490/>, [Accessed November 2017].
- Martin, C. K., Kaya, S. & Gunturk, B. K. (2009) Quantification of food intake using food image analysis. *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE. IEEE*.
- Daniel, M. (2010) Frail elderly patients 'left to starve in hospitals', admit more than two-thirds of NHS nurses; DAILY MAIL, [Online]. Available at: <http://www.dailymail.co.uk/health/article-1307250/Frail-elderly-patients-left-hungry-hospitals-admit-thirds-NHS-nurses.html>. (Accessed February 2017).
- Mathworks (2016a) Assess Classifier Performance in Classification Learner. Available online: <https://uk.mathworks.com/help/stats/assess-classifier-performance.html>. (Accessed May 2016).
- Mathworks (2016b) Choose Classifier Options. Available online: [https://uk.mathworks.com/help/stats/choose-a-classifier.html?s\\_tid=srchtitle](https://uk.mathworks.com/help/stats/choose-a-classifier.html?s_tid=srchtitle). (Accessed May 2016).
- Mathworks (2016c) How Machine Learning Works ; Available online: <https://uk.mathworks.com/discovery/machine-learning.html> . (Accessed March 2016).

- Mathworks (2018) Texture Analysis; Available online: <https://uk.mathworks.com/help/images/texture-analysis.html> . (Accessed February 2018).
- Mathworks, U. (2010) How do I convert my RGB image to grayscale without using the Image Processing Toolbox? Available online:<http://uk.mathworks.com/matlabcentral/answers/99136-how-do-i-convert-my-rgb-image-to-grayscale-without-using-the-image-processing-toolbox> [Accessed August 2010].
- Matsuda, Y., Hoashi, H. & Yanai, K. (2012) Recognition of multiple-food images by detecting candidate regions. *2012 IEEE International Conference on Multimedia and Expo. IEEE*.
- McCulloch, N., Bedworth, M., & Bridle, J. (1987) NETspeak—a re-implementation of NETtalk. *Computer Speech & Language*, 2(3-4), 289-302
- McDougall, T., Knight, S., Kirkwood, B. & Watson, R. (2008) Reliability of nurse assessment of malnutrition risk in hospital patients. *Journal of Clinical Nursing*, 17 (20), 2791-2792.
- Moravec, H. P. (1983) The Stanford Cart and the CMU rover. Carnegie-Mellon Univ Pittsburgh PA Robotics Inst
- Moynihan, P., Macdonald, A., Teal, G., Methven, L., Heaven, B. & Bamford, C. (2012) Extending an approach to hospital malnutrition to community care. *British Journal of Community Nursing*, 17 (12).
- National Patient Safety Agency. (NPSA), (2007), Protected Mealtimes Review. Findings and Recommendations Report. National Patient Safety Agency's (NPSA), Available online: <http://www.nrls.nhs.uk/resources/?entryid45=59806>, (Accessed December 2015).
- Oliver, A. (2008) Automatic dietary monitoring using on-body sensors: Detection of eating and drinking behaviour in healthy individuals. Doctor of Sciences. ETH Zurich, Switzerland.
- Padhraic (2015) Data Mining Lectures Lecture. Available online: <http://slideplayer.com/slide/5010193/> [Accessed September 20 2017].
- Perret. (2016) Here's how many digital photos will be taken in 2017, December 2 [Online]. Available at:<https://mylio.com/true-stories/tech-today/how-many-digital-photos-will-be-taken-2017-repost>.
- Pinge, J. & Nehemiah, A. (2017) Object Recognition: Deep Learning and Machine Learning for Computer Vision. Available online: <https://uk.mathworks.com/videos/object-recognition--deep-learning-and-machine-learning-for-compu-1482957345023.html>. (Accessed December 2017).



- Prabhu, P. (2015) K mean-clustering algorithm. Available online: <https://www.slideshare.net/parryprabhu/K-meanclustering-algorithm> [Accessed Sep/20 2017].
- Qureshi, S. (2005) *Embedded image processing on the TMS320C6000TM DSP: Examples in code composer studio TM and MATLAB* [eBook] Springer Science and Business Media.
- Pouladzadeh, P., Shirmohammadi, S. & Al-Maghrabi, R. (2014) Measuring calorie and nutrition from food image. *IEEE Transactions on Instrumentation and Measurement*, 63 (8), 1947-1956.
- Rosenblatt, F. (1957) The perception, perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory.
- Rosettacode, O. (2017) K-means++ clustering. Available online: [https://rosettacode.org/wiki/K-means%2B%2B\\_clustering](https://rosettacode.org/wiki/K-means%2B%2B_clustering) [Accessed October 2017].
- Rousseeuw, P. J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65.
- Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017) Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on* (pp. 3-18). Chicago IEEE.
- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).
- Utkarsh, S. (2010) SIFT: Theory and Practice. Ai shack. Available online: <http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/> [Accessed August/15 2016].
- Weinsier, R. L. & Heimburger, D. C. (1997) Distinguishing malnutrition from disease: the search goes on. *The American Journal of Clinical Nutrition*, 66 (5), 1063-1064.
- Yon, B. A., Johnson, R. K., Harvey-Berino, J. & Gold, B. C. (2006) The use of a personal digital assistant for dietary self-monitoring does not improve the validity of self-reports of energy intake. *Journal of the American Dietetic Association*, 106 (8), 1256-1259.
- Wollaston, V. (2013) Revealed, what happens in just ONE minute on the internet: 216,000 photos posted, 278,000 Tweets and 1.8m Facebook likes. Available online: <http://www.dailymail.co.uk/sciencetech/article-2381188/Revealed-happens-just-ONE-minute-internet-216-000-photos-posted-278-000-Tweets-1-8m-Facebook-likes.html> [Accessed Sept. 2017].

## Appendix A food record chart

**INPATIENT FOOD CHART**

Patients assessed as "low risk " of malnutrition MUST be rescreened and weighed every Tuesday & Saturday

NA

Please ✓ to indicate <b>AMOUNT EATEN</b> (if double portion eaten then double tick ✓✓)	DATE: / /					Total the number of ticks for each colour and circle the colour with the most ticks at the bottom of the page at the end of each day.			
						<b>RED</b>	<b>AMBER</b>	<b>GREEN</b>	
	Nil	¼	½	¾	All	SCORE '3' FOR APPETITE ON NST; SCORE '4' IF NBM	SCORE '2' FOR APPETITE ON NST	SCORE '1' FOR APPETITE ON NST	
<b>BREAKFAST</b>						<b>EATEN (additional information not essential)</b>			
Cereal / porridge & 1x toast									
Cooked Breakfast									
<b>SNACKS AM</b>						<b>EATEN (additional information not essential)</b>			
Packet biscuits / cake									
Cheese & crackers / yoghurt									
Other:									
<b>LUNCH</b>						<b>EATEN (additional information not essential)</b>			
Soup									
Main meal [hot]									
Sandwich / Salad									
Other:									
<b>DESSERT</b>						<b>EATEN (additional information not essential)</b>			
Pudding & custard									
Milk pudding									
Fruit and / or ice cream									
Yoghurt									
Cheese & crackers									
<b>SNACKS PM</b>						<b>EATEN (additional information not essential)</b>			
Packet biscuits / cake									
Cheese & crackers / yoghurt									
Other:									
<b>EVENING MEAL</b>						<b>EATEN (additional information not essential)</b>			
Soup									
Main meal [hot]									
Sandwich / Salad									
Other:									
<b>DESSERT</b>						<b>EATEN (additional information not essential)</b>			
Pudding & custard									
Milk pudding									
Fruit and / or ice cream									
Yoghurt									
Cheese & crackers									
<b>SUPPER</b>						<b>EATEN (additional information not essential)</b>			
Packet biscuits / cake									
Cheese & crackers / yoghurt									
Other:									
<b>MISCELLANEOUS</b>						<b>EATEN (additional information not essential)</b>			
Total FOOD ✓ for each column:									
SIGNATURE:						Tally the last 3 days RAG Score			
						Day1:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
						Day2:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
						Day3:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Escalation: If patient is scoring RED for three days or more inform Medical Staff and refer to the Dietitian. If AMBER follow moderate risk care plan

