DEPARTMENT OF COMPUTER SCIENCE

Food Recognition and Nutrition Measurement System

Zirong Weng

Supervisor: Dr. Sion Hannuna

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering.

May 10, 2019

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of BSc in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Zirong Weng, May 10, 2019

Abstract

Healthy eating is increasingly drawing public attention. Their awareness of the deleterious effects of obesity, which has increased rapidly in recent years, has fueled this attention. Research has found that a major cause of obesity relates to high daily intake of sweet food, such as candy and coke, furthermore, it has been shown that the high daily intake of sweet food also potentially risks dental problems. Obesity, diabetes and dental problems are not the only risks associated with poor diet. Heart diseases, including cardiovascular disease and even cancer have diet related risk factors. These problem impose a large economic burden on the individual, their families and the state. And we believe one aspect of solving this problem, is a food recognition and nutrition measurement system. An ideal system would allow a user to take s snapshot of meal and then be provided with a nutritional breakdown automatically. Consider that different people may have different calorie-conscious requirements in their daily life, this food recognition and nutrition measurement system would be of great help not only to dietitians and people with obesity, but also those having specific healthy eating or gym goals by keeping track of their diet information and providing them advice regarding their next meal according to their different aims using the photos of the food they eat, and through generated records of their eating habits.

This is the big dream, however, we only focus on the first step, segmentation and classification of 43 types of common food classes, containing a lot of varieties. We believe this is the vital foundation of the whole solution. Moreover, both classic computer vision and deep learning approaches are used for training classifiers. For classic computer vision approach, linear multi-class SVM is build, while for deep learning approach CNN is trained for classification. After that some food image datasets are used for testing the performance of both of them, comparing with each other and human-eye system. And the final results of the traditional computer vision has an accuracy about 42% with a liner multi-class SVM, while deep learning CNN classifier has an accuracy around 86.2% among these 43 food classes.

This project is mainly based on self-study deep learning followed by tutorials and internet resource for a BSc student without deep learning pre-knowledge. During this experiment, a lot hard-understanding concepts for CNN(Convolutional Neural Network) and classifier training were successfully implemented. And we managed to manipulate Oracle cloud server for sophisticated training process via Secure Shell (SSH).

Most of these significant challenges we posed were achieved smoothly, and there are brief summary of achievements listed in bullet points.

- Extracted region based feature like LBP(Local Binary Patterns) and Gabor response.
- Extracted pixel based texture features like HSV colour space.
- Applied PCA(Principal Component Analysis) to feature space for dimensionality reduction.
- Implemented K-Nearest Neighbors to test dataset.
- Trained linear multi-class SVM with 5-fold cross validation for classification, which works with 73 food classes with accuracy about 42.4%.
- Tested linear SVM classifier with original dataset having 43.94% accurate.
- Trained deep learning CNN(Convolutional Neural Networks) using Tensorflow Inception v3 architecture model.
- Manipulated Oracle cloud server via SSH for more effective deep learning training.

- CNN works with 43 food classes with accuracy about 86.2%.
- Testing with original dataset with 82.55% accurate.
- With an 72.72% accuracy for advertising image test images.
- With an 90.90% accuracy for Instagram image test images.
- With an 63.63% accuracy for edge-case image test images.
- Built rudimentary user interface website connecting to MongoDB database.
- Nutrition information stored in MongoDB.

Contents

1 Introduction **9**
1.1 Obesity, Diabetes and Dental Problems . . . 9
1.1.1 Worldwide problem . . . 9
1.1.2 Financial and social impacts . . . 9
1.2 Reasoning . . . 10
1.2.1 Existing methods . . . 10
1.3 New Solution System . . . 11

2 Contextual Background **13**
2.1 Towards Computer Vision . . . 13
2.2 Related Work . . . 13
2.3 Food Dataset . . . 14

3 Technical Background **17**
3.1 Feature Extraction . . . 17
3.1.1 Local binary pattern . . . 18
3.1.2 Gabor filter . . . 19
3.1.3 HSV colour space . . . 20
3.2 Classifiers . . . 21
3.2.1 Dimensionality reduction (PCA) . . . 22
3.2.2 KNN classification . . . 23
3.2.3 Support vector machine . . . 23
3.3 Convolutional Neural Networks . . . 25
3.3.1 Tensorflow . . . 26
3.3.2 Oracle cloud infrastructure . . . 28

4 Project Execution and Methodology **29**
4.1 Classic Computer Vision . . . 29
4.1.1 Feature Extraction . . . 29
4.1.2 Classifier Training . . . 31
4.2 Deep Learning Approach . . . 32
4.2.1 The Retrained Model . . . 32

5 Results and Critical Evaluation **35**
5.1 Original Dataset Testing . . . 35
5.1.1 Classic method results . . . 35
5.1.2 Deep learning approach results . . . 36
5.2 New Dataset Testing . . . 38
5.2.1 Commercial, Instagram and Edge-case Datasets . . . 38
5.2.2 Test Results and Evaluation . . . 39
5.2.3 Compare and Contrast . . . 42

6 Conclusion **45**
6.1 Summary of Achievements 45
6.1.1 Multi-class SVM 45
6.1.2 Deep Learning CNN 45
6.2 Future Focus 46
6.2.1 Improving classifier 46
6.2.2 Rudimentary Website 46

Chapter 1

Introduction

1.1 Obesity, Diabetes and Dental Problems

1.1.1 Worldwide problem

Healthy eating is increasingly drawing people's attention in recent years. The public's awareness of the deleterious effects of obesity, which has increased rapidly nowadays, has fueled this attention. This is a worldwide public health problem, no matter developing countries or developed countries. Back in 2009, The World Health Organisation (WHO) has already classified obesity as one of chronic disease[18]. There are some research showing that adults with obesity in England has risen from 13 % for men, meanwhile about 16% for women in 1993 to 24% in 2007[18]. In adults, obesity and overweight are found have close connect to increasing the risk of some diseases, such as heart disease, diabetes, cancers and other chronic conditions[20]. In terms of kids, obesity for children raised from 1995 to 2007 as well, to be more exact, from 11% in 1995 to 17% in 2007 among boys and from 12% in 1995 to 16% in 2007 among girls[18]. What is worse, obesity in kids could trigger serious problems for their life later on. According to Mindell[15], every 6 American children, there are one child is obese or potentially obese. Another evidence by the National Health and Nutrition Examination Surveys[20] states some stern reality that approximately 70% of adults are overweight or obese in general.

Besides, dental problems, such as toothache, cavities and sensitive to cold or hot food, are also increasing, especially for children and young women who are crazy about sweet food, which leads to a variety of oral health concerns. According to the National Eating Disorders Association[13], more than 10 million Americans currently are affected by dental problems. However, these oral concerns may further lead to serious eating disorders, such as anorexia or binge eating. Based on The Global Burden of Disease Study 2016 [16], oral diseases have been affecting more than 3.58 billion people worldwide. Moreover, there are 2.4 billion people suffering from caries of permanent teeth globally[17].

All in all, a lot studies are continuing to suggest that obesity, diabetes and dental problems are sever challenges for social life, having an effect on both adults and children worldwide. This is a serous problem for us. Moreover, not only just about American and England those developed countries, those problems are also increasing in other 'westernised' and some other less developed countries[15]. Diabetes and dental problems have also increased nowadays, which are common, serious and costly public health problems. Moreover, no matter developed countries or developing countries, they both face this health problem, from child to adult, from man to woman.

1.1.2 Financial and social impacts

The most obvious impacts of obesity, diabetes and dental problems are physical just as we talked before, affected people are more likely to have cavities, chronic disease and other illnesses than others. Additionally, the evidence provides encouragement for viewing the relationship between these problems and some financial and social impacts as directional[23].

The impact of obesity is not only on the individuals, but also on the family and on the countries. Take UK for example, the damage of economy cost is more than £3.5 billion every year, contains about 18 million working days lost through sickness[10]. Obese patients would need special nursing, so government

keep investing a lot of money for training these nurses. Further, families also suffer a lot of the financial burn of these treatments and medication.

Moreover, there are also passive social implications. Obese individuals, especially women, are found[4] more likely to be stigmatized. Moreover, this may affect their education status, and even marriage, due to less confident. Further, women with obesity were found[5] to complete less years of education, have lower probability to get involved in marriage, and have lower salary compared with others. In terms of obese kids, when they are bullied or victimized due to their obesity situation, it could bring about feelings of shame, which may lead to mental problems such as depression, or even the idea of suicide.

To sum up, obesity, diabetes and dental problems not only are major risk causes for these chronic diseases we talked above, but also impose a large economic burden on the individual, and on families and nations. Furthermore, those economic burden and working days lost would become an obstacle for socioeconomic progress.

1.2 Reasoning

Individuals should try to reduce this worldwide problem themselves, but also, it is the duty of government and society to reduce the social consequences of obesity. Thus, people need to know the causes of obesity so they could prevent it in early stage. Some researches found that major causes of obesity has close relationship with the high daily intake of sweet food, such as candy and coke. An important intervention of those problem that we should pay more attention would be prevention.

Most of the time, people have no ware of what they are eating especially when they eating out. The number of people eating at home is declining nowadays due to the fast speed life. Plus, many people are eating out more often, and this reduces the times they cook at home by themselves. This common habit can not be shocking, and this could easily increase the probability of obesity or other disease, because home made food is usually much healthier. More importantly, people have knowledge of what they are going to eat if they prepared it. In other words, people eat fewer calories when cooking at home because they are ware of it while cooking.

1.2.1 Existing methods

In order to solve these health issues, there are a lot different methods has been introduced, such as psychological strategies. For instance, there are many weight loss programs, which includes a lot of gym training for fat-burning and maybe some diet plans. A number of these programs are advertised in magazines and newspapers, as well as on TV, and internet. However, it is hard to verify these kind of programs, especially some commercial programs. Nevertheless those programs have based on scientific and medical basics, there is a high probability that these are often not explicit due to different personal conditions. For example, it is impossible to train elder participants under a strenuous exercise, although it is helpful with the weight loss. These treatment need to consider the personal conditions and social status. Some evidences[25] support that major commercial weight loss program in America is substandard.

Moreover, some of the programs may ask participants to take weight loss pills, which could be very expensive to continue. We are trying to find a better way decrease this financial burden rather than increase it. Alternatively, people could follow the advice by the health care professionals from NHS. Although this would be the suitable and medical based approach for each individual, it is uneasy to talk about personal weight and it is a considerable burden on the NHS. People need to bring their questions with the health care professionals and practice talking about personal concerns and participants need to take record of the conditions by themselves. Health care professional may give some suggestions such as healthy eating, physical activity, and weight during general office visits, but it is hard to continue the visit regularly for most of people. In general, people are having difficulties for measuring the amount of food or nutrition intake, due to lack of nutritional information or can not recording for long time. But it is not an easy task for human eyes.

Although there are many ways to work out the problem, a more time saved, less financial burden and better measured solution still should be considered. Under the condition that people are having difficulties to measure the nutrition for their food, if there is a simple way can measure it for us by processing the image of the plate, this can help people avoid eating unnecessary or unhealthy food and give detailed calorific and nutritional information, for instance is it belongs to 5-a-day compliance rolling

calorie. Since human eyes cannot recognise or measure these information especially when eating in restaurants and home making meals. And people can use this system whenever they want, rather than booking a time slop with their health care professionals. It is easy for people to visualise and record the nutrition information.

1.3 New Solution System

There is a part of the solution we believe, which is the food recognition and nutrition measurement system. Considering different people may have different calorie-conscious requires in their daily life, this food recognition and nutrition measurement system would help not only obese persons, but also people who having diet goals. Our proposed system keeps tracking users' diet information and giving them some advises to their next meal, according to their different aim by photos of the food them eat, then make records of their eating habits if users continue uploading the images of food that they ate. Moreover, this system can recognize food and calculate calorie information by simply upload the picture of food and can be used in dietary monitoring applications. In comparison with manually recording and calculation by hand, people are easier to manage daily food intake or gym goal with our system. This proposed system considers colour, size, shape and texture and combine all different features in order to best classify the food. After recognized food type, the calorie and nutrition information will be calculated according to food nutrition table. This is convenient for every person, the only thing they need to do is taking a picture of what they eat. Compared with the traditional the advice by the health care professionals from NHS, this is more quick way and easier to maintain and continue.

The whole system would be able to extract the food from the image of the plate, then according to the nutrition table give the user information of the corresponding food and the nutrition, such as 5-a-day and calorie content. More importantly, a medium term log would be created to make record of the user, so any person who uses this system are able to know if he/she is over-eating or under-eating. Also, vitamins, such as vitamin C and vitamin E, which are important to support people well-being would be recorded and calculated based on different food. Based on a lot of researches, there are close connection between poor diet and health problems. A good system should be able to give all these information to the user, reminding user the intake of a variety of nutrition. More importantly, a solid foundation of this system is to successfully segment the food region of the input image and recognise the food itself. So our main focus of the whole food recognition and nutrition measurement system is food segmentation and classification.

Chapter 2

Contextual Background

This chapter covers relevant fundamental aspects of computer vision context of food recognition tasks, then discusses some other related end-to-end food recolonization systems. Then considers some inspirations from related work for both the food classification aspects of the project and also related diet management system. The dataset we are going to use for training is also shown with some image examples. However, the specific methodologies and algorithms used in our experiment will be discussed separately in Chapter 3 Technical Background, while this chapter is trying to give more intuitive ideas of food recolonization as computer vision tasks.

2.1 Towards Computer Vision

For computer, no matter how colourful an image is, it is still made up of with a bunch of numerical values and nothing else. Computer vision tasks always attempt bridging this kind of semantic gap in order to produce real numerical or symbolic information underneath[24]. Food recognition is in many methods a classical example of a computer vision problem: contains of feature extraction and classier training for food only, maybe with elements of prepossessing involved. As we may known, computer vision's overarching aim is how to gain semantic understanding like human brains. And this is exactly what we want in terms of our proposed system, for food especially.

Computer vision has used rapidly in the food industry for many aspects, such as inspection, evaluation and objective assessment[10]. To be more exact, some food segmentation tasks like segmenting sushi, noodles, pizza, or orange have been using computer vision systems. However, with increased expectations for food to be in higher quality, more accurate recognition needs and faster process, there are more and more requirements [1].

In order to solve these problems, we focus on the food segmentation and classification training, the very first but important step of the purposed system, and try to make some progress for that. More intuitively, the same food category may also has the similar pixel information. And our job is trying to extract as many features as we can to train a better performed food classifier.

2.2 Related Work

Our main focus is segmentation and classification. We would like to know how other people solve similar tasks. Here, we are going to talk about some related work, particularly their recognition strategies and classification methods.

In terms of food features, the work of He[11] describes the food image by both local and global features. Classifying food in images may contain huge challenges, because we know how similar food can be. The example of the large visual similarity between food would be that a brownie could be very similar to a small piece of chocolate cake[28]. According to He[11], the first step is segmenting image for food item based on the food region boundaries. Since there may be other thing in the image but we are not interested about them, features, such as colour and texture, can be extracted from that region only[11]. This gives us some intuitive ideas of how to segment food from beginning for our own experiment.

Furthermore, the work of Farinella[6] gives some ideas of how to extract different features. Farinella uses a vocabulary built on textons(linear filter bank) for the high variability of food image. This linear filter bank contains different rotation and scale, applying to food image. This filter can be applied to different food class. Then the learned class-based linear filter bank is collected in a single visual dictionary[6]. Following these steps, the food images could be represented by visual dictionary distributions and a Support Vector Machine(SVM) for further use.

Another aspect of this literature basically focus on what is the current diet monitoring system in real life, and how it is designed. Shroff[19] proposed a human computer interaction system to recognise food class, estimate calorie and keep monitoring. Although nutrition information will be provided to users while it is very expensive to keep updating with the current vision sensor, since this asks users to wear interactive devices, and keep mobile platform updated with a growing number of sensors and capabilities, such as GPS and 5.0 megapixel camera sensors[19]. However, users may not be able to ware the devices all the time, especially, users want a more cheaper and flexible way.

Summary

The related research work in this literature have often focused on real food recognition problems, such as extraction feature strategies and classification methodologies. Importantly, this gives us some inspire of our own system. For instance, the way of extracting local features within the segment of food images, and the benefits of using a filter bank for the image content. Moreover, some current vision of diet monitoring software products also encourage us to make something different, cheaper, easier-used and maybe more accurate food recognition and nutrition measurement system.

All in all, our purposed system would try to use these methods for food segmentation and classification. What is more, we are going to combine these different features and try different methods to make progress for food classification problems. Inspired by those software products above, we would make effort to our proposed system. But the main focus would still be the food segmentation and classification. Plus, some changes are made to improve the performance and other extensions are also made to reach more advantage aims.

In the following section, we are going to talk about datasets for our experiment.

2.3 Food Dataset

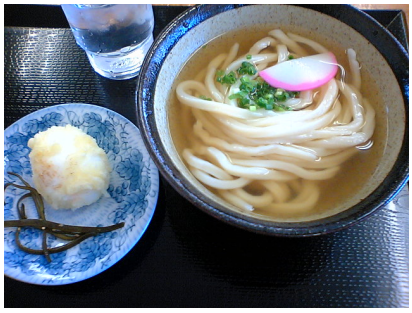
The first data set of food images provided by Gianluigi, C[2] is used to for different feature extraction and classification algorithms. This dataset is called UNIMIB2016 Food Database. And this is the dataset provide by the original paper we discussed before. This dataset is composed of 1027 tray images with multiple food containing 73 food classes in total, and most of the food are popular in western countries, to be more exact, Italy.

Another food dataset we are using is called UEC FOOD 100, and most of the food categories in this dataset are popular food in Asian, to be more exact, Japan. We know Asian food is more likely small pieces and mixed with each other, while western food are separately in different plates. We wish our proposed system can perform well in both situations. However, due to the differences between those two datasets (image quality, number of images of each food, etc.), only 43 classes are chosen for training. To this end, the whole data set was separated into two, about 65% of the images are using as our training dataset, and around 35% as our testing dataset in order to avoid overfitting problem. And some random pictures used for testing as well.

(a) UNIMIB2016 Example 1

(b) UNIMIB2016 Example 2

(c) UEC100 Example Sushi



(d) UEC100 Examples Udon

Source: Dataset from UNIMIB2016 and UEC100.
Figure 2.1: Datasets Examples

As can be seen from figure 2.1, there are more than one class of food in some images. Moreover, this suits the real world condition. For instance, Asian food are more likely placed in different small plates or blows with multiple different types food inside. Sometimes a lot of them are mixed together. The particularities of the UNIMIB2016 dataset are that each image displays food on a tray, however, some food such as oranges and yogurt are placed on the place-mats instead of on the plates. Moreover, the the place-mat is always similar white in this data set, which could make it more difficult when doing the location and ROI(region of interest) segmentation. These characteristics talked above make datasets more challenged to process.

Chapter 3

Technical Background

This chapter discusses the rational behind the feature extraction methods, classifier and techniques given the food context. In order to classify food images, we need to segment the image into food ROI(region of interests). Once isolated, specific food features can be extracted from them.

There are a lot different features to extract, such as colour and texture. We are mainly focus on region based LBP(Local binary pattern), Gabor responses and pixels based feature HSV colour space. Both LBP and Gabor filter are strong texture extractors, while HSV contains colour information. Moreover, two popular of different classifier training methods, SVM(Support Vector Machine) and KNN(K Nearest Neighbor) are also be discussed in this chapter.

Meanwhile, CNN(Convolutional Neural Networks) is a deep learning way of recognizing patterns with a large training dataset and it more often gives excellent performance. Tensorflow is a open source library to build CNN with more flexible control.

3.1 Feature Extraction

Texture features are one of the most popular image representations for computer vision tasks, such as template matching and image classification. Moreover, real world images are almost filled with colour and textures, especially food images.

People like to take pictures of good-looking food. Furthermore, texture-based segmentation methods employed in computer vision has seen various degrees of success in many ways with some claimed to be even better than human vision system. In general, food texture represents by some local attribute values distributed over a certain region, especially the region of interest(ROI). An image probably contains millions of pixels but only a few regions, and these regions are the interesting parts drawing our attention. In our case, we only need the food part of the image, and we wish to ignore other parts, such as plates, or tables. The region of interest(ROI) is the bounding box of the food area based on the ground truth. Before we extract any features of an food image, we want to crop the ROI first.

Texture is an important attribute of food image. In the following sections, we are going to discuss some approaches for analyzing and extracting food texture from an image.

Figure 3.1: Pipeline of feature extraction

3.1.1 Local binary pattern

The very first thing we want to talk about is local binary patterns (LBP). LBP are a strong visual descriptor, performing good at texture extraction, which have been used in computer vision tasks for a long time. They are popular in both traditional computer vision methods and deep learning training. In general, original LBP operator always represents the values of the pixels of an image with decimal numbers encoding the local structure around each pixel[3]. Then by comparing with the 8 neighbors, each neighbor pixel will have value 0, if that neighbor is smaller than central value. Otherwise if it is larger than central value, that neighbor will have value 1. This is how LBP map is constructed.

Figure 3.2 displays a simple example of the LBP, where that LBP code of the center pixel in the pizza image is obtained by comparing its intensity with its 8 neighbour pixels. Similarly, we can generate a row of binary number from LBP by reading 8 neighbors of each given central pixel in a clockwise direction, as shown in the figure. Furthermore, if we change that binary number back to decimal, we will get the new value of the central pixel. This is basically thresolding the window with the center pixel value. This encodes local contrast and pattern making it highly discriminative. Also, it is easy to compute LBP.

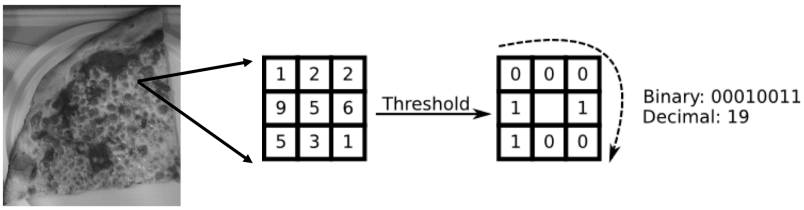


Figure 3.2: Grayscale pizza example of LBP

In the pizza example above, a histogram of the LBP responses can be generated using the binary sequences. This histogram shown in Figure 3.3, can be seen as a feature vector. This could be useful for food detection, because for the same type of food, they will have the similar texture and similar histograms, then the threshold for particular class could be found for segmentation. This is powerful feature classification and it used for both traditional computer vision tasks and deep learning models. Traditional computer vision approach consists of feature extraction then classifier training, while deep learning method involves automatically feature extraction and classification based on a huge training datasets. We will use LBP to train our traditional food classification model first.

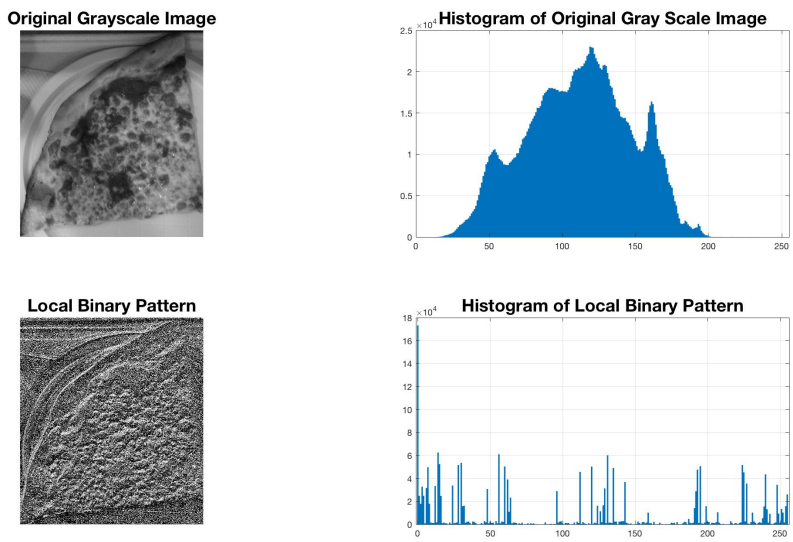


Figure 3.3: Example LBP and histogram

3.1.2 Gabor filter

Another powerful texture feature extraction tool is called Gabor filter. Unlike LBP, Gabor filter is using the spectral approach[9] for texture analysis. To to more exact, the responses of Gabor filter represent texture information. Due to the multi-channel filtering approach[9] of Gabor filter, especially when using a Gabor filter bank, each response will be selective in different spatial frequency and orientation. Intuitively, Gabor filter seems to work with segment different food classes, while the simple instance is that rice would have higher frequency than chickpea. And based on a lot literature work, Gabor filter is a popular method to extract food texture information.

In terms of the theorem behind Gabor filter, like Fourier transform, Gabor filter is based on the spectral approach. When Gabor filter is applied, the input image will be decomposed with a sinusoidal plane wave of certain orientation and wavelength. And this response encodes local texture information in the frequency domain. Normally, more than one orientation and wavelength are using, because usually we want several Gabor responses according to all the unique combinations of wavelength and orientation that we specified, instead of only one response. It is highly suitable for our food detection demand, since different food may have different texture, for instance, cake is soft while potato with skins looks rough.

Gabor filter is similar to Fourier analysis, where they are both sine and cosine functions modulated by a Gaussian kernel window sliding inside the input image[12]. Hence, a Gabor filter bank contains Gaussian kernel functions multiplied by a complex exponential with various scales and rotations. In the complex space these filters are defined as follows.

$$G(x, y, k_x, k_y) = e^{(-(x-X)^2+(x-Y)^2)/2\delta^2} \cdot e^j(xk_x, yk_y) \tag{3.1}$$

In this equation, x, y represent the spatial domain while k_x, k_y represent the frequency domain. X and Y are the spatial localisations of the Gaussian kernel window. The filter's selectivity in different spatial frequency and orientation is given by

$$\theta = \arctan(k_x/k_y) \tag{3.2}$$

Moreover, by using the Discrete Fourier Transform in equation (5) above, the filter's central frequency and orientation become as following,

$$\omega = 2\pi\sqrt{(\frac{k}{N+1})^2 + (\frac{l}{M+1})^2} \quad \text{and} \quad \theta = \arctan\frac{k}{N+1}\frac{M+1}{l} \tag{3.3}$$

Therefore, Gabor filter is very useful for extract information of spatial or frequency simultaneously from textured images. based on studies, there are some similarity between human visual system and the Gabor filter in terms of texture analysis. For food images, different food class may have different Gabor response due to they have different shape or different texture. Gabor filter bank could detect that for us. However, in our food classification scenario, the orientation is not that important like in other situations, due to we want to recognise food in all the orientation, no matter what orientation people put them in the plates. Moreover, we do care about phase frequencies. In other word, phase frequencies is a vital feature to distinguish food classes. For example, peas have higher phase frequencies because they are smaller pieces , while bread peas may have lower phase frequencies. To visualise this, there are some examples for using Gabor filter bank with different wavelength but the same orientation to extract the features of the region of the green area food as follows. The original food picture is shown in Figure 3.4, while the Gabor features with different wavelength can be seen in Figure 3.5.

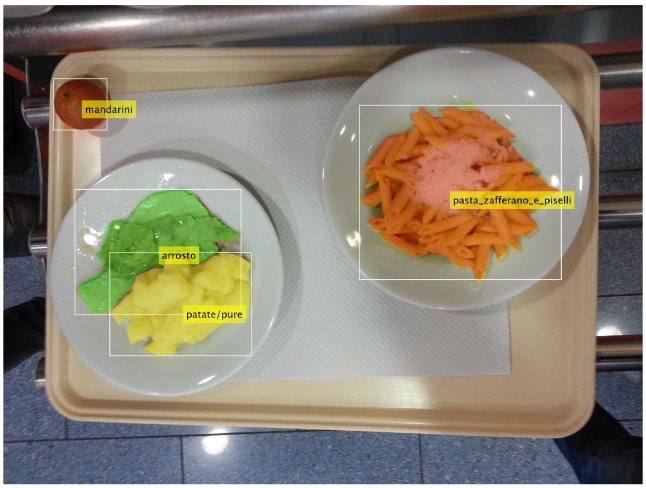


Figure 3.4: Using Gabor filter on the roast region(green)

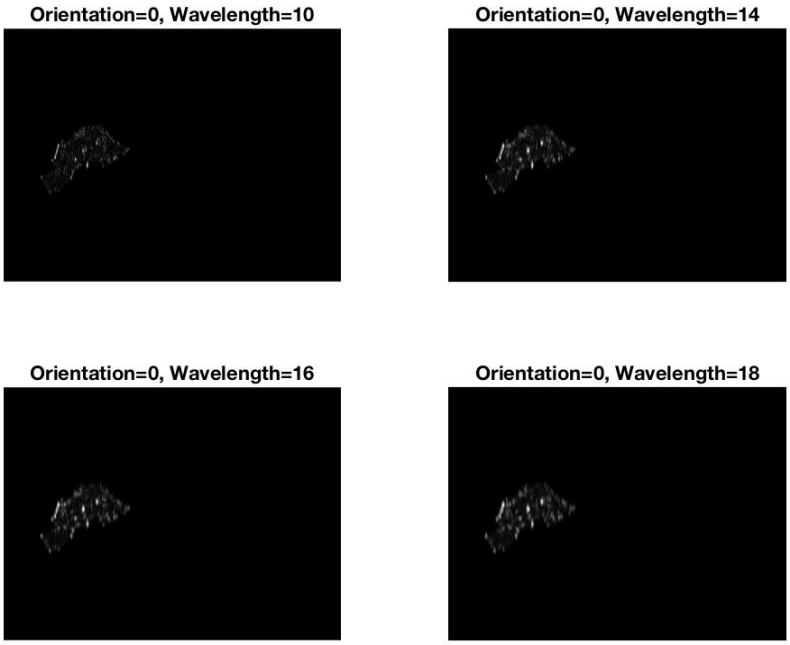


Figure 3.5: Gabor response with different wavelength

3.1.3 HSV colour space

Colour also plays an important part in the real world food recognition, since the same food material may always show the same colour. In automated image processing, colour is a powerful descriptor that can help people simplify object identification and extract feature from an image. Colour can help with identity food source and improve the accuracy of food classification. Although texture performs attractively for food pattern recognition, colour may give us extra information. Since both LBP and Gabor filter bank only takes grayscale as input, we might lose information about colour in the image. Furthermore, we need colour information combined with texture information to classify food.

RGB and HSV are main ones used for colour space. However, RGB is very sensitive to lighting. To be more exact, the three channels, R, G and B would increase if there were brighter light in the image even though they are the same food. Even worse, RGB channel depends on camera a lot. Even though there are the same pizza, the RGB of the images from Instagram or commercial advertisements could be more different from real world pizza images because people manual adjust the colour values. For example, they may prepare images using Photoshop with different RGB color profiles settings, such as changing lighting. Therefore, we are using HSV colour space to avoid such problems. HSV colour space separates lighting into intensity, known as V (values). When we deal with it, we can ignore the lighting(intensity), only take H(hue) and S(saturation) into consideration while extracting colour features. The example is as follows in Figure 3.6, and we can see from the output, H and S are better for masking the pizza region. So we can get better segmentation from H and S features.

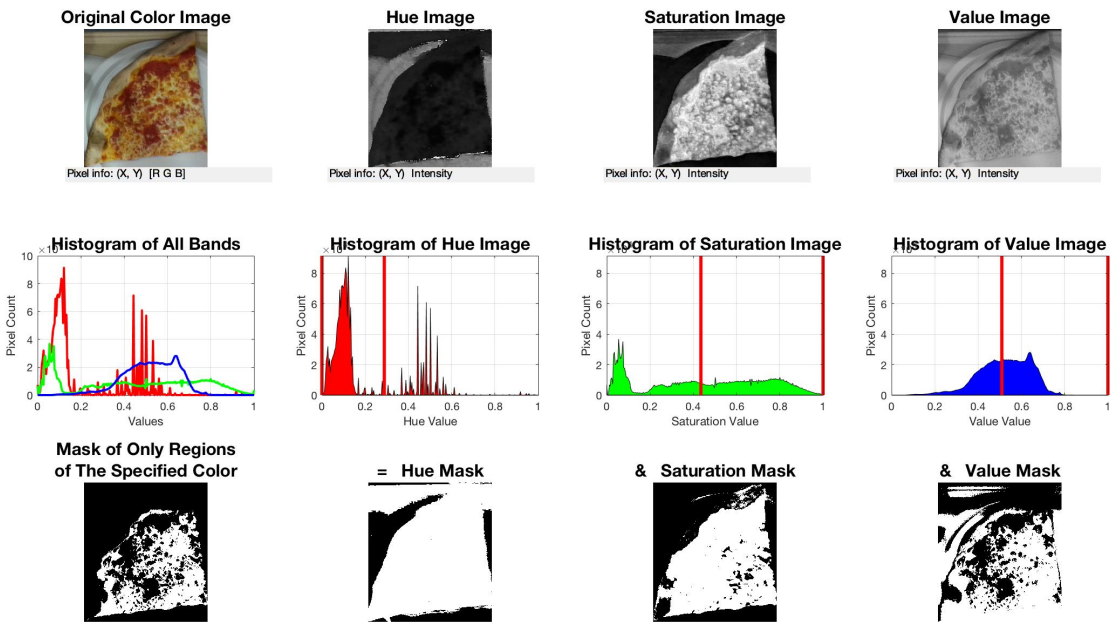


Figure 3.6: HSV of pizza mask

Moreover, HSV colour space is similar to how human vision system process colour comparing with RGB space, which give us better understanding of the colour extraction with HSV. The HSV colour space avoid mixing distinct colours. This is also a powerful extractor when dealing with object classification, especially food. There is an example of using HSV mask to extract pizza from the data set used to classify as above.

3.2 Classifiers

When we have enough features, we can consider to train our classifiers. Classification has been used for identifying the categories of a new observation based on some training knowledge containing other known category observations. Satisfying classification should identify correct categories for new observation. For instance, after learning enough spam and not spam emails, a spam detection would identity next new email to either spam or not spam emails category. Likewise, our aim is to predict the food class of a new image by using the features we extracted above involving more than two food classes. So we are trying to solve multiclass classification. Despite of the number of classes, there are many classifiers for different classification problems. For instance, we could use feature vector to describe and measure each instance.

Then apply linear classification to it by calculating a score for each potential observation using feature vectors with certain weights.

In our food classification case, we are mainly using Support Vector Machine(SVM) and neural network to train our classifier, because they are widely used and perform better results. SVM is used in traditional computer vision training process, while deep learning uses very large neural network.

However, in order to understand how well separated the classes are for features extracted, some other classification also happened, such as the k-nearest neighbors algorithm (KNN). The whole training pipeline is shown as in Figure 3.7. In the following sections, we are going to put more details of each operation during the classification.

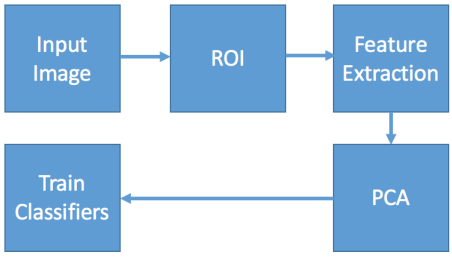
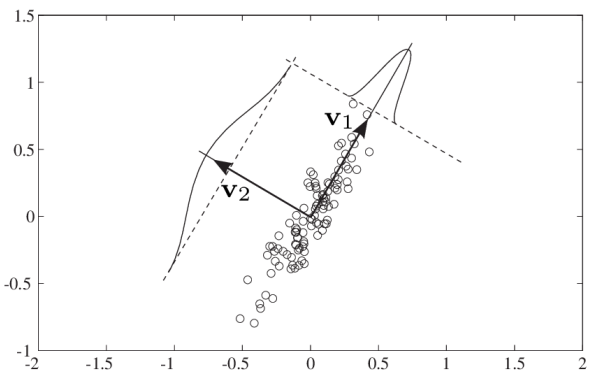


Figure 3.7: Pipeline of training

3.2.1 Dimensionality reduction (PCA)

Another thing before classification and after feature exaction people may apply is Principal component analysis. This is basically a mathematical method widely used for dimensionality reduction, especially for high dimensional dataset. PCA can reveal relationships between high dimensional data points by using the spectral decomposition of a correlation coefficient or covariance matrix[8]. In general, PCA converts multiple indicators into a few comprehensive indicators, revealing the linear relationship between variables. There is no doubt that our food images have multiple features. Some of them may have stronger relationship, while some of the features may offer duplicate information. Without PCA, the model would be more complex, even worse, it could give more errors, which will lower accuracy.

PCA is one of many dimension reduction techniques. There are other methods both from statistics theory[21], such as vector quantization and mixture models, and based on the decomposition of a matrix formed, such as non-negative matrix factorization (NMF). However, PCA is one of the most popular and easy implemented methods to efficiently decrease highly redundant data to a smaller number of variables. Moreover, there are PCA derived approach, like nonlinear PCA, PCA rotations and sparse PCA for other different data situations.



Source: image from researchgate.net
Figure 3.8: Simple PCA example

To be more exact, PCA tries to replace the original feature space by recombining them into a new

set of independent variables. Mathematically, the original features will be replaced by a new linear combination. The combined dataset that contribute most to its variance will be retained, while less contributed will be discarded. A simple example is shown in Figure 3.8 that two vectors V_1, V_2 are the eigenvectors of the covariance matrix. PCA can be very useful to process high-dimensional datasets.

3.2.2 KNN classification

After the dimension reduction of the features, we can now train our classifiers. Here we used KNN classification on each pixel. K-Nearest Neighbor is an algorithm to assign a new unknown observation to the same category as its closest instances. Intuitively, this pixel should be in the class of mashed potatoes, if all its closest neighbors are in the class of mashed potatoes. This concept do make sense in food classification, because the center pixel is more likely to be in the same class as its neighbor pixels. Hence, it is sensitive to the structure of the dataset.

This is a simple and lazy-learning algorithm, so it may not suitable for complex tasks. However, it gives some information about the dataset. The reason why we still consider to use KNN is that we want to know if the dataset used for classification will be separated into several classes. Hence we can further predict for new observation. In our case, we want to using KNN classification to classify food categories, which are separated from each food class. Another advantage of KNN classification is that it does not require any assumptions on the underline data distribution, which matches the real world scenario knowing little prior knowledge about the data. The process is assigning each pixel element to the most similar labeled group and each group would be a category. To qualify the similarity, we use the Euclidean distance between two points, described as the following:

$$d(x_i, y_i) = \sqrt{\sum_{i=1} (x_i - y_i)^2} \tag{3.4}$$

on the other hand, there are some alternatively ways to measure the distance, such as Manhattan distance shown in equation (3.5) and Minkowski distance shown in equation (3.6).

$$d(x_i, y_i) = \sum_{i=1}^{k} |x_i - y_i| \tag{3.5}$$

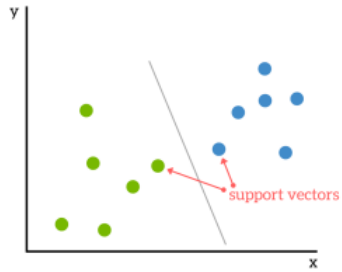
$$d(x_i, y_i) = (\sum_{i=1}^{k} |x_i - y_i|^q)^{\frac{1}{q}} \tag{3.6}$$

KNN classification is a good way of knowing how good features are, although the unknown of k would be a problem if we using it as unsupervised learning. That is the reason why we still use KNN classification, but only for the checking the features instead of training a KNN to predict food class. Because if we are going to predict food class by using KNN, the K would not be known in advance, otherwise it would be cheating.

3.2.3 Support vector machine

A multi-class support vector machine(SVM) needs to be trained to classify different food textures. An SVM, using a linear kernel, is an effective machine for our classification problem. This is a classic computer vision approach. SVM is a widely used classifier normally used for data classification containing training data then testing (supervised approach). The main challenge is how to train an effective SVM model to predict correct food class when given new images. We are going to use a linear multi-class SVM, which is a SVM that scales linearly[26]. In order to meet the demand of high accuracy, a lot labeled training data are needed. Meanwhile, the linear multi-class SVM will give an optimal hyperplane which labels new item.

The simplest way is shown in Figure 3.9, two dimensional space for easier visualization, and the hyperplane would simply be a straight line for 2D space. As shown in the figure, that line divides the plane into two parts, successfully separating green class and blue class. Basically, this is what we want SVM does for food classification, to separate each food textures. However, in more complex scenario, there will be three dimensional space, then the hyperplane separating classes would be a 2D plane.



Source: image from https://medium.com/machine-learning-1012
Figure 3.9: Simple SVM example

In our scenario, the space is more than 3 dimensions, and things are getting more complex. For instance, unlike in Figure 3.9, there would be more than 2 classes needs to be classified, since we are using a dataset with many classes. For each food class, there are more than 2 features representing it. Hence, SVM finds out a hyperplane in N dimensional space, where N is the number of features describing each class, just like a straight line in the simple 2 dimensional example by using one vs. one classifier strategy[14].

Kernels

One parameter of SVM is the kernel functions. In general, it converts not separable dataset into separable problem by data transformations. There are many different kernel functions according to different demand, such as linear kernel, polynomial kernel, Gaussian kernel, etc. Especially, we use linear SVM for classification since our data is linearly separable. Hence, the kernel we are using is a linear kernel.

For a linear kernel the equation is given using the dot product between the input (x) and each support vector y with an optional constant $C(0)$ as follows:

$$k(x, y) = C(0) + x^T y \tag{3.7}$$

This is a common kernel to be used when there are a large number of features in a linearly separable dataset.

Regularization

Usually, there are more than one way to draw decision boundaries. Like in Figure 3.9, alternative lines could also be draw. However, there are some trade off need to be considered by regularization.

(a) Low regularization value (b) High regularization value

Source: image from https://medium.com/machine-learning-1012
Figure 3.10: Regularization example

In general, regularization helps with controlling the level of mis-classifying. For example, in Figure 3.10, both (a) and (b) are separating hyperplane for black and blue points, while (a) seems to have 2 mis-classifying with lower regularization value. In (b) we got potentially all of the data points correct, however this may cause overfitting later on. We are always trying to balance this trade off with different regularization value in order to get a balanced decision boundary, meanwhile avoid over fitting.

3.3 Convolutional Neural Networks

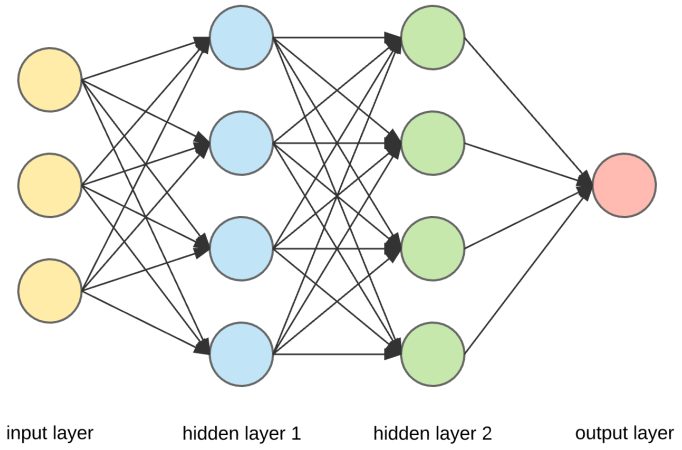
There are some modern deep learning ways to deal with this recognition problem, instead of extracting features by ourselves then feed those features to build own classifier. Artificial neural network are mainly used for deep learning models for neuron science, which is a set of units connecting with each other with different weight associated. Artificial neural network learns while training, it learns every time after adjusting different weights to predict the correct prediction.

A lot network architecture model are poplar used for image classification, object recognition and action recognition nowadays, such as convolutional and recurrent. Although there are some disadvantage sometimes of the unknown semantic meaning behind the learned weights, it has been performed impressively most of the time. It offers huge benefits like high tolerance to noisy input and satisfying performance of untrained patterns. Convolutional neural networks perform better for task like extracting feature to building classifier by itself, which means during the CNN training, good features are extracted and classifier are built together.

Convolutional neural networks(CNN) have been widely used for a lot of large-scaled image classification and recognition challenges. CNN offers huge benefits that each layer of the network transforms its input data into a slightly more abstract and composite representation then send it to next layer. As for hidden layers, there can be as many as you want if that task is complex. And the hidden layer of the convolutional neural network includes three common constructions of convolutional layer, pooled layer and fully connected layer. In short, convolutional layer is for extracting features from the input data, after that, the output feature map is passed to the pooling layer for feature selection and information filtering.

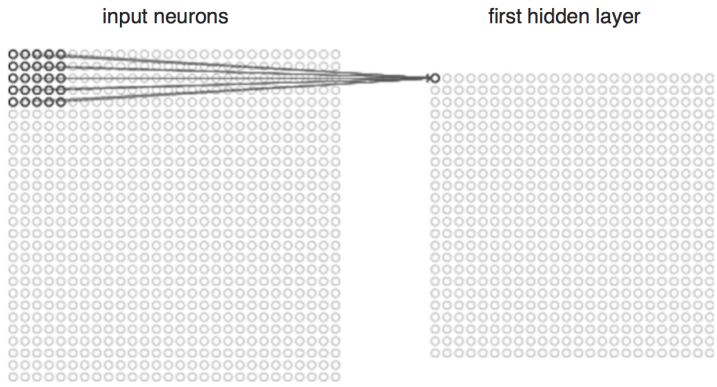
In a sushi image recognition task, after reading the input image, the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode semantic meaning; and the fourth layer may recognize that the image contains rice. Importantly, a deep learning process can learn which features to optimally place in which level on its own. Finally, when you have seaweed upon the rice, you got sushi.

Neural network training contains supervised learning. We should carry this task by labeling the images, the computer will start recognizing patterns present in cat pictures that are absent from other ones and will start building its own cognition. A neural network model that works in a similar way to the neurons in the human brain. Each neuron takes an input, performs some operations then passes the output to the following neuron. In Figure 3.11, each colour indicates one different layer, and each input passes output to next layer cell.



Source: image from medium.com
Figure 3.11: Neural network example

To be more exact what happens between layers, the convolution layer is always the first. After reading the input image as a matrix with pixel values, then it produces convolution by moving along the input image like a window slide in order to convoluted its values by the original values. After this is finished, all these values are summed up into one number obtained in the end. Then the slide window should move one further unit to next position repeating the same operation until it finishes the whole image with a smaller outcome matrix. The example of this convolution is shown in Figure 3.12.



Source: image from medium.com
Figure 3.12: Convolution

All in all, When the food image passes through one convolution layer, convolution described before happens. And the outcome of that matrix will become the input for next layer to operate on. This is repeating for each layer until it couplets the whole network results in an N dimensional vector. And N is the number of the food classes.

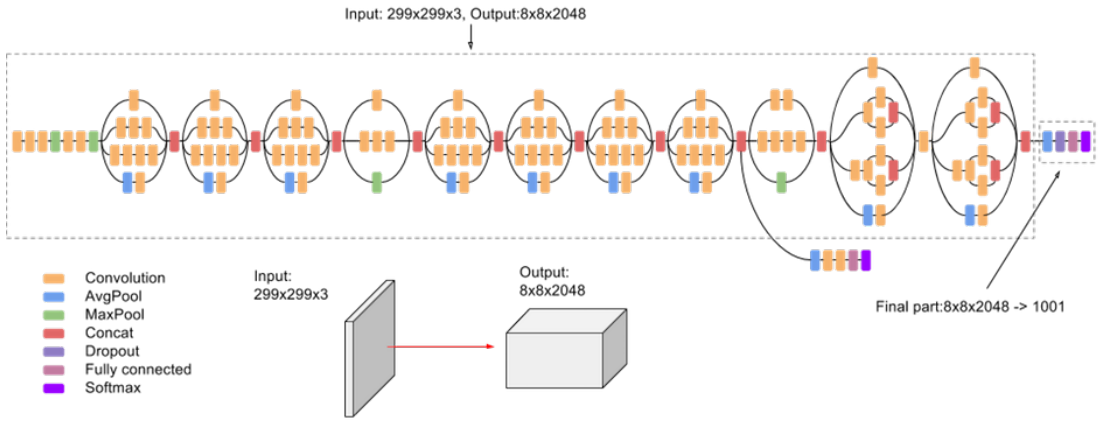
3.3.1 Tensorflow

Now we need some tools to help us manipulating the CNN training. TensorFlow is an open source framework to help with deep learning tasks, with easy used CNN models. Moreover, TensorFlow has many built-in libraries and a lot of user-friendly API to help people to implement for virtually any deep learning topic, such as image classification or generating artificial images. For instance, $TensorFlow.js$ is a JavaScript library for training and deploying models in the browser and on $Node.js$, and $TensorFlowLite$ is a lightweight library for deploying models on mobile and embedded devices. Furthermore, those models can convert into each other smoothly, which gives flexibility of the implementation. By using the same

algorithm, both an App or a website could be integrated with CNN algorithm easily. We can build the website or an App to recognize food from images based on the training.

Inception Model

Another thing, the pre-trained module, Inception v3 architecture model is widely used for CNN training. It can be used to train new classifier on top for the food recognition task. Inception v3 is a transfer learning image recognition model[22]. It works impressively on the ImageNet dataset[7], with more than 78% accuracy. It can be used to train a new model with own label by reusing the feature extraction, because itself contains convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers already. A diagram of this is shown below in Figure 3.13.



Source: image from Google Cloud
Figure 3.13: Inception v3 architecture model

None of our food classes were in the original ImageNet classes, so lower layers in the structure that have been trained to distinguish between objects can be reused for our food recognition tasks without any alteration.

Visualization - TensorBoard

Moreover, the TensorFlow library comes with a suite of visualization tools - *TensorBoard*. It helps people manipulate neural networks easier. With *TensorBoard* you can visualise neural network, and figure out what happens during the training with training progress and any additional information shown. For instance, visualising the accuracy of the training helps us to know when it has better performance. According to Figure 3.14, when the training steps is 400 or 500 shown in the x axis, the accuracy shown in y axis, is around 80%. This time reached a stable accuracy, so there is no need to increase more training steps.

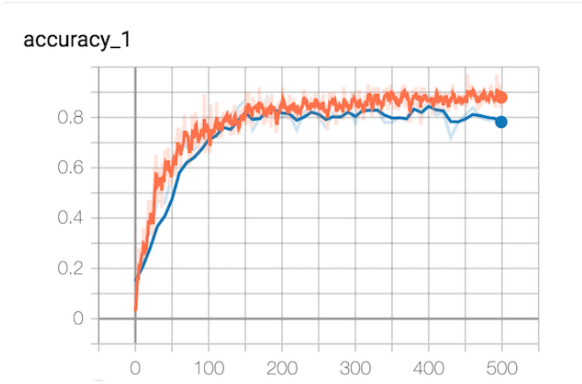
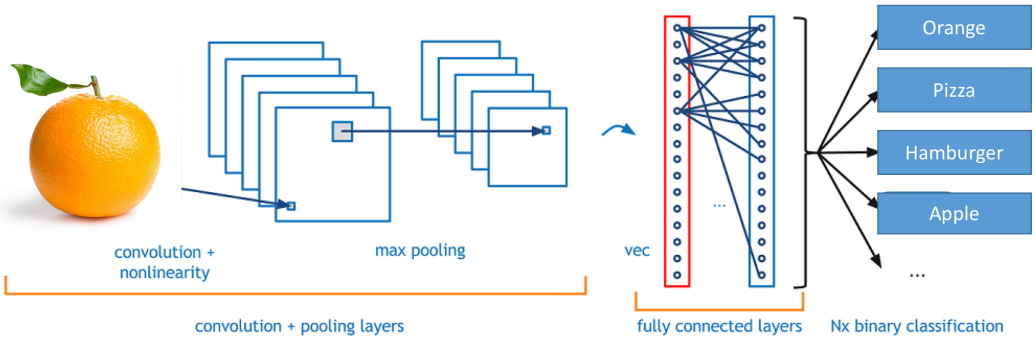


Figure 3.14: Accuracy from TensorBoard

As we all know, using CNN to build an image classifier could be complicated, but with Tensorflow, less daunting to implement, and easy to improve the accuracy of the model even further. With TensorFlow allows you to share a trained model easily. It is not so common across different frameworks. Many frameworks require a full code of a model in order to load its weights into. However, TensorFlow only asks for a checkpoint file, which will be generated during the training, and the knowledge of layers names for inference in later steps.



Source: image from Adit Deshpande
Figure 3.15: Tensorflow image classification pipeline

3.3.2 Oracle cloud infrastructure

Since CNN requires huge data workload, we are going to use Oracle server for this training. Oracle Cloud Infrastructure offers public cloud with good control and security. It also offers high-performance and high availability infrastructure services, which is suitable for training huge datasets especially for machine learning or deep learning tasks. Oracle Could Infrastructure is a web service that provides a lot effective virtual machine instance with flexible control via standard secure shell (SSH). Many modern enterprises are using Oracle developing, and running their most demanding applications.

Further, it is easy to manipulate Oracle instance by using the industry standard secure shell (SSH) protocol with a public-private key pair for authentication for Linux instances. For Windows instances, the Remote Desktop Protocol (RDP) client with a username and password would also works. Technically, we need to put our training food image datasets on the Oracle server, and training the CNN by using SSH from Jupiter notebook though the website interface. This gives us an easy way to train a huge dataset, while we can keep an eye on the progress. After the training, the classifying model could be saved and used for testing.

Chapter 4

Project Execution and Methodology

This recognition system explored both classic computer vision methods and modern neural network deep learning approach. Firstly, we are trying to build a liner multi-class SVM with LBP, Gabor responses and HSV values from the regions of interest(ROI). Then, a more modern method is using a CNN training for the same classification purpose. Both methods are implemented into classify ROI's depicts different foods in food images, in order to find the better classifier for specific food content. Those two approaches are discussed in following sections separately, using the same training and testing detasets discussed above.

4.1 Classic Computer Vision

In order to classify each food type, a lot of features are extracted, and stored in a map connected by the food type. However, the food image may contain a lot of other information that we do not want, such as background, so a pre-segmentation is required, which means only regions of interest(ROI) are used for feature extraction. In this case, ROI is food region in the bounding box, which is cropped by ground truth. The pipeline is to extract features from ROI of one food class. And those features should be able to distinguish different food classes, which means if a certain amount of features from one class are similar to another class in our train dataset, that image would be predicted as containing that specific food. To sum up, the pipeline is extracting features from ROI, then training classifiers by using those features. After the training, some random food images can be used to test this performance. The pipeline of classical approach is shown in Figure 4.1.

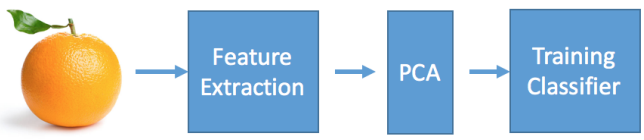


Figure 4.1: Classical Computer Vision Pipeline

4.1.1 Feature Extraction

The idea is to extract LBP, Gabor responses, and HSV values as features.

LBP

Firstly, Local binary patterns were extracted for each training food class based on the gray scale ROI, and this gives us 59 LBP features of one region for every class. The reason for using region based input is that when extracting LBP features, the neighbours need to be taken considerations. For each pixel, compare it to its 8 neighbours to encode the LBP. Then LBP features are stored according to the food

class, using an anontation in Matlab, which is like a dictionary for mapping food classes and features. The format is shown as follows.

foodClasses	LBP
pizza	[1X59] Single
roast	[1X59] Single
carrot	[1X59] Single
orange	[1X59] Single
……	……

Figure 4.2: Store region-based LBP features in anotation

HSV

Later on, other features could be appended on the right on the table. When we have enough features, we can train the classifiers. Next step is to extract Gabor features and HSV features. However, instead of region based, HSV is applied per pixel. To be more exact, we extracted HSV value from each pixel in the ROI, but instead of using all of three, H, S and V values, we discarded V values. As we talked in Technical background section, V is very sensitive to lighting and cameras. This can not only gives us better performance but also speed up the program a bit.

Gabor Responses

As for Gabor responses, we made a Gabor bank with the 39x39 pixels window to slide along the ROI for texture segmentation to identify regions based on food texture. So the number of rows and columns in our 2-D Gabor filter is both 39. The orientation does not matter too much in this scenario, so we used only four angles, 0°, 45°, 90° and 135°. Because in our scenario, we wish our system could recognize the food no matter what the orientation of the plate is, either how people put the food in different orientations. And the wavelength is from 2 to the square root of sum of image length and image height, after normalizing the source image into 500x500 size in the first step. Then we extracted Gabor magnitude features from that image. Our Gabor bank contains all the unique combinations of wavelength and orientation as above. And sliding window is more explicit to image contents and semantic meanings, however, fixed window schemes may produce errors when image contents are changed[27].

foodClasses	Gabor	HSV
pizza	100x1x24 double	100x1x2 double
roast	100x1x24 double	100x1x2 double
carrot	100x1x24 double	100x1x2 double
orange	100x1x24 double	100x1x2 double
……	……	……

Figure 4.3: Store pixel-based Gabor and HSV features in anotation

As can be seen from Figure 4.3, we got a lot mess features. HSV gives us 2 features for each pixel, H and S, while Gabor filter bank gives us 24 responses for each region. In order to map the features both based on pixel, we go through the Gabor response for each region then find the values for each pixels. This means we have 26 features for each pixel. However, not all of them are useful, and too much data slows down the training program. In the training section below, we will handle this problem.

4.1.2 Classifier Training

Followed by the classic computer vision approach, often these feature descriptors are combined with traditional machine learning classification methodologies, such as Support Vector Machines(SVM) and K-Nearest Neighbors(KNN) to solve the classification problem, and our food classification is no exception. Firstly, by using region-based LBP, the linear multi-class SVM only with PCA is build. By reducing the number of the components to 25 (59 LBP components in total as first), we got the best performance of this linear SVM classifier with accuracy around 42%. This is build with 1-against-1 multi-class SVM, this means we have to train a separate classifier for each different pair of food classes. Moreover, we used 5-fold cross validation for this training to estimate the size of this effect. It usually used to estimate the accuracy of this model by performing in practice. The aim of cross-validation is to test the ability of current classifier to give a prediction of a new observation which is not used in estimating it, especially in terms of overfitting or selection bias. Similarly, this can also give a preview of how the model will generalise to an unknown dataset. Some results of the performance is shown as follows.

PCA	LBP	
	SVM	KNN
11	36.5%	26.7%
15	37.6%	26.9%
20	38.1%	26.5%
25	42.4%	27.7%
27	41.9%	27.3%
30	41.7%	27.2%
59(all components)	37.4%	22.1%

Figure 4.4: Performance of SVM and KNN for LBP

This performance is encouraging given 73 food classes from UNIMIB2016 food dataset, and classes the food with many classes being unusually similar. For instance, there are a lot of different type of pastas, such as pasta sugo pesce and pasta sugo vegetariano. They are even different to tell from human eyes. Another classifier we build is based on Gabor respond and HSV values. Similarly, a multi-class SVM is build for pixel based. The performance of this classifier is shown in figure 4.5.

PCA	Gabor + HSV	
	SVM	KNN
6	13%	5.8%
8	13%	5.7%
11	12.9%	5.5%
20	13%	5.6%
26(all components)	12.8%	5.8%

Figure 4.5: Performance of SVM and KNN for Gabor and HSV

Pixel based classifier(Gabor and HSV classifier) looks like it has worse performance, but if we take

the mode class label for all pixels it may still produce good results. To be more exact, if we label all pixels, the most likely class might have 13%, then the second likely class would have 8%. By taking the mode class label of all pixels, we could find the correct prediction class.

Now we have two classifiers both based in region and based in pixels. However, as can be seen from the result table, the accuracy of both of them are quite unsatisfying, although under some conditions ,we could find the correct food class. We could somehow combine them together to increase the accuracy, but it does not make any sense if we break down the LBP features into each pixel, because it requires neighbours when calculating it. If there is only one pixel, it is hard to encode LBP information. We still need a better way for the classification, and deep learning gives us a lot other choices to increase the performance of our classification.

4.2 Deep Learning Approach

Deep learning methods have been introduced to solve computer vision tasks, such as face recognition and pattern recognition, and it produces valuable results. One of the benefits is that there is no need to figure out which features should we extract, and how to combine them for the training classifier. The artificial neuron would do all of these for us. Due to these benefits, the process of CNN training pipeline looks simpler, as in the Figure 4.6.

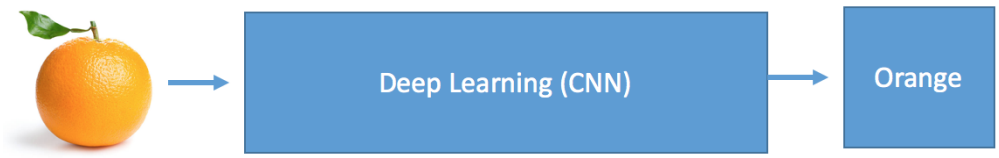


Figure 4.6: Deep Learning Pipeline

4.2.1 The Retrained Model

Training food images from scratch requires a lot of labeled training dataset and a lot of computing power, because the model could have millions of parameters to finally recognize food. Transfer learning is a technique that shortcuts much of this by taking a piece of a model that has already been trained on a related task and reusing it in a new model. We will reuse the feature extraction capabilities from powerful image classifiers trained on ImageNet from Tensorflow and use this to train our new food classification layer on top. In order to be able to get a better performance, we need enough dataset, much more than traditional computer vision methods. Only because of this amount of data generalization of the training set can be continually increased to some level then high accuracy can be achieved. Moreover, we need to teach the network about the new classes for recognition. So we put the name of the directories of our food images to be the classifications, so all our training dataset is clearly organized by their classes. Different from traditional computer vision method, each food class should be in the same folder named as the food class name.

Training

the training process handles calculating the bottleneck values. Final output will be given when CNN successfully built the classification. During the training, penultimate layer are keeping training until it gives outcome, which can distinguish between all potential classes. Meanwhile, we are getting a series of step outputs, each one showing training accuracy, validation accuracy, and the cross entropy. The training accuracy displays the current correct accuracy, while validation accuracy is the accuracy for randomly-selected group of images from a different set. The key difference is that the training accuracy is based on images that the network has been able to learn from so the network can overfit to the noise in the training data. If the train accuracy is high but the validation accuracy remains low, that means the network is overfitting and memorizing particular features in the training images that aren't helpful more generally. Cross entropy is a loss function which gives a glimpse into how well the learning process is

progressing. The training's objective is to make the loss as small as possible, so we can tell if the learning is working by keeping an eye on whether the loss keeps trending downwards, ignoring the short-term noise. This time we need to use Tensorboard to help us visualise the process by plotting figures.

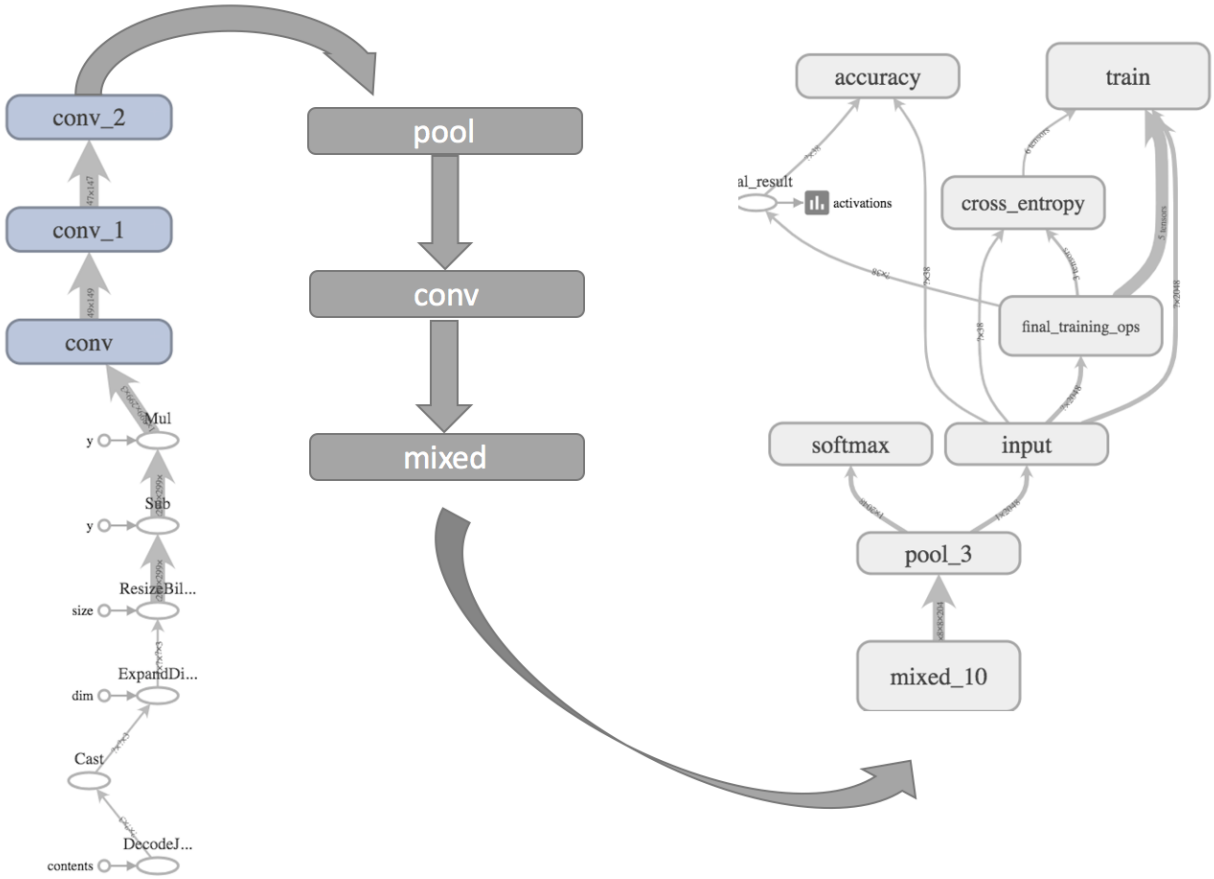


Figure 4.7: Tensorflow Pipeline

By using this CNN training, we got a much better performance on classification food classes, with an accuracy about 86.2% of 43 classes of food.

Chapter 5

Results and Critical Evaluation

In this chapter, we are going to evaluate feature extractors with accompanying classifiers on different test data images, then compare and contrast that results. The very first tests are using the testing images from the original dataset. This means our classifier was testing with the testing datasets split from the same datasets(UNIMIB2016 and UEC100), and those images are not used for training in previous steps in order to avoid overfitting. Another testing is under some new datasets, which are not in UNIMIB2016 either UEC100. Those images are grouped by their source, for instance, Instagram food images, commercial advertisement images.

Moreover, another human involved recognition test would also be involved in order to compare the performance of human eyes and computer vision classifiers. Some evaluation would be discussed based on those test outcomes, which also gives us some ideas for future plans.

5.1 Original Dataset Testing

5.1.1 Classic method results

In the first test using the same source food images, we are doing is to test our traditional computer vision SVM classifier with LBP features on the images from the same dataset(UNIMIB2016). There are 350 test images in total, containing more than 70 types of food. Those images are followed by the same pre-processing steps as training dataset, which means these 350 images give us more than 1000 food ROI. Then the next step is to extract the LBP features from test ROI. After that the multi-class SVM will classify the input by its' LBP values. This pipeline of this testing is shown as follows.

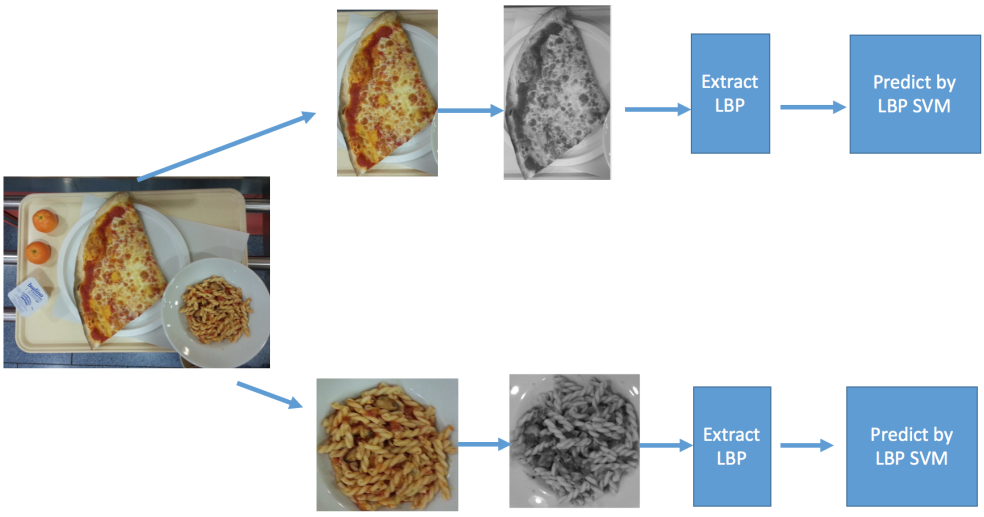


Figure 5.1: LBP SVM Testing Pipeline

Following the pipeline, we tested all of the 350 images containing 1254 food ROI, which contains more than 70 food classes. And the result is that about 551 of them are classified to the correct class. so the accuracy is 43.94%. Considering there are a large number of classes usually similar to the human eyes. So this result is relatively impressive.

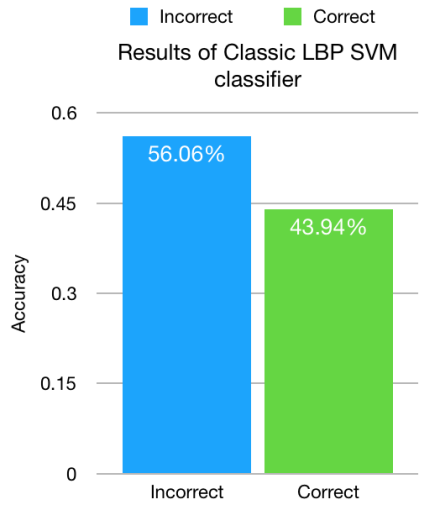


Figure 5.2: Results of Classic LBP SVM

As we talked before, this result is alright for region based multi-class LBP SVM with the same dataset. There are more than 70 classes in total. And these images are taken by the same camera with similar background(tray in the same canteen). This means those images have the same lighting, which benefits the testing. There could be more features to build the SVM classifier, however, our main focus is CNN classifier.

5.1.2 Deep learning approach results

As results for deep learning CNN classifier, test images from dataset UEC100 are used for testing the CNN classifier. There are about 600 testing images in total containing basically every food classes we have trained before. The process is that we run the classifier for each test image, then record the predication from the results then compare with ground truth. It it worth to notice that each prediction by CNN will be given with the score, which is the probability to be that specified food type. And we will take the class with the biggest score to be the final prediction.

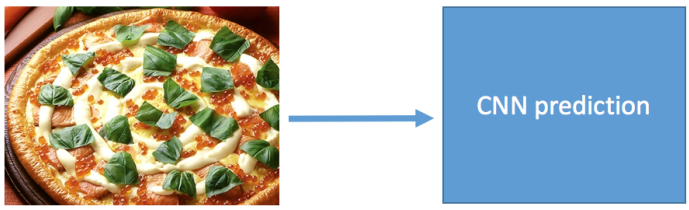


Figure 5.3: CNN Testing Pipeline

There are more than 600 images to test the performance of the CNN classifier, and the results is that 82.55% are classified into the correct food class. Although there are 43 classes of food, the results is rather good with more than 80% accuracy. Further, the results of both classic SVM and CNN is shown

as below in figure 5.4. Clearly, the algorithm pipeline employing a CNN has better performance than classic SVM one.

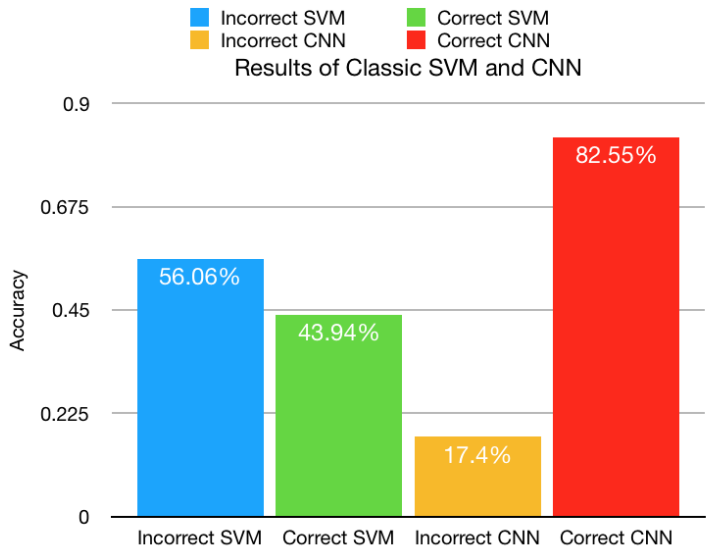
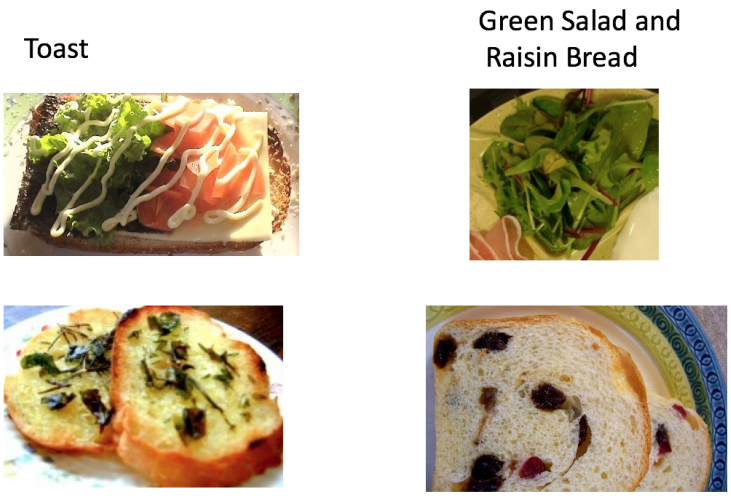


Figure 5.4: Results of LBP SVM and CNN

It looks like our CNN performed rather good. For food having a quite different textures from others, like popcorn, the accuracy quite high, which means the CNN extracted these unique texture very well. However, there are still some incorrect prediction of CNN. And we would like to reason a bit about these wrong classification cases. Some predication are not satisfying. In our case, we have toast and raisin bread from testing dataset and training dataset shown in figure 5.5. The first col in figure 5.5 are toast covered with salad and toast covered with seaweeds, while the second col shows green salad and raisin bread. It is can not deny that those toasts look a bit like salad and raisin bread even for human eyes.



Source: images from UEC100 dataset.
Figure 5.5: Toast and other two type of food examples

Ideally, testing covers almost every environment that this food recognition and nutrition measurement system is expected to operate in so that the system could be used in all the suitable situations. The simplest example is that people are more likely to use images of food from their phone, or at least taken

by them. That food is what users are going to eat, so our proposed system should be able to recognise those images from real life. However, those tests above are only using images from existed dataset. There are a lot difference between real life food image and the dataset images. For instance, the dataset images are always taken by the same camera with the same lighting, while in daily life, people have different cameras and the lighting varies a lot. So in the following section, we are going to use some more accessible food images in daily life for testing.

5.2 New Dataset Testing

This testing section might considered ambitious involving some other testing cases beyond normal classification testing cases. There are a lot of food images that have close connection with people in daily life, and we care about more for theses daily food images. So we are going to use those images to further test the performance of our classifier to check if it has the ability to deal with real world problem.

Those testing datasets are grouped into 3 groups by where are they from. In the following section some brief introduction of these datasets are given with some intuition examples. Note that in the literature real world images are generally not used for testing, including the paper which provided our datasets. We make no claims of statistical significance for this section, the results of these tests are, subjectively, quite encouraging.

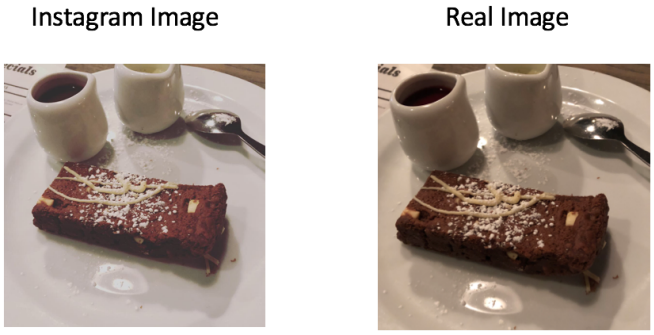
5.2.1 Commercial, Instagram and Edge-case Datasets

Firstly, we have food images from commercial advertisement. Advertising photography is everywhere around us in our daily life, specially food advertising, on the billboards along the roads, in shopping malls, in magazines, etc. They are familiar to us. These images would try the best to make food products look more delicious, even if it does not look the same as real situations. For instance, the lighting. Dodgy shadows, dingy lighting and night time are popular ways to make good photo. Another example is that when shooting burger images for advertising, photographer may use fake water droplets on the fresh red tomatoes, and photographer could use glue on the thick sesame bun. Hence these burgers are made to stand on their own perfectly, with plenty of light and a plain white backdrop. However, the burger you actually get might look quiet different from the advisement photography. The example is given as follows in Figure 5.6.

Source: image from businessinsider.com
Figure 5.6: Advertisements(Left) Vs. What You Actually Get(Right)

The second group of test images are from Instagram. Instagram is a social networking service for people to share photos and videos from their daily life. This means those food images people taken are actually what they are going to eat, instead of using fake material to fake the food in advisement photography. However, there are some secrets to capturing that perfect food photo making food look nicer, although the content is real food. For instance, there are many edit tools in Instagram. Instagram's built-in filters can reduce the quality of the photo rather than enhancing it, so users can edit with other image control app, which gives users great control. This means the images might look a bit different from the actual food, but within small changing such as filters. Moreover, the angle from camera is quiet different from how original datasets were taken. To be more precise, the original images are almost taken

from a vertical view. But images from Instagram are more flexible. There is an example of same brownie image from the original and Instagram in figure 5.7.



Source: image from Instagram
Figure 5.7: Instagram Vs. Reality

The next group of test images are made from an artistic perspective, which means those food in the image are trying to shape into something else, so it does not look like the actually food even to human eyes. Those food usually occur in some theme restaurants, and chefs from there always use food to build other stuffs to make fun. This could be the most confusing test even to human eyes, so in our experiment we call it edge-case testing. Furthermore, this is a kind of place people would like to take photos to record those food images. There is an example of the Nike shoes sushi image and the normal sushi in figure 5.8.

Source: image from google image
Figure 5.8: Nike Sushi Vs. Normal Sushi

5.2.2 Test Results and Evaluation

There will be 11 images from each of commercial, Instagram and edge-case group of some food classes to test the performance of our CNN classifier under these daily conditions. Firstly we chose sushi to test with, since sushi is a popular food worldwide and it is easy to find sushi images for commercial, Instagram and edge-case. Not all food is suitable for this kind of testing, for example burger is a kind of fast food, so it is relatively harder to find edge-case image of burger. Details of the results are described as following.

Advertising test

There are 11 advertising images from google of sushi, and we used those for CNN classifier to predict. The result is that 8 of 11 are classified to the correct label(sushi). However, we are more interested into the failed cases. Those three cases are shown as follows, with the wrong class labels they predicted into

in the first row and the sushi images of that test then the wrong prediction class image examples are in the last row.

Source: image from google image
Figure 5.9: Failed Advertising Test of Sushi

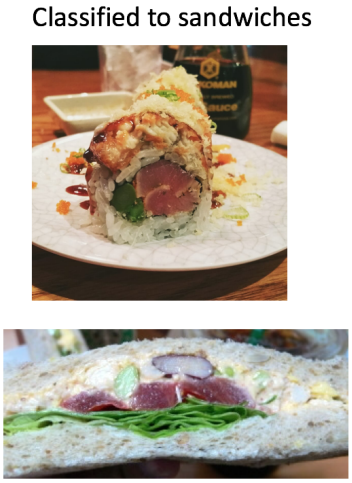
There are several things to reason about wrong predictions. For the first wrong tempura case, this advertising image shooting sushi in a row with curve, moreover the sources on sushi looks similar to tempura. As for paella de marisco case, both sushi and paella de marisco contain rice and fish. There are some common between them. For the testing sushi image, there are a lot of sushi in a big black plate. And for paella de marisco, it always be put into in a big black plate just like in the sushi image. We may think the background of the food would have some impact on the prediction. In order to get better performance of the classifier, user may need to crop the food image like in then training step for ROI. This could reduce the background influence on the prediction, but this is not a good way for users. For some time people take a photo of the food not only because the food itself looks wonderful, but also it goes well with the background decorations.

Finally, the last case might look surprising. Popcorn does not look like sushi at all. However, just like the second case, background sometimes confuses our classifier. If you pay attention to the background of the sushi image for the last case, you will notice that there are a lot green leaves. Furthermore, those background green leaves are blur for the advertising. It is hard to deny that the frequency of those green leaves are quite similar to the frequency of popcorn.

However, if people are using the classifier to recognize food for nutrition information, people would not using images from adverting. They would more likely to use photos of real food they are going to eat. So in the next test, we are going to forces on a more realistic food datasets, Instagram food images.

Instagram test

There is only one incorrect prediction of sushi image from 11 Instagram sushi images. And the testing image is shown as following, with the sandwiches example. The testing image focus on the cross section of the sushi making the sushi looks like a sandwiches with red meat in the middle. The testing image is shown is figure 5.10 with the contrast image of red meat sandwiches.

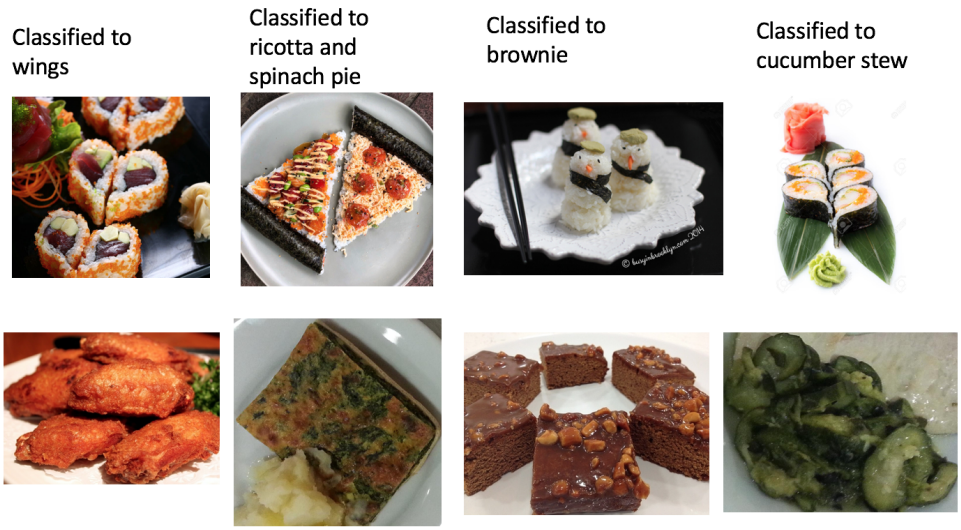


Source: image from google image
Figure 5.10: Failed Instagram Test of Sushi

This informed us that the sometimes the angle of a photo might make a food look like another one. It is easier for human eyes to tell the type of food from different angles with all the life experience. However, our classifier is rather naive. This could be improved by using more training images from a variety angles, however, in our training process, we used images from datasets and those images are almost taken from the same angle with the same camera setting. In general, we are happy with the performance of the CNN classifier for Instagram images.

Edge-case test

Next, we tested our CNN on those creative sushi images. There are 4 incorrect predictions for this test displaying in figure 5.11. The first row are testing sushi images, and the images below them are wrong prediction class images.



Source: image from google image
Figure 5.11: Failed Edge-case Test of Sushi

As can be seen from figure 5.11, those images are less sushi like. Some of the sushi are shaped into heart shape, and some are shaped into snowman. What is more, there are some pizza look sushi. These

examples are even hard to recognise for human eyes at the first time. Although sushi are made into other shapes rather than traditional sushi roll, the integrate material does not changed(still made with rice, seaweed and some fish). In spite that the colour and maybe the shape changed a lot, the texture information should not change so much. However, our classifier could not predict correctly, so maybe more texture information on the food material itself should be extracted for training.

5.2.3 Compare and Contrast

Here are some results of testing from our CNN classifier, and some human involved testing are also shown below in figure 5.12. Some participants are asked to recognize food type from the same image datasets(sushi advertising, sushi Instagram and sushi edge-cases) by using eyes. The y-axis is the accuracy of the prediction and the blue bar is for our CNN classifier while the green bar shows the human eye.

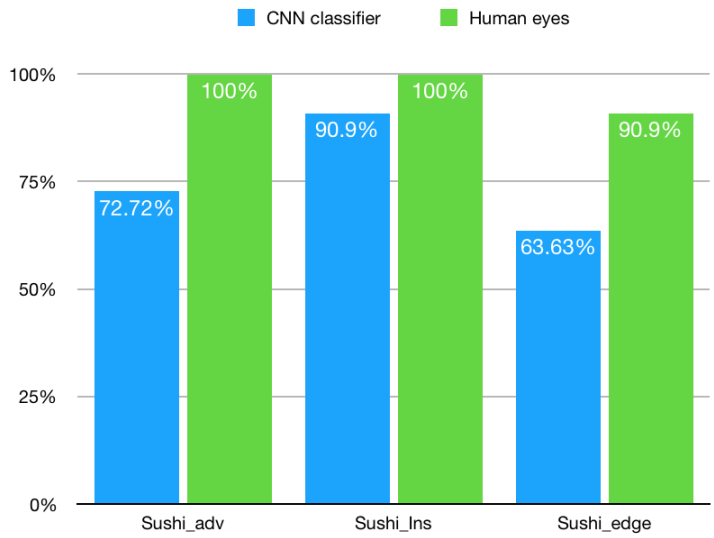


Figure 5.12: Accuracy of CNN classifier and human eyes

It is pretty clearly from the bar chat that we have the worst performance for edge-case images for both CNN classifier and human eyes. However, for human eyes the prediction is always correct for advertising and Instagram images. Moreover, people seem could spot the fake food from advertising, although they known which food they were faking. Participants claimed they can tell if the food is real by the background of the image. For example, if there are some worlds in the image for description, they would know it is more like a poster for food. Moreover, human could easily recognise the number of the food pieces from images. Apparently our CNN classifier could not do this compared with humans. In spite that there are some fake sushi images in group advertising, our CNN classifier still gives the sushi prediction with a rather lower score probability. There are three fake sushi images from advertising group, and the score is highest the probability among all the food classes we trained. For these three fake food cases, although our classifier gave the sushi prediction, the score is quite low (below 0.2). In other situations, the score is always above 0.2, which shows our classifier is slightly confident about the prediction.

Summary

Both classic SVM and deep learning CNN perform impressively for the original datasets, even the results for the traditional method are acceptable considering there are a lot similar food classes. Moreover, CNN performs even better. To be more precise, results on default datasets provided by previous authors are impressive, but does not guarantee it would generalise well to different (real-world) environments. So another three datasets were used for testing the performance of CNN. Although there are some

misclassifications, the general performance is quite satisfying. To sum up, results of our real life image testing also gives encouragement, and it gives us some ideas to improve the system.

Chapter 6

Conclusion

6.1 Summary of Achievements

The big dream is that people can take a picture of the food they are going to eat, and our system would give the prediction of that food according to the image with the nutrition information. Since we only focus on the foundation of the system, segmentation and classification, the software part is not our focus. But we implemented a rudimentary website of the proposed system. Some current work of the website would be demonstrated in future focus section.

In terms of our main focus, we have built two different classifiers followed by different pipelines. Both multi-class SVM and CNN could classify food with good accuracy. CNN works better than SVM in our case. Hence we explored CNN with further method with other new image datasets in order to simulate real world environments. Furthermore, results on original datasets provided by previous author are impressive for both classifiers. And results of CNN on simulation environment also gives encouragement.

Extensive software development was beyond the experiment of this project, however, we managed to implement a rudimentary website connecting to MongoDB database, where we stored our nutrition information. Ideally, when our system recognise the food from the image users uploaded, the corresponding nutrition information would be shown in front of users.

Again, the achievements of each one are listing in bullet points with rather brief descriptions.

6.1.1 Multi-class SVM

- Extracted region based feature like LBP(Local Binary Patterns) and Gabor response.
- Extracted pixel based texture features like HSV colour space.
- Applied PCA(Principal Component Analysis) to feature space for dimensionality reduction.
- Implemented K-Nearest Neighbors to test dataset.
- Trained linear multi-class SVM with 5-fold cross validation for classification, which works with 73 food classes with accuracy about 42.4%.
- Tested linear SVM classifier with original dataset having 43.94% accurate.
- Built rudimentary user interface website connecting to MongoDB database.
- Nutrition information stored in MongoDB.

6.1.2 Deep Learning CNN

- Trained deep learning CNN(Convolutional Neural Networks) using Tensorflow Inception v3 architecture model.
- Manipulated Oracle cloud server via SSH for more effective deep learning training.

- CNN works with 43 food classes with accuracy about 86.2%.
- Testing with original dataset with 82.55% accurate.
- With an 72.72% accuracy for advertising image test images.
- With an 90.90% accuracy for Instagram image test images.
- With an 63.63% accuracy for edge-case image test images.
- Built rudimentary user interface website connecting to MongoDB database.
- Nutrition information stored in MongoDB.

6.2 Future Focus

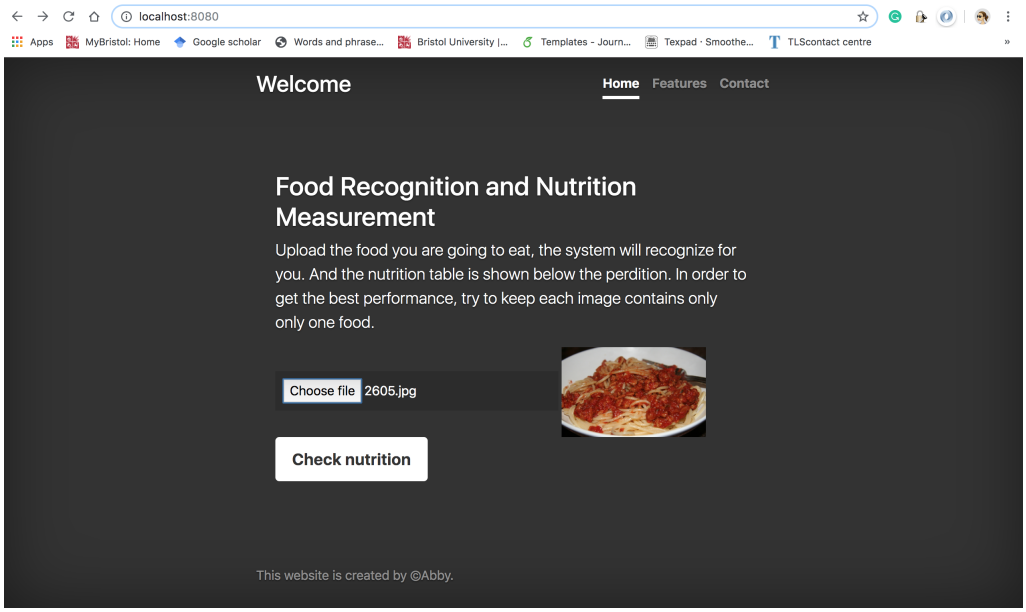
6.2.1 Improving classifier

In terms of classifier performance, there are two main aspects which could be improved. Firstly, the number of food classes could be increased. Currently we have 43 classes, however in the real world there are more food types, so more classes need to be trained with our classifier, even though some of them may look similar, for instance, sushi and sashimi, or fettuccine and spaghetti. Another thing we could improve is the accuracy of the classifier. Although 86% is not low, it could be more accurate if we could use the same food image from different dataset during the training. This could reduce the bias of the same dataset image would have the same angle of the image with the same camera settings.

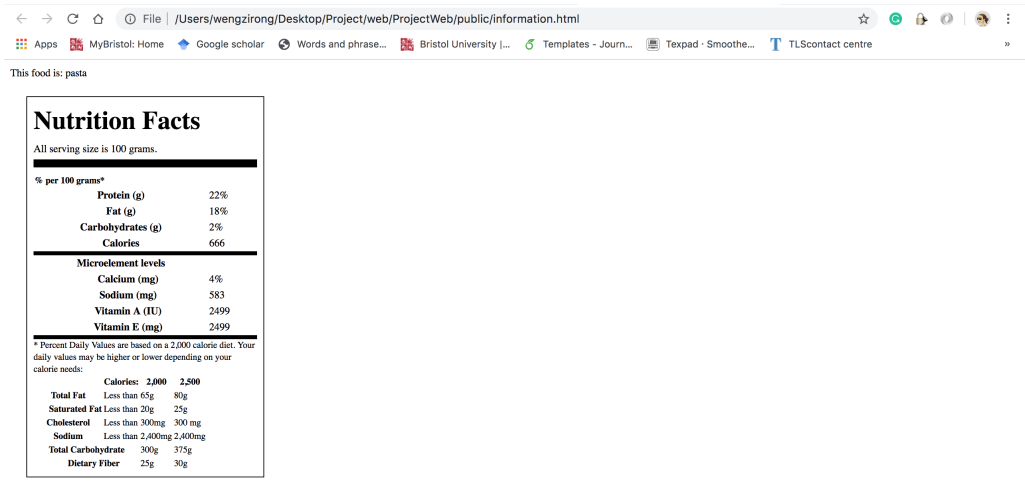
Moreover, inspired by human involved testing, it would be interesting if we train our classifier to recognize how many pieces of food are in the image, or what is the portion of that particular food. Think about the whole system again, the final goal is to give nutrition information. The more precise food information classifier gives, the more accurate nutrition information would be. Alternatively, we can give reference nutrition values, for example, we can give all the information based on serving size is 100 grams, then users could evaluate the whole nutrition values.

6.2.2 Rudimentary Website

More importantly, we need to have a user interface so that people without computer science background could use this system easily. Currently this classifier can be only used through terminal, and there is no nutrition information of the predicted food. Ideally, we need either an APP or a website for user to upload the image for the food. And users could use this system by simply upload images and click some buttons for prediction and nutrition table of that food. The potential website would look like in the figure 5.13.



(a) Main page



(b) Nutrition page

Figure 6.1: Potential Website example

User could upload the image by choose file, then they can visualise this image from the right hand side preview. Finally, clicking the check nutrition button leads user to the prediction and nutrition table in (b). This is a intuitive example, and this could be an App as well, not necessary to be a website.

Moreover, there are other potential functions our system could have, for instance, the system might record user's food images into another user dataset, then produce the eating habits of that user then give some feedback and advice from every week or mouth if user would continue uploading images.

Bibliography

[1] Tadhg Brosnan and Da-Wen Sun. "Improving quality inspection of food products by computer vision—-a review". In: Journal of food engineering 61.1 (2004), pp. 3–16.

[2] Gianluigi Ciocca et al. "Food recognition: a new dataset, experiments and results". In: IEEE Journal of Biomedical and Health Informatics 21.3 (2017), pp. 588–598. DOI: 10.1109/JBHI.2016.2636441.

[3] Fadi Dornaika et al. "A Comparative Study of Image Segmentation Algorithms and Descriptors for Building Detection". In: Handbook of Neural Computation. Elsevier, 2017, pp. 591–606.

[4] Laurel D Edmunds. "Social implications of overweight and obesity in children". In: Journal for Specialists in Pediatric Nursing 13.3 (2008), pp. 191–200.

[5] Giuliano Enzi. "Socioeconomic consequences of obesity". In: Pharmacoeconomics 5.1 (1994), pp. 54–57.

[6] Giovanni Maria Farinella, Marco Moltisanti, and Sebastiano Battiato. "Classifying food images represented as Bag of Textons." In: ICIP. 2014, pp. 5212–5216.

[7] Li Fei-Fei. "ImageNet: crowdsourcing, benchmarking & other cool things". In: CMU VASC Seminar. Vol. 16. 2010, pp. 18–25.

[8] K Frederick and P Ciborowski. "SWATH-MS: Data Acquisition and Analysis". In: Proteomic Profiling and Analytical Chemistry. Elsevier, 2016, pp. 161–173.

[9] W. Geisler, M. Clark, and A. Bovik. "Multichannel Texture Analysis Using Localized Spatial Filters". In: IEEE Transactions on Pattern Analysis Machine Intelligence 12 (Jan. 1990), pp. 55–73. ISSN: 0162-8828. DOI: 10.1109/34.41384. URL: doi.ieeecomputersociety.org/10.1109/34.41384.

[10] David Haslam, Naveed Sattar, and Mike Lean. "ABC of obesity: obesity—time to wake up". In: BMJ: British Medical Journal 333.7569 (2006), p. 640.

[11] Ye He et al. "Analysis of food images: Features and classification". In: Image Processing (ICIP), 2014 IEEE International Conference on. IEEE. 2014, pp. 2744–2748.

[12] Mahamadou Idrissa and Marc Acheroy. "Texture classification using Gabor filters". In: Pattern Recognition Letters 23.9 (2002), pp. 1095–1102.

[13] Corinna Jacobi et al. "Coming to terms with risk factors for eating disorders: application of risk terminology and suggestions for a general taxonomy." In: Psychological bulletin 130.1 (2004), p. 19.

[14] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. "Introduction to information retrieval". In: Natural Language Engineering 16.1 (2010), pp. 100–103.

[15] Jennifer Mindell et al. "Cohort profile: the health survey for England". In: International journal of epidemiology 41.6 (2012), pp. 1585–1593.

[16] CJ Murray, AD Lopez, et al. "The global burden of disease study". In: Lancet 349 (1997), pp. 1269–1276.

[17] Poul Erik Petersen. "The World Oral Health Report 2003: continuous improvement of oral health in the 21st century–the approach of the WHO Global Oral Health Programme". In: Community Dentistry and oral epidemiology 31 (2003), pp. 3–24.

[18] D Reilly, CA Boyle, and DC Craig. "Obesity and dentistry: a growing problem". In: British dental journal 207.4 (2009), p. 171.

[19] Geeta Shroff, Asim Smailagic, and Daniel P Siewiorek. "Wearable context-aware food recognition for calorie monitoring". In: Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on. IEEE. 2008, pp. 119–120.

[20] S Sinharay. "An Overview of Statistics in Education". In: (2010).

[21] Carlos Oscar Sánchez Sorzano, Javier Vargas, and A Pascual Montano. "A survey of dimensionality reduction techniques". In: arXiv preprint arXiv:1403.2877 (2014).

[22] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 2818–2826.

[23] George W Taylor, Michael C Manz, and Wenche S Borgnakke. "Diabetes, periodontal diseases, dental caries, and tooth loss: a review of the literature." In: Compendium of continuing education in dentistry (Jamesburg, NJ: 1995) 25.3 (2004), pp. 179–84.

[24] Burghardt Tilo, Sion Hannuna, and Andrew Calway. Lecture 05-Segmentation Basics. 2018. URL: `https://www.ole.bris.ac.uk/bbcswebdav/courses/COMS30121_2018/content/coms30121tb05.pdf` (visited on 10/30/2018).

[25] Adam Gilden Tsai and Thomas A Wadden. "Systematic review: an evaluation of major commercial weight loss programs in the United States". In: Annals of internal medicine 142.1 (2005), pp. 56–66.

[26] Lipo Wang. Support vector machines: theory and applications. Vol. 177. Springer Science & Business Media, 2005.

[27] Yalin Xiong and Steven A. Shafer. "Variable window Gabor filters and their use in focus and correspondence". In: CVPR. 1994.

[28] Chang Xu et al. "Image-based food volume estimation". In: Proceedings of the 5th international workshop on Multimedia for cooking & eating activities. ACM. 2013, pp. 75–80.