FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Exploring computer vision strategies of food recognition for dietary assessment

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Media Informatics and Visual Computing

by

Andreas Fermitsch
Registration Number 0406978

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Martin Kampel
Assistance: Dr. Rainer Planinc

Vienna, 7th December, 2017

Andreas Fermitsch	Martin Kampel

Erklärung zur Verfassung der Arbeit

Andreas Fermitsch
Neubaugürtel 23 / 9 1150 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 7. Dezember 2017

Andreas Fermitsch

Kurzfassung

Zunehmende Gesundheitsprobleme wie Diabetes oder Herz-Kreislauferkrankungen können unter anderem auf unsere Ernährung zurückgeführt werden. Diverse Applikationen benötigen eine Monitierung ernährungsbezogener Metadaten, wie etwa die Nahrungszusammensetzung der täglichen Kalorienaufnahme. Existierende Methoden für Ernährungsauswertungen, wie *24-Stunden Erinnerungsprotokolle* oder *Verzehrsprotokolle*, sind oft ungenau, zeitaufwändig, teuer und schwerfällig. Auswertungen die auf Photographien von Malzeiten basieren, bieten eine Alternative zu den traditionellen Methoden. Sie eröffnen die Möglichkeit, diese zu unterstützen und zu verbessern.
Um State-of-the-art Methoden zur Erkennung von Essen zu identifizieren, wird eine Recherche gängiger *Computer Vision*-Methoden durchgeführt. Die *Bag-of-Features*-Codierung wird als Bemessungsgrundlage, und die *Fisher Vector*-Codierung als Erweiterung implementiert. Diverse Farb- und Textur-Bildbeschreibungen werden verglichen und analysiert. Um die Codierung weiter zu verbessern werden zwei räumliche Sampling-Methoden für die Bildbeschreibungen verwendet. Um mehrere Bildbeschreibungen zu kombinieren, werden zwei unterschiedliche Strategien miteinander verglichen. Eine dritte Implementierung verwendet *Convolutional Neural Networks* für die Extraktion von Bild-Merkmalen und die Kategorien-Klassifizierung. Zwei Netzwerk-Architekturen, *AlexNet* und *GoogLeNet* werden verwendet. Alle drei Techniken werden auf drei unterschiedlich großen Bild-Datensätzen von Malzeiten angewandt. Die *Fisher Vector*-Codierung übertrifft die *Bag-of-Features*-Codierung, und die *Convolutional Neural Networks* die *Fisher Vector*-Codierung. Die besten Klassifizierungsresultate liegen bei etwa 80% bei 100, und bei etwa 71% bei 256 Essens-Kategorien. Bei allen drei Datensätzen, liegt die Top-5 Erkennungsrate in einem Bereich von 90-96%.

Abstract

Diet is a contributing factor for growing health concerns, such as diabetes. For various applications the need arises to monitor meta data, such as the caloric dietary intake composition of daily life. Existing methods of dietary assessment, such as *24-hour Dietary Recalls* or *Dietary Records*, are often inaccurate, time-consuming, costly and cumbersome. Assessment on basis of photographs of food, promises to be an alternative to, or a support of traditional methods.
Research of *computer-vision* techniques for food recognition is conducted and the state-of-the-art methods identified. The *Bag-of-Features* technique is implemented as a baseline method, the *Fisher Vector*-encoding as an improvement of the technique. Several colour and texture descriptors are compared and analysed. To further improve the method, two spatial sampling techniques are used for each descriptor. For the combination of various single descriptors, two fusion strategies are compared. The third implementation uses *Convolutional Neural Networks* as feature extractor and classifier. Two network architectures, *AlexNet* and *GoogLeNet* are used. The three techniques are applied on three food image-datasets of different sizes. *Fisher Vector*-encoding outperforms *Bag-of-Features*-encoding, and *Convolutional Neural Networks* outperform the *Fisher Vector*-encoding. The top results achieved in the image classification task are around 80% recognition rate in a 100 food-category problem, and around 71% for 256 food categories. Top-5 recognition rates are in a range of 90-96% for all three datasets.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Diseases like diabetes, obesity, cancer and heart-related health issues are on the rise globally and a growing concern in our society [Shim et al., 2014, McAllister et al., 2009]. One besides other lifestyle factors of the increasing incidences of these diseases, like lack of physical activity, is an unhealthy diet [Rhyner et al., 2016]. Studies show that diet changes can reduce cancer incidences by one-third and are associated with a low risk of all-cause mortality [Shim et al., 2014]. Dietary data from long-term observation can help to predict risks of cardiovascular diseases. Epidemiological studies are necessary to identify risk factors for mentioned diseases to then be able to detect patterns for prevention of those diseases. To conduct such studies a great collection of data is needed.

The number of adult people currently with diabetes are estimated by the [International Diabetes Federation, 2016] to be 415 million, with expectations of 642 million by 2040. In the United States health care in 2014, US $1 out of US $9 was spent on diabetes [Rhyner et al., 2016], illustrating the financial implication of the disease.

There are two main types of diabetes mellitus: type 1 (T1D) and type 2 (T2D). 85% of diabetes patients suffer from T2D [Forouhi and Wareham, 2014]. T1D is an autoimmune process in which the pancreatic beta cells are destroyed, cells that produce insulin, making it necessary for the patient to supply exogenous insulin [Rhyner et al., 2016]. The key factor to determine the postprandial insulin dose is to estimate the content of carbohydrates of the meal [Anthimopoulos et al., 2015]. For an affected person it is therefore a necessity to have a close observation of all dietary intake, especially the carbohydrate intake throughout each meal.

Also obesity is one of the major health issues in developed countries. In the US the majority of the population is overweight or obese [Karl and Roberts, 2014]. While

in 2008 one in ten of the worlds adult population was obese, in 2012 it were already one in six. Recent studies show that obese people have higher risk of hypertension, heart attack, type 2 diabetes, high cholesterol, breast and colon cancer, and breathing disorders [Pouladzadeh et al., 2013].

Although the reasons for an obesity epidemic since the mid-20^{th} century are not completely understood, two strong contributors are modern food marketing practices and physical inactivity. "Increased portion sizes in commercially marketed food items, inexpensive food sources such as fast food, increased availability of vending machines with energy-dense items, increased use of high fructose corn syrup", [McAllister et al., 2009, p. 2], are some examples of the aftermath of methods of our food industry, affecting our dietary intake and overall nutritional behaviour.

A general consensus of a reason for weight gain lies in consumption of unnecessary calories. Key to prevent weight gain is to monitor overall energy intake in relation to energy expenditure of physical activity, making it necessary to assess a long-term nutritional intake estimation of people suffering from obesity.

Existing traditional methods for short- and long-term dietary assessment used to study the aforementioned amongst other disease and health related issues, have many known shortcomings when it comes to accuracy of the dietary intake. Methods like the 24-Hour Recall (24HR) and the Food Frequency Questionnaires (FFQ), suffer from under-reporting. Food diary methods often have the effect on an observed person, to change their nutritional behaviour during the assessment [Shim et al., 2014].

Therefore the first motivational aspect is to improve the accuracy of existing assessment methods. In the past years there has been a growing interest of developing technologically assisted systems for more automation and the goal of a high accuracy of dietary assessment [Illner et al., 2012, Stumbo, 2013]. Computer-vision methods are advancing at a fast speed through growing research interest. Object recognition methods are getting closer and closer to compete with humans in more and more recognition tasks [Borji and Itti, 2014].

The second motivational aspect is improving usability and relieving the assessment process for patients and users. Existing methods are time-consuming, costly and cumbersome. Through new technological advances and mainly automation, dietary assessment becomes easier for users and patients to conduct in daily life and it is cost-effective because no medical personnel is needed for qualitative methods of data enquiry.

The following list identifies target groups of users for such a system, to illustrate the applicational range of dietary assessment systems:

- People depending on assessing their macro-nutrient(carbohydrates, fats, protein and alcohol) intake composition like diabetes patients.
- Obese people that try to estimate their calorie intake to loose weight

- People with fitness goals, e.g. gaining muscle mass or loosing fat benefit from tools to keep track of their macro-nutrient intake
- For individual medical investigation on specific micro- and/or marco-nutrient composition or deficiencies.
- For epidemiological studies, because of the lack of accurate cost effective methods to assess data in a large scale.

1.2 Challenges

To support a person in counting their daily calorie intake or in documenting their dietary habits and trends, an easy and practical way of obtaining this data for the user, is through image data of photos of the foods taken before consumption, e.g. with a smartphone camera. This can be done from any setting like a restaurant, at school, at home or while travelling.

The main components of a dietary assessment system are the recognition of the food categories and the volume quantification. With the resulting meta-data of the assessment (such as an ingredient list and their corresponding volume), the structural meta-data can be looked up in food databases. E.g. macro-nutrient composition estimation of the food components, overall calorie content estimation or a list of the micro-nutrients (vitamin and mineral content), depending on the research goal or application.

The first step of a computer vision based system is to identify the content of the available image, i.d. recognising the kinds of foods and ingredients on the image. The difficulty of food object recognition lies in the ways food can be represented in images: the intra-class variation of a class of food is very high [Yang et al., 2010], due to the creative freedom of the chef and the many ways a certain dish may be prepared. Same instances of food items appear in different shapes and sizes, depending on how they are cut or prepared, same food ingredients often appear in different colours or shades of colours, depending on their freshness, also the texture of same instance foods vary etc. These obstacles complicate the problem, but there are approaches in object recognition to deal with these difficulties, such as invariances in rotation or colour intensity of descriptors or increasing the training data to statistically cover more variance. Occlusion however poses a problem that is different in nature. While recognition of single food items, like single fruits or a clear bowl of rice is an easy problem, recognising complex meals with lots of chopped and mixed ingredients covering other ingredients is challenging [Knez and Šajn, 2015]. If not all information is included on the photo (like a piece of meat deep inside a soup, not visible on the photo), then this information can therefore not be recognised, neither from a human nor from an algorithm, making accurate dietary intake assessment on basis of image data as a general solution not plausible in real life [Pouladzadeh et al., 2013]. To improve the estimation errors resulting from occlusions, additional sensors may bring an improvement [Knez and Šajn, 2015]. Computer vision does not provide a one-fits-all

solution, but has the potential to contribute to an improvement of the dietary assessment process, increase its usability and lowering its costs [Sharp and Allman-Farinelli, 2014]. Also the fine-grading of what items constitute as a single food category is debateable and dependent on the application, considering the very large number of different kinds of foods and combinations of ingredients [Puri et al., 2009].

The following list summarises identified challenges [Beijbom et al., 2015, Myers et al., 2015]:

- open-world recognition (unbounded number of categories)
- fine-graded recognition (differentiating between subcategories of food items)
- hierarchical label spaces (handling related labels)
- visual attribute recognition (e.g. distinguishing between fried vs. baked)
- occlusion (unlikely to convey complete compositional information from visual data)
- volume estimation is very challenging
- assumptions (such as background, calibration targets) reduce usability or lead to unrealistic arrangements
- multiple instances of items
- segmentation of items

1.3 Scope

The theoretical scope of this thesis is first a profound analysis of state-of-the-art approaches that apply computer vision techniques for assessing dietary information and/or performing visual food classification. The heart of such a system, is the method of recognition of food objects on digital photographs of the specific dishes. The second part of the theoretical scope is a detailed discussion of three of the recognition methods. The selected methods are

- Bag-of-Features (BoF)-encoding of texture and colour features
- Fisher-Vector (FV)-encoding of texture and colour features
- Deep Convolutional Neural Network (DCNN)

Due to the simplicity and the good performance, the Bag-of-Features (BoF) approach has become well established [O'Hara and Draper, 2011], and for that reason is selected as a baseline to show the advances of the other two methods. Encoding with the Fisher Vector (FV) has shown to outperform the BoF-encoding significantly in recognition tasks [Sanchez et al., 2013]. [Kawano and Yanai, 2015b] show an improvement of accuracy and processing time of FV over BoF, applied to food categorization. Deep Convolutional Neural Networks (DCNNs) recently got increasingly popular for large-scale image recognition tasks [Simonyan and Zisserman, 2014], after winning the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [Krizhevsky et al., 2012]. Food recognition experiments conducted by [Kawano and Yanai, 2014] showed improvement in accuracy using pre-trained DCNNs over using FV-encoding, and [Kagaya et al., 2014] show improvements of DCNNs over BoF-encoding.

The practical scope of this thesis is limited to the computer vision perspective of food object recognition. Specifically, to the implementation of the mentioned object recognition algorithms, to compare the different techniques with experiments on the same data. The performance of the methods will be tested in various experiments of several image descriptors and two DCNN architectures on three datasets of food images: the two UEC-FOOD datasets [Kawano and Yanai, 2015b, Kawano and Yanai, 2015a], and the FOOD101 [Bossard et al., 2014] (described in detail in Section 5.2.1). Those datasets consist of 100/256 and 101 classes of food dishes. The images are collected from the internet and represent images that are taken from real users (snapshots) under uncontrolled conditions. The total amount of images in the sets are 31.394 and 101.000.

1.4 Contribution

The following enumeration summarises the contribution of the thesis:

1. Research of the history and applications of dietary assessment, to understand the motivation and needs of possible applications, and establishing existing methods and tools used for food related assessments in fields such as medical applications and research.
2. Research of work related to food recognition that uses computer vision methods to identify food categories on images. The collected work is then examined to establish the recognition methods that are currently used for this purpose. The result of the examination is a selection of the most promising methods.
3. A presentation of the selected methods. This serves for a deeper theoretical understanding of the used recognition techniques.
4. A practical implementation of three selected object recognition methods.
5. An evaluation of the methods. For a direct comparison all methods are tested on three datasets of food images.

1.5 Outline

In Chapter 2 the historical background of dietary assessment, presenting common practical methods used in medical application. The chapter also presents computer aided tools such as web-platforms or smartphone apps. In Chapter 3 state-of-the-art projects that take a computer vision approach for identifying food categories are presented. The general approach and the used techniques are described for each project. From this overview of current research, the most promising recognition methods are identified, and three methods selected. Chapter 4 gives a detailed description of the selected object recognition methods. For each method, results from the previously described projects are compared. In Chapters 5 the methodology used in the practical part of this thesis is discussed, and the implementation details are described. Finally Chapter 6 presents the results from all experiments, including various combinations of techniques and an evaluation of the best results achieved for each of the three recognition methods on all three datasets. And Chapter 7 draws conclusions from the comparison of the recognition methods as well as from the achieved experimental results.

CHAPTER 2

Background of dietary assessment

2.1 Dietary assessment methods

Dietary assessment in medical research goes back to the 1930s [Stumbo, 2013]. Since then a great number of methods have been developed, from paper-based methods and hand calculation of nutrient intake in the first half of the 20^{th} century to computer-assisted systems in the second half. In this section some traditional paper-based methods for dietary assessment are described. Some computer technology assisted systems are presented in Section 2.2 and computer-vision based systems in Chapter 3.

One possible purpose of those methods is to accurately estimate the *usual* food intake of a person, meaning the food intake over a long-term period of months or a year. For most research questions a long-term intake estimation is of interest, providing enough data to reach conclusions, detect habitual patterns or defficencies. To improve the methods, trials were conducted of how many days of dietary data are needed to reflect a usual intake and how to evaluate the accuracy of the assessments. Additionally statistical methods for estimating the usual intake of food were developed to reduce the needed data. However the advances made, there is still no easy-to-use and reliable solution for assessing the long-term intake [Stumbo, 2013]. [Shim et al., 2014] provide a review of popular dietary assessment methods for medical applications. [Zhu et al., 2010] give a similar survey. The following paragraphs give a short summary of the methods described in [Shim et al., 2014] and [Zhu et al., 2010]. The methods discussed here are solely self-reporting methods[1], meaning the data of information is reported directly from the respondent, either through actively collecting the data or passively assisting with answers.

[1] Opposed to observational methods of information collection.

2.1.1 24-hour Dietary Recall (24HR)

The 24HR is conducted as an interview of approximately 20 to 30 minutes, where the respondent reports to a trained interviewer from memory, all food and beverages that were consumed in the past 24 hours. The accuracy of the method relies on the skills of the interviewer, but mostly on the memory of the interviewed person, which often leads to under-reporting. The volume of food portions are estimated in reference to a standardised container like a tea cup or a spoon, to help the respondent with the quantification of the food items. With detailed questions from the interviewer about exact ingredients, brands or preparation (e.g. use of oils etc.) of the dishes, the interviewer tries to determine as exact as possible the complete list of food items.

The concentrated information of the interviews are then coded with a food composition database, the volume of the portions converted into actual weights and the macro-nutrient composition of the intake from the analysed period can be calculated, a time-consuming and expensive process.

[Baranowski et al., 2014] add that the method is among the most precise, because it avoids reactivity issues of diaries, like avoiding foods during the assessment. The authors also note that it is necessary to assess multiple days, along with statistical modeling to reach a long-term conclusion.

2.1.2 Dietary Record (DR)

Similar to the 24HR method the food intake from the past 24 hours is being estimated. The respondent has to go through a training process first, and is then entrusted to record all ingredients and foods during the period of one day. The record keeping is to be executed in real-time, i.e. before, during or after the meal. This method therefore does not rely on the respondents memory as much as the 24HR method [Shim et al., 2014]. If executed conscientiously, the estimation will have a very high accuracy. In theory a food diary like the Dietary Record (DR) is the most precise method, though studies show that it also is prone to error, as they have the effect of avoiding certain ingredients or foods from the usual dietary behaviour, when the respondent is knowingly under observation [Baranowski et al., 2014]. Also they are often not completed in real time during the assessment. For children its necessary to provide assistance and supervision to complete the process. Also, the method is time-consuming and the training of the participant causes costs. Keeping a DR is very burdensome on the participant, especially when executed for a longer period of time [Small et al., 2009].

The 24HR and DR methods are not well suited to study chronic diseases, as they are limited to a short-term window of information collection for the data inquiry. For the study of chronic diseases a long-term exposure is of interest. These assessment methods are used for national surveys, etiologic studies of chronic diseases, randomised clinical trials and cohort studies [Shim et al., 2014].

2.1.3 Dietary History (DH)

The protocol for a Dietary History (DH) is to perform a food diary for 3 full days, and an additional check-list of frequently consumed foods. Its necessary to conduct an in-depth time-consuming interview of about 90 minutes. The method was developed by B. S. Burke in 1947 [Burke, 1947]. Due to its time effort and costs it is not used in epidemiological studies [Shim et al., 2014].

2.1.4 Food Frequency Questionnaire (FFQ)

Using this method the respondent fills out an in-depth check-list about which kinds of food he or she consumed and how frequently in a predefined time-frame. The FFQ is more cost-effective than the previously mentioned methods, and they assess a long-term period of time. Therefore they provide information of the usual dietary behaviour. FFQs are a widely used technique in epidemiological studies since the 1990s, because they are much simpler and faster to assess and cheaper to conduct than 24HRs, DRs or DHs [Shim et al., 2014]. FFQs are usually designed for an intended group of respondents, e.g. targeting certain ingredients of the local cuisine or a specific research question, e.g. focus of intake of specific nutrients. This method relies on the memory of the respondent over a much longer period of time compared to the 24HR and DH methods, one of the reason that the reliability and accuracy of FFQs are widely discussed. FFQs are not suited to estimate an accurate measure of calorie intake or macro-nutrient composition.

2.2 Computer aided systems for traditional dietary assessment methods

First computerised systems for dietary assessment started to replace the manual calculations in the 1970s and 1980s [Stumbo, 2013]. With further technological progress, data acquisition became increasingly easier with computers and later mobile systems such as Personal Digital Assistants (PDAs) and mobile phones. Through the revolution of the internet, online systems were developed, that improved accessibility. Further those technologies improved the usability of the assessment process by features such as interaction, visualizations or prompts to remind respondents to submit reports.

Smart devices that exist on the market today are increasingly used by health conscious individuals to track health parameters such as workout regime journaling, tracking ones daily footsteps or estimating the amount of burnt calories [Nabi et al., 2015].

In this section examples of computer-aided systems of traditional dietary assessment methods are introduced. They are grouped into technological application, this categorization is independent of the methodology of the dietary assessment. Systems based on computer-vision technologies are left out here on purpose, as these projects are discussed in more detail in Chapter 3.

2.2.1 Personal Digital Assistant(PDA)-based systems

Starting from the mid-1990s on, PDAs became a popular mobile tool for assessing dietary intake [Forster et al., 2016]. These devices often provided the respondent with predefined drop-down lists of food items, typically ranging from 180 to >4000 ingredients. Evaluations show comparable results with traditional dietary assessment methods [Illner et al., 2012]. Through technological progress and the innovation of smartphones the use of PDA systems decreased.

Wellnavi is an image-based assessment system from Japan from the year 2007. It is based on PDA technology although it works with the technological components of a mobile-phone system, i.e. a camera and network-connection for data transfer [Kikunaga et al., 2007]. Consumed food is photographed by the respondent before and after the meal. Additionally a description of the meal in written form (with a stylus on the devices display) can be given. A dietitian evaluates the respondents dietary assessment parameters based on the observation of the collected photos and the descriptions. For portion size quantification information for the dietitian, a fiducial marker of a fixed size is to be placed next to the food [Illner et al., 2012].

The authors of the Wellnavi project conduct a validity study by comparing their system to the weighted DR method. They report of overall under-reporting of dietary intake using their method, in all test-groups (total of 27 men and 48 women from the general population). They assume the reason for under-reporting to be the low image quality of the Wellnavi system [Kikunaga et al., 2007].

2.2.2 Web-based systems

Automated Self-Administered 24-Hour Dietary Assessment Tool (ASA24) Developed by the National Cancer Institute in the US., ASA24 is a web-platform based on the Automated Multiple Pass Method (AMPM) [Shim et al., 2014], a method that is structured into five interaction steps to assess the food intake of a 24-hour period. The system works with a visualization of each step in the process of creating a food diary and provides digital images for over 10.000 foods and up to eight images per food [Baranowski et al., 2014].

The first step is to report meals and the time they were consumed, after that the meals can be described in detail with all included ingredients. After selecting a food ingredient, the quantification of the item is specified. This step is visualised to help with the estimation of the portion size, that relies on the respondents memory. The assistant for the selection of a portion size for a chosen food item is illustrated in Figure 2.1. Additionally the system asks questions during the passes about forgotten foods and drinks, for every gap of three or more hours between reportings, and asks for a review of the recall before finishing [Baranowski et al., 2014].

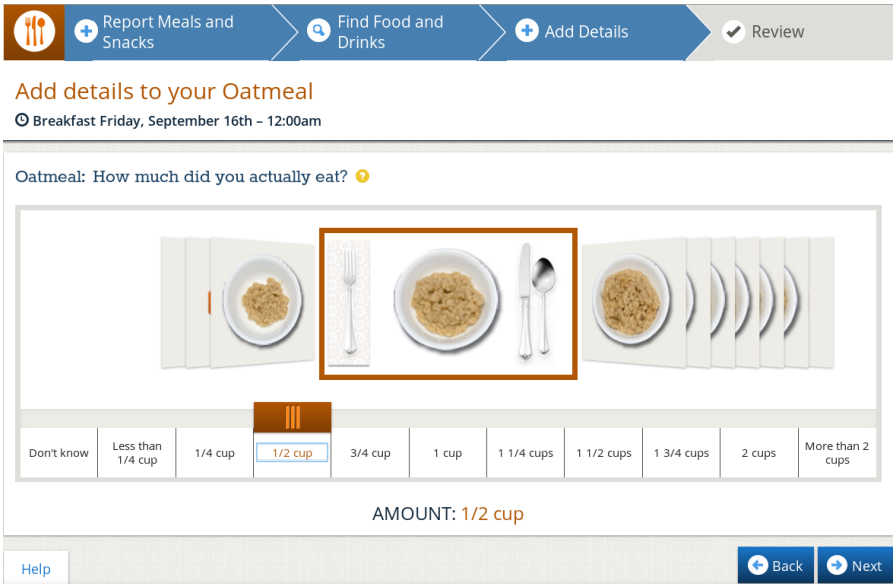


Figure 2.1: ASA24, Assistant for selection of portion sizes.

The ASA24-project can be accessed at `http://epi.grants.cancer.gov/asa24` and a demonstration of the system is available at `https://asa24.nci.nih.gov/demo` [2].

Food Intake Recording Software System (FIRSSt) is an adaptation of ASA24, that is developed specifically for childrens needs. The project is currently in version 4. An animated avatar is added with the goal of keeping focus and interest on completing the process. General consideration on childrens knowledge about food and ingredients was included in the systems design [Baranowski et al., 2014].

DietDay, *NutriNet Sante* and the *Oxford WebQ* are further examples of web-based 24HRs [Forster et al., 2016]. The Computer Assisted Personal Interview System (CAPIS) [Shin et al., 2014] is a Korean open-ended web-based assessment tool. An example of a development of an online FFQ is *Food4Me* FFQ, that was recently developed across seven European countries and has been translated into six languages. The design is similar to traditional paper-based FFQs, and evaluations showed good agreement with the paper-based method. The recently developed *GraFFS* FFQ has a more interactive approach, presenting illustrations of food items to choose from [Forster et al., 2016].

[2] accessed October 10, 2016

2.2.3 Smartphone-based systems

Rising availability and use of smartphones among a variety of age groups enable low-cost and large-scale potential for convenient real-time data acquisition [Forster et al., 2016]. Through the high number of sensors and features that smartphones are equipped with, such as built-in cameras, global positioning systems (GPS), accelerometers, high-speed microprocessors, portable designs, and connectivity to external devices via bluetooth and infra-red, they represent powerful tools for dietary assessment and research [Sharp and Allman-Farinelli, 2014].

My Meal Mate (MMM) Like the PDA-based systems, MMM is an electronic food diary to support weight loss. The food entries are selected from a 40.000-item food database that include generic and branded items, therefore supports the ability to create your own food entries. The system is also capable of saving photographs, which has the function of memory support for the respondent for entries that are not entered in real-time, but at a later opportunity. No computational analysis is performed on the image data. The study shows that the dietary intake assessment correlates well with a conducted 24HR comparison. Incorrect portion quantification introduced the biggest outliers both in the MMM system and the 24HR method. Figure 2.2 shows a screenshot of the food diary entry page of MMM [Carter et al., 2012].

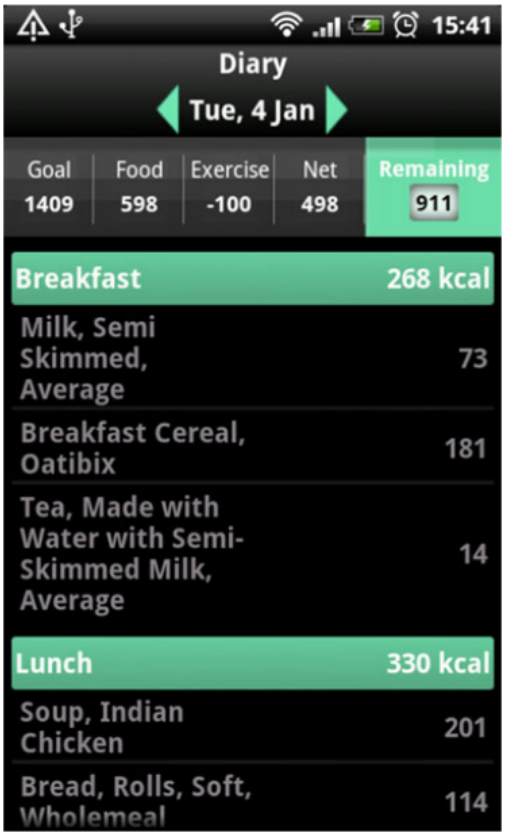


Figure 2.2: A screenshot of *My Meal Mate* illustrating the food entry process [Carter et al., 2012].

Remote Food photography Method (RFPM) has the same system design as Wellnavi but on actual smartphones. An image-based food diary is collected in real-time by the respondent and can be labeled with more detailed descriptions. This data is then sent over the cellular network to be evaluated by trained dietitians. In the study they conducted several trials [Martin et al., 2009]. A comparison of the RFPM method with weighed ground truth values of available food in over a non-consecutive 3-day period showed excellent correlations with calorie intake, but an under-estimation of 88 kcal and 97 kcal (4.7% and 6.6% of

the total energy balance), in one laboratory setting and one other mixed setting of a laboratory lunch and a home dinner. They also determined a satisfaction of participants with the system of 78.8% and a preference over paper-based diaries of 96.6% [Sharp and Allman-Farinelli, 2014].

Nutricam Dietary Assessment Method (NuDAM) also uses images as the reported information, and adds the ability to report voice recordings for more detailed food descriptions [Sharp and Allman-Farinelli, 2014]. The data is evaluated and coded into calorie intake by trained dietitians. Experiments suffered from bad quality of both data. Later the system was extended through adding a daily phone communication between the respondent and the dietitian, for clarification of the data from the proceeding day. In comparison to a weighed food record NuDAM correlated strongly for protein and alcohol measures but poorly for fat intake. The additional phone check revealed many misreportings, the main reason was that the participants simply forgot to take photographs.

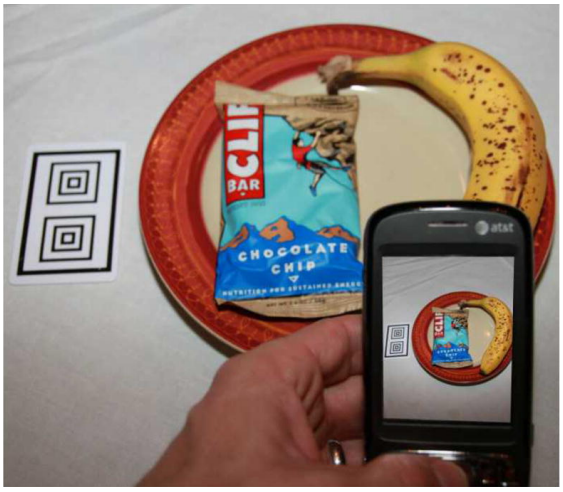


Figure 2.3: The RFPM image-capturing with smartphone [Martin et al., 2012].

Two examples of projects that use crowdsourcing to estimate the calorie content on basis of images, are **Meal Snap** and **Platemate** [Noronha et al., 2011], both discussed in [Stumbo, 2013].

2.2.4 Impacts of technological assistance on the methodology of dietary assessment

Software assisted assessment methods are an expensive investment initially, but through automation may reduce the cost in the long-run, for large-scale systems or studies with many users, because the personnel resources for data collection and organizational tasks are reduced. Computer systems have the advantage to deliver data in real-time, and the collected data is more consistent coming from one objective interface instead from many subjective sources like many different interviewers, they can improve usability providing interactive processes and save time for both respondents and dietitians. The calculation process of the estimation parameters on the basis of the collected data like calorie intake or nutrient composition, is fully automated. The system can be accessed from basically any place with an internet connection and an appropriate device. That also introduces new problems to the assessment process: the respondent has to be capable

of using a device like a computer, tablet or smartphone, and having access to it for the duration of the assessment. Also technical dependencies like battery life of devices can introduce new problems. For experienced users, the ubiquity and simplistic use of a system that is available anywhere and at any time enables the respondent the freedom to focus on the assessment task itself, without organisational overhead or time-costly interview appointments [Shim et al., 2014].

2.3 Inaccuracies of traditional self-report methods

To understand the need of improvement in existing dietary assessment methods and the potential of technologically assisted assessment systems, limitations of traditional dietary assessment methods are analysed and discussed in this section.

Several clinical studies on carbohydrate counting (for postprandial blood glucose estimation) identify inaccuracies from human test groups: [Graff et al., 2000], performed a survey with 350 patients and observed that even patients that regularly estimate their insulin dose obtain problematic inaccuracies. More information on carbohydrate counting can be found in references [4-9] in the work of [Rhyner et al., 2016]. Carbohydrate estimation error should be no more than ±10 g per meal, to maintain postprandial blood glucose control. With a variation of ±20 g per meal, the probability to have a problematic effect on the patient is already quite high. The authors state that there was a one in three chance of hypoglycaemia occuring two to three hours after the meal intake, if an insulin dose for 60 g carbohydrates was given for the actual intake of 40 g of carbohydrates [Smart et al., 2012].

Not just carbohydrate counting, but self-reporting methods for dietary assessment in general are prone to error among adults and more so among children [Baranowski et al., 2014]. In an extensive review of evaluation studies of traditional self-report assessment methods with children, 15 studies (between 1973 and 2009) were identified using the doubly labeled water method[3] for validation, with the majority of the studies having less than 30 participants. A majority of the studies identified a degree of misreporting, eight studies showed significant misreporting ($p < 0.05$). The misreporting comparisons varied from 19% to 41% in 5 DR studies, from 7% to 11% in 4 24HR method studies, 9% to 14% in 3 DH method studies, and 2% and 59% in 2 FFQ studies [Burrows et al., 2010]. The Observing Protein and Energy Nutrition (OPEN) study with 484 participants conducted a comparison of the 24HR and the FFQ methods with the unbiased biomarkers of energy and protein intakes through the doubly labeled water method. For the 24HR method 9% of the men and 7% of the women were underreporters, for the FFQ the comparable values were 35% and 23%. The average underreportings in energy intake was 12-24% on 24HRs and 31-36% on FFQs for men, and 16-20% on 24HRs and 34-38% on FFQs for women [Subar et al., 2003]. Currently available dietary assessment methods make it

[3]Commonly used bio-marking method for asserting energy consumption. Hydrogen and oxygen atoms are replaced by isotopes for tracing.

very difficult to estimate an accurate dietary intake due to the inherent and extrinsic methodological problems. Associations gained from inacurate data between diet and diseases are potentially erroneous [Sharp and Allman-Farinelli, 2014].

Through assisting the process with technological advances, the shortcomings on estimating accurate dietary data of the paper-based methods are not overcome. The inaccuracies are inherent to the methodology of the assessment process of self-reporting. The individual bias of self-reporting recording methods like the DR manifests in under- or misreporting for three reasons: erroneous weighing or estimation, forgetting to report and consciously not reporting. The first reason can be avoided by shifting the estimation to a trained dietitian. Forgetting to report occurs in traditional real-time paper-based food recording methods and we have seen it occur also in image-based systems where participants often forget to take a photo [Sharp and Allman-Farinelli, 2014]. With e.g. mobile phone systems however, prompts can be sent to the participant to remind of the assessment. [Martin et al., 2012] showed improvement of calorie intake estimation with customised prompts over standardised prompts. The third reason of misreporting, the conscious leaving out of certain ingredients might be overcome with additional technological measures that observe the participant without his control, e.g. with wearable camera systems. [Sun et al., 2010] present a prototype for such a dietary assessment system. Although this approach inherently raises privacy concerns, and an unwilling user will find ways to turn the system off in any case. To avoid erroneous self-reporting as one error-factor we have seen systems that report records of images instead of records of ingredients of traditional DR systems. This brings the burden to the dietitian that is evaluating the imagery data. To ease and assist this process, automated and semi-automated systems of image-analysis of food images are currently being researched in many projects (introduced in Chapter 3).

2.4 Contributions and feasibility of computer-vision assistance to dietary assessment

Inaccuracies such as under-reporting or lack of reporting occur as a human factor in application of paper-based dietary assessment methods [Baranowski et al., 2014]. Technological assistance of the paper-based methods through automation of some processes does not bypass the limitations of self-reporting.

Through interaction and communication capabilities of smartphone-based systems, some responsibility of the respondent can be moved to a dietitian, by including an expert in the assessment process. There are a variety of studies in the literature that show improvements of image-supported dietary assessment compared to traditional assessment methods. [Gemming et al., 2015] give an extensive survey of ten image-assisted assessment methods. The image acquisition in most studies was generated from mobile cameras but also systems with wearable cameras were included. The results show that images are able to reveal unreported foods and identify misreporting errors of the traditional methods. The authors determine that image-reporting methods applied as the primary

method of assessment is able to provide valid intake estimates, however can be prone to under-estimation if the images suffer from inadequate quality. [Sharp and Allman-Farinelli, 2014] conducted a similar survey of 16 studies, eight based on photograph analysis by trained dietitians, six were automated image-analysis studies, the other two were self-directed image-based (with no external or automatic analysis of the images, the images serving solely as a memory support for the respondent). The authors identify problems with photo quality and angle and missing photographs in five of the eight image-based dietitian directed studies, as difficulties for the dietitians. They suggest a backup recording method for forgotten photos, or prompts to the phone as a reminder to assess meals. Assuming high image quality, the authors determine a potential of shifting the responsibility from untrained respondents to more objective trained dietitian to improve portion quantification.

Those insights suggest that computer-vision methods for dietary assessment have valid prerequisites to be able to compete with results of human self-reporting methods, as it shows that when documenting the foods with images, these images contain information that can potentially lead to more accurate results.

Computer-vision algorithms work with the input of data produced by visual sensors like digital cameras that produce digital images. A dietary assessment system therefore starts out initially having to its disposal one or more digital photographs of food objects. A question that has to be asked is: how much of the information that we want to assess, the food ingredients and their quantification, is actually contained on the visual data. This data can be 2-dimensional (2-d) or even 3-dimensional (3-d), if a 3-d scene is reconstructed from multiple 2-d images or recorded with a 3-d sensor. In any case, what visual camera sensors can detect are the surface of the objects, but information of the inner structure of the object is not included. For images of food, that means that ingredients can occlude other ingredients by covering them. For example a photograph of a bowl of chicken soup may not contain that much information about the actual quantity of chicken meat inside the bowl, as some parts of it are occluded by the surface of the soup and other ingredients. These are natural limitations to the recognition and quantification estimation on the basis of digital photographs. Additional information from the user or additional sensor data could be acquired and may improve the estimation. The statistical models of the machine learning process, will be generally improved by providing more training data to the learning process.

The overall goal of a technologically aided system is to perform better than humans at the task, or supporting the assessment process for an improvement in usability and/or time effort. As we have seen humans tend not to perform self-reporting tasks very accurately. Not many evaluations of the accuracy of fully automated computer-vision approaches compared with traditional assessment methods or unbiased biomarkers of energy intake through the doubly labeled water method exist at this point. Evaluations of the computer-vision algorithms usually base their accuracy on the ground truth of the annotated data they use, not on comparable dietary information of *competing* methods. [Rhyner et al., 2016] claim that to the best of their knowledge, their study is

the first to give a comparable study of an automated dietary assessment system together with end users. Future development in the active research community of computer-vision systems for dietary assessment may soon show more evaluations of automated computer-vision assessments compared with traditional methods.

CHAPTER 3

Related Work

In this chapter some of the most prominent works on dietary assessment that use computer vision techniques on food images are presented. The goal of the analysis of the projects is to identify the most successful methods used for food recognition and classification. To this end, the reported results of the experiments of the works presented in this chapter, are structured into the methodological approaches in Chapter 4, were they are also compared, with consideration to the details and the data that was used in the specific approaches.

3.1 Mobile phone Food Record (mpFR)

The mobile phone Food Record (mpFR)[1] was developed by the Technology Assisted Dietary Assessment (TADA) group at the Department of Foods and Nutrition at Purdue University in the United States. [Zhu et al., 2010] report an image analysis system that automatically segments, recognises the food items and quantifies the volume of the identified items from images taken by a mobile phone. mpFR is designed as a client-server architecture, shifting the computationally expensive image analysis tasks to the server side. Images are captured before and after the food consumption, and on the basis of those images the total energy of the items is determined. The system is designed for images only from a controlled environment of a dark background (black table cloth) and the placement of a fiducial marker for colour correction and estimating scene dimension parameters. Figure 3.1 shows an example image, and also the ideal segmentation of the items is illustrated. The lighting conditions vary in the used dataset. The authors follow the paradigm of hand crafted descriptor extraction and BoF-encoding

[1] Inconsistently named by various authors of the TADA group, in other works also referred to as *mobile device Food Record* (mdFR) and *mobile Food Record* (mFR).

them before classification. [Zhu et al., 2011] focus on the segmentation method using *Normalised Cuts* [Shi and Malik, 2000], [Bosch et al., 2011b] concentrate on the details of the employed descriptors which are a combination of global and local descriptors. For global colour descriptors the first and second order statistics for each resulting segment of the segmentation process is computed for each channel of the colour spaces RGB, HSV, C_b and C_r from YC_bC_r and the colour opponent dimensions a and b from Lab. Further they compute statistics of entropy [Shannon, 1948] and predominant colours as descriptors and the averages of responses of a Gabor filter from divided blocks of the segments (discussed in more detail in Section 4.1.2). For local descriptors the colour statistics, entropy and Gabor filter responses are computed from local patches. Additionally they chose the Scale Invariant Feature Transform (SIFT) descriptor (described in Section 4.1.2), Tamura perceptual features descriptor [Tamura et al., 1978], a descriptor constructed from Haar-wavelets based on the Speeded-Up Robust Feature (SURF) descriptor [Bay et al., 2008], Steerable filters [Freeman and Adelson, 1991], and DAISY descriptor [Tola et al., 2010]. The descriptors are classified individually and combined by majority vote rule. The global descriptors are classified with Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel, the local descriptors are encoded following the BoF principle (described in Section 4.2.1) and also classified with SVM. The experiments ran on a very small scale dataset of 179 images that contain a total of 39 different foods. The images were obtained under controlled conditions from nutritional studies conducted at Purdue University, [Bosch et al., 2011a]. The experiments ran on hand segmented images, instead of the resulting segmentations of the mpFR algorithm, to isolate the recognition performance from the segmentation estimation results. The results show a general superiority of the colour features over the texture features. The best performing colour features were the local and global colour statistics with 79.2% and 78.6%, and the global colour entropy descriptor with 78.2%. The best performing texture features were SIFT, Haar-wavelet and DAISY descriptors with the accuracies of 65.2%, 64.1% and 60.3% respectively. The accuracy of the descriptor combination was 86.1%.

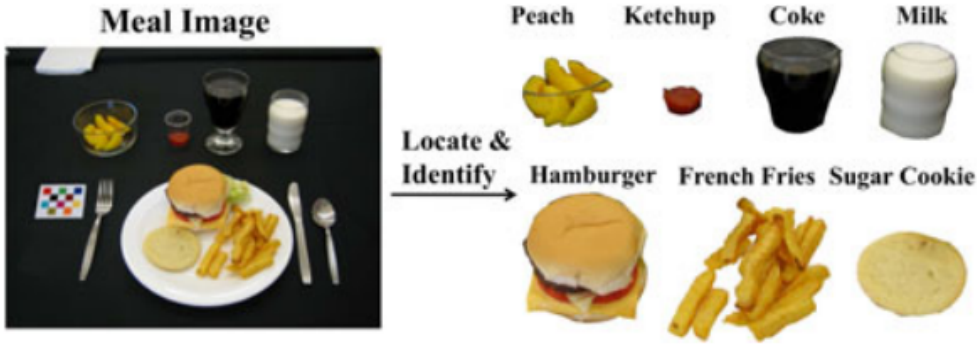


Figure 3.1: The ideal segmentation and recognition of the mpFR-system [Zhu et al., 2015].

[Lee et al., 2012] evaluate the mpFR in a trial conducted with 15 adolescents in semi-

controlled living conditions, by comparing the results with intake values determined through weighting of each food item. Each participant assessed three meals, 19 unique food items including beverages were chosen for the trial. The result for the mean intake over the three meals was 3588 ± 180 kcal, a considerable overestimation, the weighted ground truth was 2723 ± 51 kcal. About 50% of the items were estimated within a 15% margin of the true energy values. In [Boushey et al., 2015] studies have been conducted to explore the attendance of adolescents to record their food with the mpFR.

More recently [Zhu et al., 2015] refined their recognition method but stayed with the hand crafted feature and BoF-encoding approach. The novelty of the presented work compared to the previous of [Bosch et al., 2011a], is an iterative multi-pass flow of segmentation step and recognition step, using the recognition estimation for refinement of the segmentation step, and creating multiple hypothesis of the segmentations. In addition to the descriptors of their previous approach in [Bosch et al., 2011a], three global texture descriptors and three SIFT descriptor variants (Section 4.1.2), computed on each channel of the RGB colour space individually, were compared in the study. The three texture descriptors are Gradient Orientation Spatial-Dependence Matrix (GOSDM) which describes a spatial relationship between gradient orientations [Haralick et al., 1973], the Fractal Dimension estimation (EFD) descriptor which is based on multifractal analysis and Gabor-based image decomposition and Fractal Dimension estimation (GFD), that uses Gabor filters. Detailed information on these three descriptors and how the TADA authors constructed them was published in [Bosch et al., 2011c]. The small scale dataset for the experiments contain an average of 30 segments from food images that contain complete meals (multiple food items), with 83 different classes considered (79 food classes, the others were utensils, glasses, plates, and plastic cups). The results of their classification show that the added global texture features perform worse than their colour features and the SIFT descriptor (detailed results are listed in Table 4.3). For combination of the descriptors they used a late fusion approach, combining the features after classification. Two methods were implemented, majority voting and comparing the confidence scores of the classifier. The results were in the range of 70% and 74% for the different combinations of classifiers (K-Nearest-Neighbour (KNN) and SVM were used) and fusion methods.

3.2 National University of Taiwan

The system reported by [Chen et al., 2012] from the national University of Taiwan is similar to the TADA system. It is implemented as an Android application, performs quantity estimation through a 3D sensor, and food recognition is performed by descriptor extraction, encoding and classification with SVM. They use a dataset of 50 categories of chinese dishes, each category is represented by 100 images. The descriptors extracted for texture are SIFT and Local Binary Pattern (LBP) (described in Section 4.1) and are coded with a method called *sparse coding* [Yang et al., 2009], which bases its image representation on a linear combination of descriptors from *visual words* of a dictionary of descriptors. Third texture descriptor used is the concatenations of the means and the

variances of the Gabor filter(Section 4.1) responses with six orientations on 5 scales, from 4×4 blocks of the divided image. The descriptor was not encoded before classification. For colour description colour histograms are computed for a 4×4 grid of the image. A histogram for each of RGB channel quantised into 32 bins, and concatenated which results in a 1536 dimensional vector. For evaluation 5-fold cross validation is adopted, the overall recognition rate achieved was 68.3%.

3.3 Type 1 Diabetes Self-Management and Carbohydrate Counting (GoCARB)

GoCARB is a novel system that aims to estimate carbohydrate content with an error less than twenty grams per meal, designed especially for type 1 diabetes patients [Rhyner et al., 2016]. GoCARB is a *Marie Curie Industry-Academia Partnerships and Pathways* project, funded by the European Commission's 7th Framework Programme. The projects aim is an automatic carbohydrate assessment of meals in a controlled setting, realised in a smartphone application that analyses recorded images by the respondents and outputs the estimated insulin bolus dosage using the USDA database [United States Department of Agriculture, 2016]. The image-analysis components of the GoCARB project are: plate detection, food segmentation, food recognition of each segment, 3-d model reconstruction, volume estimation of each segment [GoCARB Project, 2016] (illustrated in Figure 3.2).

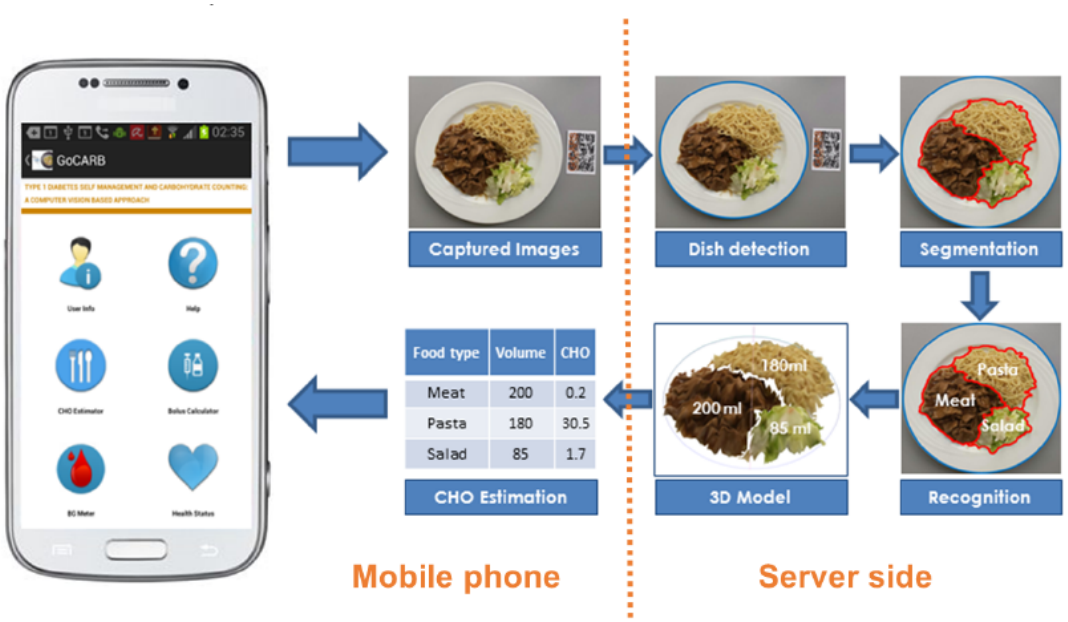


Figure 3.2: Overview of the GoCARB system, with its image-analysis components [Rhyner et al., 2016].

The assumptions the GoCARB system is based on, are 1) that there is exactly one dish visible, 2) the food is inside a round plate and 3) there are no occlusions among the food [Rhyner et al., 2016]. Those assumptions reduce the applicability of the system for real-world images vastly as all foods with ingredients mixed with each other, are uncovered by the systems recognition abilities. The contribution of the project is a thorough execution of a working prototype with an extensive evaluation of the assessment methodology.

The project started out in September 2011 and was initially planned for 48 months. The following paragraphs give a summary of the productive output of the GoCARB research team.

In [Anthimopoulos et al., 2013] the GoCARB authors present their first prototype. The dish detection and segmentation methods were improved in the later published [Dehais et al., 2015]. For recognition they follow the paradigm of hand crafted feature extraction and classification, the method of encoding the local descriptor is not mentioned. The authors use one feature for texture description, LBP (described in 4.1.2) and one colour feature, a histogram of the 1024 most dominant food colours through clustering the colour space with hierarchical k-means. After combination of the two features to one vector of 1280 dimensions, they use a SVM classifier with RBF kernel to distinct between six classes of foods. To evaluate their method 5000 images from the internet were manually annotated and used for training and testing the approach. In a 10-fold cross-validation they report an accuracy of 87% for the six class problem. For the quantification estimation of the carbohydrate content a detailed description of their volume estimation method can be found in [Dehais et al., 2013].

In [Anthimopoulos et al., 2014] the authors extend their experiments comparing 14 different local image descriptors, the encoding method BoF and extensive investigation of its components and parameters, and six different classifiers. The colour image descriptors they analyse are five Colour Histograms, Generalised Colour Moments and Colour Moment Invariants and for texture description they use the SIFT descriptor and six colour variants of the SIFT descriptor. A detailed description of the image descriptors can be found in Section 4.1. The classifiers compared were SVMs (with three types of kernels: linear, RBF and χ^2), a feed-forward Artificial Neural Network (ANN) and a Random Forest (RF). The highest classification rate was yielded computing the BoF dictionary with 100.000 patches for each of the eleven food classes, using 10.000 visual words in the dictionary and using the SIFT_{HSV} descriptor. An Overall Recognition Accuracy (ORA)[2] of 77.6% was reached using a linear SVM for classification. 4868 images were used, 60% for training, 40% for testing.

In [Anthimopoulos et al., 2015] they present a first evaluation of the complete system with all its components, including the segmentation and volume estimation, as illustrated

[2] $ORA = \frac{\sum_i^N CM_{ii}}{\sum_i^N \sum_j^N CM_{ij}}$, where CM_{ij} is the number of images that belong to class i and were classified in class j, and N is the number of classes, [Anthimopoulos et al., 2014, p. 1265]

in Figure 3.2. For the image descriptors and classification they use the same setup as in [Anthimopoulos et al., 2013]. For their purpose of carbohydrate estimation they define nine broad food categories[3] pasta, potatoes, meat, breaded food, rice, green, salad, mashed potatoes, carrots, and red beans. 1620 images from 248 multi-food served meals were acquired under controlled conditions: a reference card (for the volume estimation) was placed next to the dish, the foods do not overlap, the lighting conditions were the same on each photo, the food was placed on elliptical plates and the angles the photos were taken from were consistent. For the evaluation a set of 24 different dishes was selected and the carbohydrate content estimated by the GoCARB system and compared to the real values determined through weighing. The average mean error over the 24 dishes was 6 grams (10%) with a standard deviation of 8 grams (13%), which is well in the aim of the variance of ±20 grams per meal. In fact 95.5% of the dishes were within an error range of ±20 grams. They detect motion blurring as a factor for deviations.

[Dehais et al., 2015] propose an approach for dish detection with Canny filter edge detection, followed by filtering to eliminate junctions between edge curves, sharp corners and small segments. Of the left-over segments outliers that do not support an elliptical model, are removed with Random Sample Consensus (RANSAC). The segmentation method discussed is an automatic method based on the Seeded Region Growing (SRG) algorithm with distance measure in CIELab colour space (focuses less on intensity changes that are often caused by shadows). They propose an automatic and a semi-automatic segmentation and evaluate both methods. The automatic algorithm distributes seeds on a regular grid within the designated dish area. The grown regions produced by the SRG algorithm are then combined with Statistical Region Merging paradigm (SRM), which merges iteratively the two regions with the smallest cost based on ratio of colour distance. The semi-automatic method is based on user inputs for the seed-centers, acquired through the smartphone interface, and designed for the case where the automatic method fails. The evaluation showed a 99% accuracy for the dish detection, 88% for the automatic segmentation and 91% for the semi-automatic, outperforming [Anthimopoulos et al., 2013] a previous study of the GoCARB group using a Mean-Shift algorithm, Local Variation [Felzenszwalb and Huttenlocher, 1998] and a contour detection approach [Arbelaez et al., 2011]. The average processing time on a Intel i7-3770 CPU was 0.19 seconds for dish detection and 0.45 seconds for the proposed automatic segmentation.

In [Christodoulidis et al., 2015] the GoCARB authors explore DCNNs as an alternative for their hand crafted feature approach. As in their previous system design in [Anthimopoulos et al., 2013], the segmented regions are classified to corresponding food classes. For classification with a Convolutional Neural Network (CNN), overlapping square patches of the segmented regions are classified individually and the majority class is designated. The training is performed from non-overlapping 32×32 pixel patches that are each multiplied 16 times through flip and rotation transformations to extend the training set.

[3] in previous work [Anthimopoulos et al., 2013] they used six categories, in the later work [Anthimopoulos et al., 2014] eleven, and in [Rhyner et al., 2016] the number of categories was reduced to seven.

The authors ran experiments with different network layouts, testing convolutional layer depth of two, three and four, and altering also the sizes of the layers. The net that yielded the best F-score was a net with four convolutional layers and two fully connected layers. Each convolutional layer is followed by a 3×3 pooling layer and a stride of two. A deeper insight on DCNNs is presented in Section 4.3. The experiments were conducted using the deep learning framework Convolutional Architecture For Fast Feature Embedding (CAFFE) [Yangqing, J., 2013] on a single GPU (GeForce GTX 760, 2GB Memory, 1152 Cores). The presented results showed an accuracy of 84.9% of identified food items, a slight improvement compared to 82.2% of their previous hand crafted feature based approach from [Anthimopoulos et al., 2013]. The corresponding average processing times per image for both methods were 0.28 sec and 0.1 sec.

In [Rhyner et al., 2016] the authors conduct an evaluation of the GoCARB system comparing it to self-reported carbohydrate counting of the participants. The assessment with the GoCARB system was conducted over a period of ten days with 19 adult diabetes patients, six different meals from the hospital restaurant were evaluated every day, 60 different meals in the total trial. A meal consisted of three food categories each, the authors define a category being a one *unmixed* food item e.g. rice, chicken or vegetables. The six portions were of different sizes. Each of the participants did one assessment of all six meals on one randomly chosen day, therefore on some days there was more than one individual performing the assessment. The carbohydrate portion of a total of $19 \times 6 = 114$ meals was assessed by the participants, once with the GoCARB system and also on their own via self-report assessment. Each user was trained how to use the system prior to the trial and was assessed additionally with a questionnaire about the usability of GoCARB. The results of the trial were an average absolute error of about 12 grams (26%) of carbohydrates for the GoCARB system versus about 28 grams (55%) error of the participants self-assessment. About 81% of the estimations of the GoCARB system were within the aimed goal of a ± 20 grams variance, where only about 59% of the self-reported estimations reached this criteria. Two participants produced high deviations from the rest of the test group, the authors also ran a Mann-Whitney U-test with exclusion of the participants data producing the biggest outliers (average of 158 grams of over-reporting over the six meals) and show that the GoCARB systems results were still a significant ($P = 0.01$) improvement over the assessed data from self-reporting. A significance test without the participant that produced the second greatest outliers (around 70 grams average over-reporting) was not presented in the study. The resulting distribution of the error including all participants, was broad with outliers up to 200 grams for the self-assessment, the corresponding GoCARB error distribution was more centred around zero and was symmetric. The results of the study for the segmentation was a 75% success-rate for the automatic segmentation, and the recognition was correct in about 60% of the cases for all three food categories, in 36% two out of three were correct and in 4% only one was recognised. The recognition is based on their previous work [Anthimopoulos et al., 2013] combining two image descriptors, LBP and a histogram of the 1024 most dominant colours and classifying the concatenation of the two vectors directly with a SVM. The results of the qualitative questionnaire were high agreement

about the system to be easy to use, and lower agreement about its speed, about which the authors argue that depended mostly on the quality of the network connection.

The GoCARB group basis the insulin bolus solely on the carbohydrate portion. [Smart et al., 2013] show that fat and protein also have a significant effect on the bolus estimation. They conducted a trial where 33 test subjects are examined in a time window of five hours after the meal. They show that different meals with a constant carbohydrate content, but in combination with a high fat content and/or a high protein content, the mean glucose excursions is significantly higher in the later phase of three to five hours after the meal. Thus considering this additive effect to the effect of the carbohydrate content of a meal when estimating the supplementary insulin dose, a more precise insulin dose estimation will be reached.

3.4 FoodCam

[Hoashi et al., 2010] report a food recognition procedure that is based on hand crafted image descriptor extraction without further encoding for all but for the SIFT descriptor, which was encoded using the BoF paradigm. The system was trained on 85 food categories of Japanese food but include also international food categories such as pizza or hamburgers. The descriptors used besides SIFT are a Colour Histogram in RGB colour space, the averages of 24 Gabor filter (four scales and six orientations) responses (discussed in more detail in Section 4.1.2) and the Histogram of Oriented Gradients (HOG) descriptor (detailed descriptions of the descriptors can be found in Section 4.1.2). The best result of the single descriptor was an accuracy of 33.47% with SIFT and BoF encoding. The best result of combined descriptors was 62.52%. The method they used for *feature fusion*, which refers to combining the features, is an SVM with Multiple Kernel Learning (MKL), that uses an individual kernel for each descriptor. For the implementation they use the SHOGUN large-scale machine learning toolbox [Sonnenburg et al., 2006], which supports MKL. The classification rate reached with uniform weights instead of the MKL was 60.87%. They also performed an evaluation with a prototype system under uncontrolled conditions without any instructions to the users (which contained photos in dark environment or photographed from not optimal distance), and classified 45.3% of 785 images correctly.

In [Kawano and Yanai, 2015b] the authors present a mobile phone based prototype of FoodCam, a system for automatic recognition of multiple meals from a photograph, with semi-automatic segmentation of each meal and manual portion size input from the user. The top five results of the classification are displayed to the user for selection. The main focus of this paper lies on the computational efficiency of the system to be able to run solely on a smartphone, without any computation on a server. The segmentation is executed by a manual bounding-box selection by the user. The selections then get refined by the graph-cut segmentation algorithm *GrabCut* [Rother et al., 2004]. Within each resulting bounding-box the food recognition is performed. For recognition, a combination

of BoF-encoding of SURF [Bay et al., 2008] local descriptors and a colour histogram in RGB colour space (slightly modified procedure from previous paper [Hoashi et al., 2010]), are compared with their new approach of FV-encoding HOG patches and colour patches. The FV-encoding is computed with a Gaussian Mixture Model (GMM) with 32 gaussians, the BoF-encoding with a k-means clustered dictionary with the size of 500 codewords. The BoF descriptors are additionally mapped with a χ^2 kernel to a triple of their original dimensionality. Kernel feature maps were proposed in [Vedaldi and Zisserman, 2010] (described in detail in Section 4.2.1). For classification they use one linear SVM for each class, training each SVM with the one-versus-rest manner, where descriptors from one class are positive samples and from all other classes are negative samples. The system design takes advantage of multiple cores of Central Processing Units (CPUs) on modern smartphones(for the experiments a processor of 1.6GHz with 4 cores, running Android 4.1 was used), carrying out computation of descriptors, encoding and classification in parallel. Computation time results of the two recognition approaches including the classification with linear SVM were 0.26 seconds for the SURF and colour histogram computation with BoF-encoding, and 0.065 seconds for computation of the HOG and colour patches and their FV-encoding, making the FV setting four times faster than the BoF setting. The computation times for the encoding procedures were 0.018 seconds (including mapping computation) and 0.0099 seconds (including Principal Component Analysis (PCA) computation) respectively, making FV-encoding 1.8 times faster than BoF-encoding, were the sizes of the resulting descriptors were 500 (1500 after the kernel mapping) and 1536 for the FV. The classification accuracy reached with the SURF-BoF+colour histogram was 42% and with the HOG+colour patch-FV was 49.7% in 100 food classes with a total amount of 12905 images in the dataset, which they called *UEC-FOOD100* (more details in Section 5.2.1). Through incorporating flipped images increasing the training set, the accuracy of the FV was pushed to 51.9%. An extension of the FV parameters to 64 gaussians and without reduction of the HOG dimensionality and incorporating additional spatial coding with the spatial pyramid paradigm [Lazebnik et al., 2006] they reached 59.6%, but was not considered for the mobile use because of computational complexity.

In [Kawano and Yanai, 2014] the FoodCam authors explore the use of DCNNs for food recognition. They pre-train (discussed in Section 4.3.3) their net with the 1000 class ILSVRC-2012 dataset with 1000 images per class as a feature extractor. Following [Oquab et al., 2014] they additionally added another 1000 food-related categories from the ImageNet[4] 21,000 classes database, resulting in 2000 categories for pre-training. The experiments were implemented using the CAFFE library [Yangqing, J., 2013]. They used a net based on [Krizhevsky et al., 2012] (described in Section 4.3.4) with the modification of increasing the sizes of the last convolutional layer and the first fully connected layer, illustrated in Figure 3.3. They extract the previous to the last layer (*Layer7*), apply an l^2-normalisation and perform classification with a linear SVM classifier on the resulting feature vector. For a baseline a HOG and a colour descriptor[5] with FV-encoding is

[4]ImageNet 2011 Fall release

[5]more details of the parameters of the descriptors, can be found in [Yanai and Kawano, 2015]

implemented, which achieve 65.32% and 52.85%, the 2000-category pre-trained DCNN achieves 71.80% and 58.81% for the UEC-FOOD100 and the UEC-FOOD256 datasets respectively. A combination of the FV and the DCNN extracted vector achieve accuracies of 77.35% and 63.77%. The total dimensionality of the FV combination was 57344, of the DCNN extracted vector 6144.

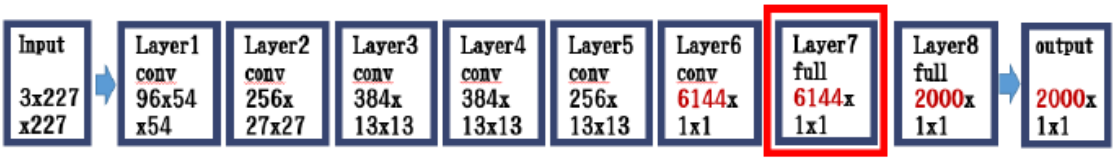


Figure 3.3: Overview of the FoodCam DCNN of [Kawano and Yanai, 2014].

The most recent work [Yanai et al., 2016], explores the trade-off between speed and accuracy of performing DCNN classification on mobile phones. The demo application called *DeepFoodCam* implements a Network In Network (NIN) structure [Lin et al., 2013] to reduce memory and computational time, and was tested on iOS and Android platforms. They report an average computation time for classification for one image of 66.6 ms on an iPad Pro, with an accuracy of 78.8% on the UEC-FOOD100 dataset.

3.5 Im2Calories

[Myers et al., 2015] from Google describe a mobile system for food recognition and quantification with the goal of macro-nutritional composition estimation from a single image. Instead of full automation the goal is to offer smart *auto-complete* functionality, considering the difficulties of food recognition. Two problem definitions are distinguished for the experiments: the first assumes the image of the food items to be from a public restaurant menu, the second approach tries to solve the problem of discriminating between 201 generic food categories (a variation of the FOOD-101 dataset, described in Section 5.2.1, structured into more detailed categories).

As a first step a food detection is performed, with a DCNN structure of the *GoogLeNet* model (Section 4.3.5), utilizing the FOOD-101 as the *food* category and adding 100000 images from ImageNet for the non-food category. The classification accuracy reached for the binary problem is 99.02%.

For the restaurant specific approach, the closest restaurant to the current GPS location of the used device is looked up via Google's Places API[6]. A DCNN of the *GoogLeNet* model was trained and used as a multi-label classifier to support recognition of multiple menu items (dishes) on one image. This is done by comparing to a threshold of the probability of each value in the output layer. The dataset of [Beijbom et al., 2015] with images from three restaurants was extended to cover 23 restaurants in the US, with a

[6]Google places API. https://developers.google.com/places/

total of 2517 food items. The images were collected by a Google search on social media platforms *Yelp*, *Flickr*, *Instagram*, *Pinterest* and *Foodspotting* to raise the probability of user-generated photos. 270000 images were verified with Amazon Mechanical Turk and resulted in a set of 99000 images of 2517 menu items. The DCNN was pre-trained[7] and then fine-tuned on the FOOD-101 dataset. Only the final layer was then trained with 75% of the *Restaurant* dataset, to adapt to the categories used for classification. The error rates of the classification of the test data were very high. They do not publish the exact values, but show a figure where the top-1 error rates of the individual restaurants are illustrated. They are located roughly in the range of 0.5 and 0.77. The high values are attributed to the fact that the restaurants have many categories that are almost the same (e.g. items like Quarter Pounder Deluxe Burger and Quarter Pounder Bacon Cheese Burger). To reach a better result they then merge the most confused classes.

For the approach to classify generic food images, the FOOD-101 dataset was used again. 50000 of the 101000 images were selected, and newly classified with Amazon Mechanical Turk, after which emerged a total of 201 main categories (occurrence of minimum 100 instances for each category). Again a DCNN is used as a multi-label classifier. The mean Average Precision (mAP) inside the 101 original categories was 0.8, but outside only 0.2. As a reason for that the authors point to the sparseness of the occurrence of these items, as they are mostly side-dishes. The average mAP was 0.5.

They also perform classification on the *original* FOOD-101 dataset with the pre-trained GoogLeNet architecture (described in Section 4.3.5) and fine-tune with the FOOD-101 training set. On the test set a 79% top-1 accuracy is achieved.

For size estimation, a segmentation of the food items is performed. The segmentation algorithm *DeepLab* is used, an adoption of [Chen et al., 2014]. A semantic segmentation method that combines the classification scores, with low-level information of pixels and edges, using a Conditional Random Field (CRF) graph. For the quantification a proposal of [Eigen and Fergus, 2014] is followed, a depth prediction performed by a multi-scale DCNN architecture based on the *AlexNet* model (Section 4.3.4, [Krizhevsky et al., 2012]). The obtained depth-map is then converted into a voxel representation for volume calculation of the segments.

There was no evaluation of calorie estimation performed. According to the authors this was due to not having sufficient nutritional composition information, as no database was found with a coverage broad enough for all categories.

3.6 IBM

[Wu et al., 2016] from IBM Research, propose a DCNN based food recognition framework that incorporates the semantic relationship among food classes. A hierarchy of semantic groups of food items from the FOOD-101 dataset (Section 5.2.1) was defined. The

[7] The net was pre-trained with the 1000 ILSVRC-2012 categories from the ImageNet dataset.

semantic hierarchy is incorporated into the learning process by basing the loss function of the DCNN on a *multitask-learning*, considering each level of the semantic hierarchy. A method of *label inference* is also implemented, which incorporates an influence in decision-making of the final category by considering the membership of the parent structures. The authors observe that in cases of misclassification, decisions fall into the same semantic category more likely than with the non-hierarchical approach, because the model produces more semantically coherent predictions. [Wu et al., 2016] argue that this behaviour is useful for a nutrition information estimation application, since semantically close predictions in case of misclassification provides more relevant nutrition estimation than an entirely unrelated prediction. For the base network structure of the DCNN they adapt *GoogLeNet* [Szegedy et al., 2014] (Section 4.3.5). The reported accuracy with their approach of hierarchical semantic learning was 72.11%, compared to an accuracy of the unchanged *GoogLeNet* architecture that achieved 69.64%.

3.7 Menu-Match

[Beijbom et al., 2015] from Microsoft Research follow the approach of recognition of restaurant specific food items. Location information (e.g. GPS) is used to reduce the selection to a set of images of items from nearby restaurants. The goal is to estimate the calorie value from single images. The approach is specifically designed for standardised food items from restaurant menus, therefore does not consider varying sizes or ingredients.

The authors collected a dataset from three restaurants with a total of 646 images, 1386 tagged food items in 41 categories. Ground truth calorie meta data was estimated for each of the 41 items by a dietitian.

The implementation is based on the BoF approach of hand crafted features. Specifically they extract SIFT, HOG, LBP, colour and MR8 [Varma and Zisserman, 2005] descriptors and apply Locality-constrained Linear Coding (LLC)-encoding. Classification of the individual descriptors is performed with linear SVMs[8]. The results are combined with a late fusion strategy of all scores of the SVMs, by training another linear SVM on the fused data.

Its reported that the colour feature discriminates best, followed by the MR8 texture descriptor. HOG and SIFT achieve worse discrimination of the used food images. The evaluation of the proposed system resulted in an absolute error of 232±7.2 kcal (mean ± standard error) on their restaurant dataset. On the generic dataset with 50 classes used in [Chen et al., 2012] a recognition accuracy of 77.4% was reached, an improvement compared to Chen et al., which achieved 68.3%.

[8] A total of 205 linear SVMs were trained, one for each descriptor and class.

3.8 Analysis

In the following sections **applications** (task, research or assessment question), **assumptions** on the composition of food items on the images, the **granularity** of the categories and **level of automation** are analysed.

3.8.1 Application

From the introduced projects, following common general tasks are identified:

- Category Identification
- Caloric estimation
- Specific macro-nutrient estimation (e.g. carbohydrate content)
- Complete macro-nutrient composition (ratio of protein, carbohydrates and fats)
- other nutrient composition, e.g. specific micro-nutrients in epidemiological studies [Shim et al., 2014] (more detailed ingredient list needed).

Observing the current works on dietary assessment and food recognition in general, a trend exists to solve particular subdomains of the problem. A strong recent trend is to identify food items purchased from restaurant chains. [Beijbom et al., 2015] argue that especially for food consumed at restaurants it is difficult for users to estimate the caloric value of the items, as they are unaware of preparatory details, e.g. how much oils or other fats were added. [Myers et al., 2015] work on an extension of the Menu-Match dataset [Beijbom et al., 2015]. [Bettadapura et al., 2015] presented a similar system that also leverages the restaurant context. This particular subdomain promises a high accuracy e.g. compared to generic food recognition, as the food categories are limited to the size of the menu and portion sizes and ingredients tend to be standardised in most restaurants, therefore a good estimation of the nutritional composition can be calculated (e.g. by a dietitian) for each item. [Beijbom et al., 2015] report that their approach is extendible for food from take-out or delivery, by adding a manual restaurant selection from the user.

Canteen food is a similar example of a specialised subdomain. The GoCARB group (Section 3.3) reduce their carbohydrate assessment solely to images from hospital canteen restaurants. [Ciocca et al., 2017] recently developed a similar dataset called *UNIMIB2016*, including over 1000 images of canteen trays with multiple food instances per tray.

Others, reduce the problem to a subdomain that considers categories of food of a specific geographical region only, such as in the work of [Chen et al., 2012], that limit their system to Chinese-food categories.

Another approach is to focus on the computer-vision perspective, such as [Kawano and Yanai, 2014], reducing the task to the categorization of food images into the predefined categories[9].

3.8.2 Assumption of Visually Separable Items

There is a general distinction of two approaches, for defining the recognition task of food related problems. The first approach identifies one dish as an atomic object, the second approach identifies separable items (e.g. ingredients) within one object (e.g. by performing segmentation).

The decision for the approach is influenced by the expected image data, and also affects the available nutritional meta data that is necessary to map the visual information to the assessment meta data.

The GoCARB group (Section 3.3) assume clear visual separable food items present on their images and use segmentation of the individual items. The GoCARB system performs well in estimating carbohydrate content of a meal, within their desired limits of ± 20 grams. But the system is limited to images that follow strong assumptions, their experiments are similar to a near laboratory environment. [Dehais et al., 2015] argue that such assumptions produce a system insufficient to deal with meals of arbitrary content and portions, and only works for meals with specific composition and sizes (e.g. fast food restaurants or cafeterias). Dietary assessment systems towards open-world food categories is only realistic, if handling a high number of complex food items, as for most dishes a simple segmentation is not feasible.

The two approaches map two different kind of food preparations, that can be observed when taking a closer look at food photographs. One way of preparing dishes is to mix ingredients into one *connected* entity. Another way of meal preparation is to combine multiple separable *disconnected* food items on one plate next to each other. Examples of the two *dish designs* are illustrated in Figure 3.4.

The approach directly affects the number of categories, as there are theoretically unbound number of combinations of ingredients (dishes), but a smaller number of *raw* ingredients. Therefore the design choice defines the necessary structure of the database used for mapping to nutritional information. Inherent to the *atomic* approach is an extensive database of meta data for whole dishes.

An ideal estimation would be to extract a list of every single ingredient, and use a calorie/nutrition database to map the visual information to the meta data that we want. The standard source for mapping *raw* ingredients to calorie values is the National Nutrient Database (NNDB) of the United States Department of Agriculture (USDA) [USDA, 2016], [Rhyner et al., 2016, Myers et al., 2015]. However, mapping visual information to

[9] In other reports such as [Kawano and Yanai, 2015b] it is mentioned and illustrated that FoodCam is estimating calorie values, but there is no description of how these are estimated.

(a) Mixed, *connected* ingredients.

(b) Separated, *disconnected* ingredients.

Figure 3.4: Illustrate two dish designs of different preparation style. All four images are from the FOOD-101 dataset, from the category *chicken-curry*.

a ingredient-based database is hard and inevitably inaccurate [Beijbom et al., 2015], as discussed above, strong assumptions for the image generation are necessary.

For *atomic* food-dishes [Myers et al., 2015] suggest to use an extension of the NNDB: Food and Nutrient Database for Dietary Studies (FNDDS), which includes nutritional information for whole dishes. However e.g. the calorie content depends a lot on the exact preparation of the food (e.g. grilling versus frying) [Myers et al., 2015]. Consequently, for a good estimation each preparatory detail occurring in the data, would ideally have a mapping in the database. Another issue are large deviations within an *atomic*-dish category (e.g. one hamburger may differ radically from another in calories if not specified more precisely) [Beijbom et al., 2015]. For dietary assessment with fine-graded visual categories, there is a need for the same graduation in the nutrition database (E.g. it is more likely to have a more accurate estimation, if the domain *hamburger* has subcategories

of details like *with/without cheese* or *one or two slices of meat*, etc.). Due to this fact a database with broader coverage of prepared foods is needed [Myers et al., 2015]. Because of the difficulties of fine-graded recognition, a semi-automated routine with interaction steps by the user for verification and selection could increase the performance.

[Shim et al., 2014] discuss the distinction of separable ingredients versus atomic dishes on the assessment level, comparing food-ingredient based versus dish based FFQ assessments[10]. The conclusion is that the food based FFQ tends to underestimate dietary intake more than a dish based FFQ. The explanation is that in the assessment of individual food items, the preparation specific ingredients for the dish (various seasonings, like salt, sauces, pastes, oils...) are not considered enough. Those ingredients highly contribute to nutrients (e.g. energy, fat, sodium or β-carotene intake). One strategy for dealing with challenges of undetectable ingredients from images could be resorting to recipe databases (in combination of interaction with the user for confirmation or selection of possible ambiguities).

[Zhu et al., 2015] compute a contribution ratio of single descriptors to the total classification result that is achieved by decision fusion of all descriptors, in their approach of combining local and global descriptors. When classifying images with more complex foods, the contribution of local features increases, whereas global descriptors cover the description of simpler food items. That means, when choosing connected ingredient food categories (whole meals) for what constitutes its own class in a food recognition system, then descriptiveness of local descriptors is going to be higher than for a system that classifies disconnected ingredients only.

An alternative for segmenting individual items of one dish, or for segmenting the image in case of multiple category occurrences on one image, is the use of a multi-label classifier as e.g. performed by [Myers et al., 2015].

3.8.3 Granularity of Semantic Assessment Categories

The definition and number of categories is influenced by the research question/assessment application. Depending on the desired meta data of the assessment, multiple categories or subdomains of food items can be grouped together into one semantic category.

In [Rhyner et al., 2016] the GoCARB group is classifying into only seven categories of food types to estimate the carbohydrate content with an average error of 12 grams (49 kcal) carbohydrates per meal, for a total of 60 different meals (combinations of three separable food items per meal). They reduced the problem to a minimum number of food types that share a similar carbohydrate density, as there is no benefit to differentiate between categories of similar carbohydrate content (e.g. different categories of meat). This problem-reduction works if fine-grading into subcategories, does not increase the accuracy of the assessment information.

[10]The discussion of the report is in the context of asian specific food.

Other applications where categories could be grouped (do not have to be fine-graded), are assessments similar to the FFQ (Section 2.1.4), where the goal is a general survey of a patients diet. E.g. the occurrence of tendencially considered *unhealthy* categories like *cake* or *burger* would be sufficient, and the exact composition would not alter the assessment evaluation.

The task of full nutritional macro- or micro-composition estimation could be referred to as the holy grail of food recognition tasks, as there is a necessity of a certain detailed visual granularity to match the detail of the necessary meta information assessed.

3.8.4 Full / Semi Automation

Another approach for increasing the accuracy are semi-automated systems, where user input is incorporated into the recognition routine. As suggested in [Beijbom et al., 2015], hybrid interfaces could reduce the barrier to food logging: [Branson et al., 2010] study the incorporation of human interaction in the visual recognition process, concluding that it can drive up the recognition accuracy to levels good enough for practical application. [Myers et al., 2015] propose to minimise user effort by offering smart *auto-complete* functionality, rather than complete automation. [Beijbom et al., 2015] also suggest for future work to incorporating user-specific customization, such as learning a user trend over time as priors in the inference model.

3.8.5 Summary of the projects

For the state-of-the-art methods, the most used techniques are:

- hand-crafted feature extraction and BoF encoding, used in [Hoashi et al., 2010, Bosch et al., 2011a, Chen et al., 2012, Anthimopoulos et al., 2014, Zhu et al., 2015, Kawano and Yanai, 2015b]
- hand-crafted feature extraction and FV encoding, used in [Bossard et al., 2014, Kawano and Yanai, 2014, Kawano and Yanai, 2015b, Zhu et al., 2015]
- feature extraction and classification with DCNNs, used in [Kawano and Yanai, 2014, Myers et al., 2015, Wu et al., 2016].
- sparse coding [Chen et al., 2012]
- LLC-encoding used by [Beijbom et al., 2015]

Table 3.1 shows an overview of the described projects from this chapter.

Project	Segmentation	Volume Estimation	Recognition Method
mpFR	normalised cuts	from single image	BoF
GoCARB	region growing	3-D reconstruction from 2 images	BoF, DCNN from patches
FoodCam	manual bounding box selection	none	FV and DCNN
IBM	none	none	DCNN
Im2Calories	DCNN	multi-scale DCNN	GoogLeNet-DCNN
Menu-Match	none	none	LLC

Table 3.1: Overview of the general approaches of the described projects.

Results from the state-of-the-art research on food recognition reach accuracies from 87% [Anthimopoulos et al., 2013] and 84.9% [Christodoulidis et al., 2015] and 86.1% in [Bosch et al., 2011b] for experiments in controlled environments and with limited variety of food items (between 6 and 39 categories). In experiments closer to real-world conditions, conducted on generic food images generated by users (e.g. from social food platforms), the results are in a range of 77-79% [Kawano and Yanai, 2014, Myers et al., 2015] in 100 food category benchmarks. Evaluations from user studies for calorie estimations from images as the only information source, exist only in very limited number. [Lee et al., 2012] and [Anthimopoulos et al., 2015] present such evaluations (as presented in Sections 3.1 and 3.3), but with very limited food categories and image data, for both training and evaluation. A fully automated system for intake estimation is currently not feasible for satisfying results in a real-world application with very high variety of foods and combinations of foods. Distinguishing food ingredients from a photograph alone is not at an acceptable level of accuracy to be used as an alternative for assessing dietary intake [Sharp and Allman-Farinelli, 2014].

Current realistic applicability lies in assisting during the individual decision steps of segmentation and recognition during the dietary assessment process. This assistance can be of help for the user to save time or improve accuracy, suggesting e.g. ranked hypothesis for each step that the user can choose from or edit the suggestion. To satisfy the requirements of accuracy of dietary research, participants currently are required to verify and correct the results from the automated processes [Sharp and Allman-Farinelli, 2014].

Still, the state of the art research in food recognition shows improvements in the recognition results through the application of DCNNs. [Kawano and Yanai, 2014, Myers et al., 2015, Wu et al., 2016] show experiments with the use of DCNNs (results listed in Table 4.8). The accuracy for 100 generic food classes lies roughly between 70 and 80%. Results reached with feature extraction combined with encoding techniques (BoF and FV) lie in an area of 40% to 65% [Kawano and Yanai, 2015b, Kawano and Yanai, 2014] in a 100-class problem.

In this thesis the most promising and most discussed methods currently used, are explored and compared on the same datasets:

- Bag-of-Features (BoF)-encoding
- Fisher-Vector (FV)-encoding
- Deep Convolutional Neural Network (DCNN)

Detailed results of the individual approaches of the state-of-the-art projects are listed in the next chapter. The chapter is structured by the methods that are used for the recognition.

CHAPTER 4

State-of-the-art recognition methods

In this chapter the state-of-the-art methods for object recognition and image classification used in the projects from the previous chapter are introduced. Instead of only picking the best performing method, all methods that are commonly used in current food recognition applications are explored. At the end of the chapter a short analysis of the methods is discussed.

Hand-crafted visual descriptors and their encoding methods are discussed in detail, as well as food recognition using DCNNs. First local and global colour descriptors are presented, then local and global texture descriptors are categorised into wavelet based descriptors and the local descriptors LBP, HOG and SIFT. For an overview of how the descriptors and recognition methods perform with food related data, results are presented for each category of descriptors/method from the experiments of a total of eleven state-of-the-art works on food recognition. For each result a short description with detailed information of the used recognition approach, encoding method and also information about the data that was used for training and testing is provided.

4.1 Image Feature Descriptors

4.1.1 Colour Descriptors

Colour Histograms

Colour histograms are commonly used to describe colour distribution of an image [Anthimopoulos et al., 2014]. They can be computed on different colour spaces, with the

result of covering different invariants. Colour histograms are easy to compute and have been successfully used in various object recognition tasks.

Colour histograms are computed separately on each channel of the chosen colour space. For multichannel colour spaces, the histogram values of each channel are concatenated to one vector.

Table 4.1 shows a selection of the histograms used in the state-of-the-art research on food recognition, and the computation of the colour values from the RGB space.

Name	Colour channels	Colour space computation from RGB
$Hist_{RGB}$	RGB	-
$Hist_{OP}$	Opponent colour channels	$OP = \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{2}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}$
$Hist_{RG_{norm}}$	normed R and G channels	$RG_{norm} = \begin{pmatrix} R_{norm} \\ G_{norm} \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \end{pmatrix}$
$Hist_{Hue}$	hue from HSV	$Hue = atan2(\sqrt{3} * (G-B), 2 * R - G - B)$
$Hist_{RGB_{trans}}$	transformed RGB colour space	$RGB_{trans} = \begin{pmatrix} R_{trans} \\ G_{trans} \\ B_{trans} \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix}$

Table 4.1: Histogram types and the computation of the colour space from RGB-space [Anthimopoulos et al., 2014]

Colour histograms are used in [Chen et al., 2012] on the RGB colour space reaching 40% recognition rate in a 50-class problem, [Hoashi et al., 2010] 27% in 85 classes and in [Kawano and Yanai, 2015b] they reach 28% in 100 classes. [Anthimopoulos et al., 2014] experimented with histograms constructed from all the colour spaces listed in Table 4.1 with the conclusion that for their ∼4800 food images, the histogram descriptor in the opponent colour space achieved the best results, reaching 52% in eleven classes. [Kawano and Yanai, 2015b] report to have compared colour histograms in RGB, HSV and La*b* colour spaces and without showing specific results, they report RGB to have performed best out of the three variants.

Colour Moments

[Anthimopoulos et al., 2014] conduct experiments with the colour descriptors *Generalised Colour Moments* (GCM) and *Colour Moment Invariants* (CMI). The idea behind colour

moments is that any probability distribution can be uniquely characterised by its moments [Anthimopoulos et al., 2014]. Colour moments are a generalization of the traditional moments, in one sum they combine powers of the pixel coordinates and the intensities of each colour channel [Mindru et al., 2004].

The generalised colour moment M_{pq}^{abc} of order $p+q$ and degree $a+b+c$ is defined as

$$M_{pq}^{abc} = \sum_{x=1}^{W} \sum_{y=1}^{H} x^p y^q R(x,y)^a G(x,y)^b B(x,y)^c$$

where x and y are pixel positions and $R(x,y)$, $G(x,y)$ and $B(x,y)$ are the RGB colour channel values. W and H are the image width and height.

[Anthimopoulos et al., 2014] use orders of 0 or 1 and degrees 1 or 2 in their study to compute the Generalised Colour Moment (GCM), leading to 27 possible combinations, and then compute 24 invariant functions from these moments following [Mindru et al., 2004]. The results (Table 4.2) show a good discrimination for food images compared to the other descriptors used in the paper. The best result achieved around 59% on a dataset with 4868 images with the Colour Moment Invariant (CMI) descriptor, other results on the same data were 52% with colour histograms on opponent colour space, around 61% with standard SIFT and 77.6% with SIFT on HSV colour space.

[Mindru et al., 2004] compute 30 combinations of the GCMs, with up to the first order and the second degree (including the degree of zero for all colour channels). The limitation to lower order moments are argued with higher robustness to noise compared to when using higher moments. This leads to a selection of GCMs of $M_{00}^{abc}, M_{10}^{abc}$ and M_{01}^{abc}, with $(a,b,c) \in \{(0,0,0),(1,0,0),(0,1,0),(0,0,1),(2,0,0),(0,2,0),(0,0,2),(1,1,0),(1,0,1),(0,1,1)\}$. This set of GCMs build the basis for the CMI descriptor, which are functions of rational expressions of combinations of the colour moments. The exact functions used for the experiments in the implementation are listed in Section 5.3.1. The invariant functions are constructed so that they do not change under the selected geometric (viewpoint) and photometric (illumination) transformations. [Mindru et al., 2004] show how to make the colour moment descriptor invariant to viewpoint and illumination, based on theoretical models of photometric transformations. Three models are discussed: scaling, scaling with offset and affine transformations. Obtained are the invariants by a method the authors call *Lie group* approach, where the invariants are solutions of systems of partial differential equations.

Colour Patch Descriptors

[Kawano and Yanai, 2014] report of a local colour descriptor which the authors call *colour patches*. The neighbourhood is divided into 2×2 blocks, and from each block the mean and variance is computed for each colour channel of the RGB colour space. That results in a description of each patch of a 24-dimensional vector. The neighbourhood size used

in the experiments is not reported. The FV-encoding of the descriptor performs with an accuracy of 53% for 100 generic food classes and 41.6% for 256 classes (see Table 4.2). The results are slightly higher than the results of the HOG descriptor (Table 4.4), 50.1% and 36.4% respectively, showing the descriptive power of colour in food recognition. [Bosch et al., 2011b] report of a similar construction of a colour descriptor. The mean and variance of ten colour channels (R,G, B, C_b, C_r, a, b, H, S, V) are computed from local keypoints, resulting in a dimensionality of 20 values per descriptor. They also compute the descriptor globally in the domain of the segmented food item areas. The results of the two methods are 79.2% for the local patches and 78.6% for the global segment-wide computed descriptor. The dataset size was 179 images in 39 categories. The close results are presumably due to the small segments which are local patches of its own, and the resulting similarity of local and global values.

Paper	Details	Dataset size	Classes	Accuracy
[Chen et al., 2012]	$Hist_{RGB}$ with 96 bins on a 4×4 grid of the image.	5000	50	40.3%
[Anthimopoulos et al., 2014]	BoF (10k dict. size) enc. of $Hist_{RGB}$	4868	11	$\sim$ 37%
	BoF (10k dict. size) enc. of $Hist_{OP}$ (opponent colour space)	4868	11	$\sim$ 52%
	BoF (10k dict. size) enc. of $Hist_{RG_{norm}}$	4868	11	$\sim$ 47%
	BoF (10k dict. size) enc. of $Hist_{HUE}$	4868	11	$\sim$ 39%
	BoF (10k dict. size) enc. of $Hist_{RGB_{trans}}$	4868	11	$\sim$ 23%
	BoF (10k dict. size) enc. of CMI	4868	11	$\sim$ 59%
[Bosch et al., 2011b]	BoF of μ and σ of 10 local colour components	179	39	79.2%
	global colour, μ and σ of 10 colour components of whole segment	179	39	78.6%
	global entropy, μ and σ of the R, G, B channels are estimated (in blocks and then avg)	179	39	78.2%
[Zhu et al., 2015]	μ and σ of 10 colour components of whole segments (global)	30[a]	79	68.0%
	Entropy colour statistics (μ and σ of RGB components for whole segment, global)	30[a]	79	35.0%
	Predominant colour statistics (distribution of salient colours) within segment (global)	30[a]	79	60.0%
[Hoashi et al., 2010]	Hist_{RGB} with 64 bins of 4 sub-images from a 2×2 grid division	85×100	85	27.08%
[Kawano and Yanai, 2014]	FV encoding of μ and σ^2 of RGB from patches	100×100	100	53.04%
	FV encoding of μ and σ^2 of RGB from patches	256×100	256	41.60%
[Kawano and Yanai, 2015b]	576-dim. $Hist_{RGB}$ with 64 bins (from 3×3 divided image blocks), χ^2 kernel feature mapping triplicating the descr. to 1728-dim.	12905	100	$\sim$ 28%
	BoF-enc. of colour patches / colour histogram	12905	100	$\sim$ 29%
	FV-enc. of colour patches	12905	100	$\sim$ 41%
	FV-enc. of colour patches (flipped training images)	12905	100	$\sim$ 42%

[a] patches of segmentation samples, average of 30 per class.

Table 4.2: Summary of the results of experiments with colour descriptors from researched papers.

4.1.2 Texture descriptors

Wavelet filters:

Gabor filter responses A two-dimensional Gabor filter is used to pass spatial frequencies in a fixed direction. Using Gabor filter responses as a descriptor, is able to capture the properties of spatial localization, orientation information, and spatial frequency information and is widely used in texure representation and image recognition [Chen et al., 2012, Li et al., 2010].

For the descriptor, a set of Gabor filters, also called *Gabor filter bank*, of different scales and orientations are convolved with the image (Figure 4.1 shows the impulses of a bank of Gabor filters). To encode the responses of each filter into the descriptor, [Chen et al., 2012] e.g., compute mean and variance of image blocks of the magnitudes of the filter responses and concatenate the values to form the texture descriptor. To make the descriptor invariant to scale and orientation, [Rahman et al., 2011] propose to shift around the responses of each scale and orientation filter impulse circularly, so that the strongest impulse is on the first position. This method produces similar descriptors for similar textures with different scales and rotations.

The results of experiments with Gabor filters on food images are summarised in Table 4.3. [Chen et al., 2012] reach 26.6% with the Gabor filter descriptor, compared to a result of from 39.9 to 45.9% with sparse coding of LBP descriptor (Tables 4.3 and 4.4).

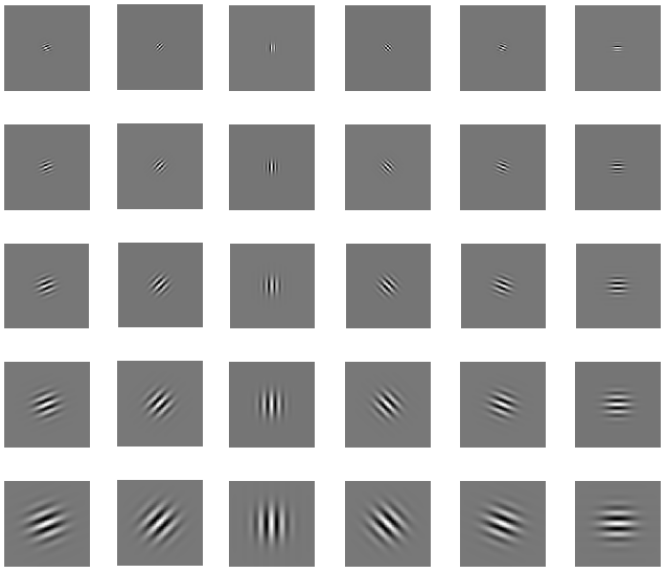


Figure 4.1: Impulses of a Gabor filter bank with five scales and six orientations [Rahman et al., 2011].

Paper	Details	Dataset size	Classes	Accuracy
[Chen et al., 2012]	the μ and σ^2 of magnitudes of Gabor filter (6×5)	5000	50	26.6%
[Bosch et al., 2011b]	BoF of μ and σ^2 of Gabor filter (6×4) response, on local interest points	179	39	29.1%
	μ and σ^2 of Gabor filter (6×4) response, on blocks of whole segment (global), then averaged	179	39	40.2%
	BoF of Haar wavelet responses in horizontal and vertical directions, on local interest points	179	39	64.1%
[Hoashi et al., 2010]	avg. responses of Gabor filter (6 dir. × 4 sc.) of 16 sub-images from a 4×4 grid division (384 dims), no encoding	85×100	85	25.35%
	avg. responses of Gabor filter (6 dir. × 4 sc.) of 9 sub-images from a 3×3 grid division (216 dims), no encoding	85×100	85	23.60%
[Zhu et al., 2015]	EFD (global, whole segment)	30[a]	79	47.0%
	GFD (global, whole segment)	30[a]	79	27.0%
	GOSDM (global, whole segment)	30[a]	79	32.0%

[a] patches of segmentation samples, average of 30 per class.

Table 4.3: Summary of the results of experiments with wavelet filter descriptors and global texture descriptors from researched papers.

Local texture descriptors:

LBP-descriptor The LBP descriptor [Ojala et al., 2002] was introduced for discrimination on basis of texture information. It is invariant to gray-scale, making it robust against illumination changes that occur within a class, and it is invariant to rotation, making it robust to variance in textural orientation. It has a low computational complexity, resulting in simple and fast computation.

The LBP descriptor is a local descriptor, describing the pattern of differences in intensity of the central pixel in relation to its neighbouring (local) pixels. The neighbourhood is defined as the pixels of a fixed distance located angularly around the center pixel, forming a circular symmetric neighbour set. The LBP descriptor has two parameters: P and R, where P determines the quantization of the angular space (number of neighbours), and R determines the spatial resolution (radius), illustrated in Figure 4.2. The intensity

value of neighbouring pixels that do not fall into a center of a pixel get estimated by interpolation.

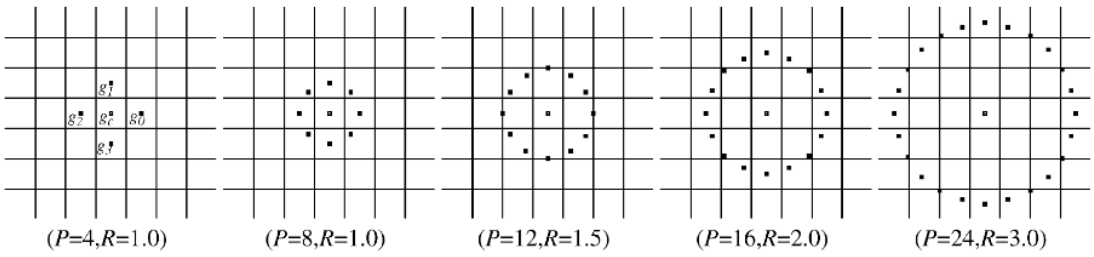


Figure 4.2: Examples of sampling parameters R and P, of the LBP descriptor [Ojala et al., 2002].

To achieve invariance with respect to shifts in gray scale, first the intensity value of the central pixel is substracted from all sampled points. Then all P sampled neighbours get compared to the threshold value of the central pixel, getting assigned 0 if the value is less than zero, and 1 if the value is greater or equal than zero. The outputs are concatenated, producing a binary pattern.

To achieve rotation invariance the resulting 2^P possibilities of binary patterns get reduced to the minimum of different patterns possible when shift rotating the pattern around the central pixel. The formed pattern can be seen as feature detectors, forming edges, lines, spots or flat areas. E.g. the pattern with all zeros would detect bright spots and for a LBP descriptor with $P = 8$, the pattern with four consecutive ones in it would detect edges. Those cases are illustrated in the first and the last patterns in Figure 4.3. There are a total of 36 rotation invariant unique patterns for the LBP with eight neighbours.

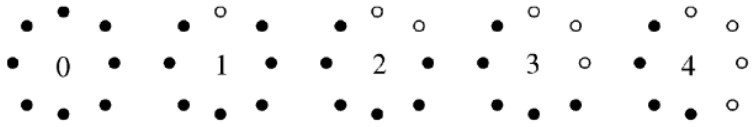


Figure 4.3: Five examples of unique binary patterns of the LBP descriptor [Ojala et al., 2002].

Further improvements of the descriptor were reached by reducing to patterns that [Ojala et al., 2002] define as *uniform* patterns, having the maximum of two transitions between *0* and *1* or vice versa inside the pattern. In the case of the 8-neighbour LBP descriptor there are eight such patterns, describing detectors for spots and lines. Exactly $P + 1$ *uniform* patterns occur in a circularly symmetric set of P neighbours.

[Chen et al., 2012] achieve a recognition rate of 39.9 to 45.9% with sparse coding in 50 classes. Results of the LBP descriptor and other local texture descriptors in food recognition experiments from the state of the art projects are summarised in Table 4.4.

HOG-descriptor [Dalal and Triggs, 2005] propose a descriptor that computes histograms of the orientations of the gradients from local points on a grid. The gradients are computed with the 1-dimensional derivative mask $[-1, 0, 1]$, there is no smoothing of the image performed for preprocessing.

The histogram is formed of nine bins evenly spaced over $0° - 180°$. The neighbourhood of the descriptor spans 16×16 pixels, consisting of four *cells* with each 8×8 pixels of size. The stride of dense sampling is 8 pixels, covering each *cell* four times. The descriptor then gets l^2-normalised to unit length.

The descriptor is invariant to intensity changes but not to rotation and scale. Results of the HOG descriptor in food recognition experiments from state of the art projects are summarised in Table 4.4.

In [Kawano and Yanai, 2014] the RootHOG[1] descriptor achieves a recognition rate of 50% in 100 classes and 36% in 256 classes with FV encoding.

SIFT-descriptor The SIFT descriptor was proposed in [Lowe, 2004]. The total neighbourhood considered for the computation is a window of 16×16 pixels. In the original descriptor the computation is based on the intensity values of the image, omitting all colour information. The gradients of each pixel is computed with

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2},$$

where L holds the image intensities smoothed with a Gaussian. The magnitudes of the gradients of the sample points are weighed by a 2-dimensional Gaussian function, giving more focus to the central sample points. 8-bin histograms are computed of 16 subregions of 4×4 pixels, resulting in 16 histograms of eight gradient directions per descriptor. The total dimensionality of the SIFT descriptor therefore is 128. The descriptor is invariant to light intensity changes, as relative information in gradient shifts of intensity is encoded with the histograms. The original SIFT descriptor is operating on intensity values enabling a descriptiveness for texture information, but it not being capable of capturing colour information. A colour variant of the SIFT descriptor computes the gradients on each colour channel of the colour space representation of the image, and combines the resulting histograms of the channels.

In [Zhu et al., 2015] the SIFT descriptor was computed on the individual colour channels of the RGB colour space but the descriptors were not combined, which did not lead to a significant change in accuracy (Table 4.5), compared to the SIFT descriptor obtained from intensity information. Whereas a combination of the descriptors from all colour channels, the approach of [Anthimopoulos et al., 2014], add full colour information and improve the accuracy for the combined descriptors, computed for six different colour spaces (Table 4.5). The best result of the variants in [Anthimopoulos et al., 2014], was obtained from SIFT from HSV space, which increased the accuracy about 16% compared

[1]RootHOG is inspired by *RootSIFT* from [Arandjelović and Zisserman, 2012], an element-wise square root of the L1 normalised HOG descriptor [Kawano and Yanai, 2014].

Paper	Details	Dataset size	Classes	Accuracy
[Chen et al., 2012]	sparse coding of LBP: "59-bins histograms from 16×16 patches with step size of 8px in each level of [a 3 level] image pyramid (2048dimensions)"	5000	50	45.9%
	sparse coding of LBP (histogram pooling, 1024-Dim)	5000	50	39.9%
	LBP as in [Ojala et al., 2002], without encoding	5000	50	36.2%
[Bosch et al., 2011b]	BoF of DAISY	179	39	60.3%
[Zhu et al., 2015]	BoF of SURF	30[a]	79	45.0%
[Kawano and Yanai, 2014]	FV-encoding of RootHOG	100×100	100	50.14%
	FV-encoding of RootHOG	256×100	256	36.46%
[Bossard et al., 2014]	Random Forest discriminative components mining with BoF-enc.(1024 dict. size) of SURF	101000	101	33.47%
	Random Forest discriminative components mining with FV-enc. of SURF, with 64 clusters	101000	101	44.79%
[Kawano and Yanai, 2015b]	BoF-enc. of SURF (dense 8px grid), with dict. size of 500, χ^2 kernel feature mapping triplicating the descr. to 1500-dim., linear SVM	12905	100	$\sim$ 29%
	FV-enc. of HOG-patches (flipped training images) (dense 6px grid), linear SVM	12905	100	$\sim$ 37%

[a] patches of segmentation samples, average of 30 per class.

Table 4.4: Summary of the results of experiments with various local texture descriptors from researched papers.

to the original SIFT descriptor computed on the intensity values. The concatenation of the $SIFT_{HSV}$ with the CMI descriptor, showed no significant improvement compared to the $SIFT_{HSV}$ descriptor on its own, indicating that the colour variant is capable of describing colour information in addition to its textual description capability.

Sampling strategies of local descriptors To compute local descriptors, there are several common strategies for selecting the keypoint locations. The locations can be selected randomly, which potentially leads to a higher correlation if the locations happen

Paper	Details	Dataset size	Classes	Accuracy
[Chen et al., 2012]	BoF of SIFT (1024 dictionary size)	5000	50	40.2%
	sparse coding(multi-scale max pooling, 1024 dictionary size) of SIFT	5000	50	43.4%
	sparse coding(histogram pooling, 1024 dictionary size) of SIFT	5000	50	53.0%
[Anthimopoulos et al., 2014]	BoF (10k dict. size) enc. of SIFT	4868	11	$\sim 61\%$
	BoF (10k dict. size) enc. of SIFT_{RGB}	4868	11	$\sim 66\%$
	BoF (10k dict. size) enc. of SIFT_{HSV}	4868	11	77.6%
	BoF (10k dict. size) enc. of SIFT_{HUE}	4868	11	$\sim 67\%$
	BoF (10k dict. size) enc. of $\text{SIFT}_{OPPONENT}$	4868	11	$\sim 70\%$
	BoF (10k dict. size) enc. of SIFT_{C}	4868	11	$\sim 70\%$
	BoF (10k dict. size) enc. of SIFT_{RG}	4868	11	$\sim 71\%$
[Hoashi et al., 2010]	BoF (2k dict. size) enc. of SIFT with Difference of Gaussians (DoG) sampling	85×100	85	33.42%
	BoF (2k dict. size) enc. of SIFT with dense sampling	85×100	85	32.21%
[Bosch et al., 2011b]	BoF of SIFT	179	39	65.2%
[Zhu et al., 2015]	BoF of SIFT	30[a]	79	48.0%
	BoF of SIFT_{RED}	30[a]	79	48.0%
	BoF of SIFT_{GREEN}	30[a]	79	49.0%
	BoF of SIFT_{BLUE}	30[a]	79	47.0%
	FV of SIFT	30[a]	79	61.0%

[a] patches of segmentation samples, average of 30 per class.

Table 4.5: Summary of the results of experiments with the SIFT descriptor and variants thereof from researched papers.

to be dense. Another selection strategy is over a uniform grid of the image (also called dense-sampling). A third common strategy is the use of an Interest Point Operator. The DoG keypoint detector, a method developed by [Lowe, 2004] for the selection of keypoints of the SIFT descriptor, which detects local extrema in the DoG space. The idea behind sampling interest points is to find points that are more descriptive than random points,

using algorithms that filter points with a higher descriptivness, such as points on edges or with certain minimum criteria of contrast etc. Other popular interest point detectors are the Harris-Affine detector [Mikolajczyk and Schmid, 2004] and the Maximally Stable Extremal Regions (MSER) [Matas et al., 2002] keypoint detector [O'Hara and Draper, 2011]. [Anthimopoulos et al., 2013] show that grid sampling performs better than DoG- and random-sampling (illustrated in Figure 4.4), in their food recognition experiments. In [Hoashi et al., 2010, p. 300] DoG-sampling (33.42%) performs slightly better than random-sampling (30.36%) and grid-sampling (32.21%)[2], using a 2000 visual words strong dictionary.

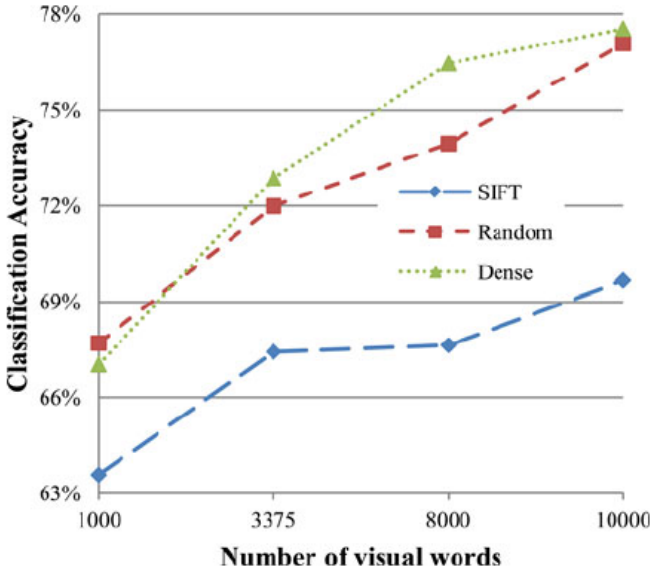


Figure 4.4: Comparisons of sampling strategies for the SIFT descriptor [Anthimopoulos et al., 2013]. The strategy denoted here as *SIFT* is the DoG keypoint detector used in the SIFT descriptor.

4.1.3 Combination of descriptors

For the combination of multiple single descriptor vectors, also called *fusion*, three strategies can be followed [Mangai et al., 2010]. Categorised by the level on which the fusion takes place, the strategies are, **information/data fusion** (low-level fusion, early fusion) that is performed on the raw data of the descriptor, creating new raw data, expected to be more descriptive. **Feature fusion** (intermediate-level fusion) combines descriptors following a selection process to remove redundancy in the combined feature space (e.g.

[2]The DoG keypoint detector selects a limited number of keypoints. In the evaluation they were compared with a higher number of samples for the random and grid strategies.

if two features have a similar distribution), or remove descriptors that turn out not to be descriptive for the task. The third strategy **decision fusion** (high-level fusion, late fusion), is performed after classification at decision level. A set of classifiers are used and the resulting decision is a combination of the single classifiers. One way of realizing the combination, is to build a feature vector from the results of the individual classifiers of each feature on which a new classifier is trained. An example for this combination strategy are the experiments of [Beijbom et al., 2015], where several single-descriptor classifications were combined with a SVM[3]. The accuracy of five individual descriptors was between 43.6% and 57.7%, and was increased to 77.4% due to the fusion. An other strategy is to implement a voting scheme, e.g. majority voting, where the decision falls to the category that is identified by the majority of the individual classifiers. Different types of classifiers can be combined and also different sets of descriptors for each of the classifiers.

Table 4.6 shows results from works on food related recognition, that experiment with fusion of two or more descriptors. [Zhu et al., 2015] observe that *decision fusion* with majority voting of 12 single descriptors (accuracy of 75%) outperforms a concatenation of a selection of the three best-performing descriptors into one vector (*feature fusion*) and applying classification (accuracy of 52%).

[Kawano and Yanai, 2014,Kawano and Yanai, 2015b] perform combinations of descriptors on a low-level. The results of the *feature fusions* are also summarised in Table 4.6. In [Kawano and Yanai, 2014], results for single classification of FV-encoding of a RootHOG descriptor on the 100 class dataset was 50.14% and of the colour patch descriptor 53.04%. The combination of the two reached 65.32%. In [Kawano and Yanai, 2015b] FV encoding of HOG-patches reached 37%, of colour patches 41% and the combination 59.6%.

[Anthimopoulos et al., 2014] compare concatenations of raw descriptors to concatenation of the histograms resulting from BoF-encoding. The results are illustrated in Figure 4.5. The best combination from the experiment with eleven classes was achieved by combining the descriptors SIFT_{HSV} with the CMI descriptor (achieved around 59% on its own) by raw concatenation before classification, resulting in 77.8% accuracy. Though this did not improve the result of the single descriptor results, the SIFT descriptor on its own achieved 77.6%. This result indicates the capability of the SIFT colour variant for describing colour information.

4.2 Encoding techniques

Local descriptors extracted with a sampling strategy from an image, results in a large collection of information. E.g. on an image with 500×400 pixel and a dense-sampled 128

[3] A one-versus-rest linear SVM was trained for each of the 41 classes and each of the five descriptors, leading to a 205-dimensional joint feature vector, that was then trained with an additional one-versus-rest linear SVM.

Paper	Details	Dataset size	Classes	Accuracy
[Anthimopoulos et al., 2014]	BoF (10k dict. size) enc. of SIFT$_{HSV}$ + CMI	4868	11	77.8%
[Kawano and Yanai, 2014]	FV of colourpatches and RootHOG-patches	12905	100	65.32%
	FV of colourpatches and RootHOG-patches	100×256	256	52.85%
[Kawano and Yanai, 2015b]	BoF (dict. size of 500) of SURF, kernel feature mapping to 1500 dims + RGB colour histogram, linear SVM	12905	100	42.0%
	FV-enc. of PCA reduced 24 dim. HOG patches + FV-enc. of colour patches with 32 gaussians, linear SVM	12905	100	49.7%
	FV-enc. of PCA reduced 24 dim. HOG patches + FV-enc. of colour patches(flipped training images) with 32 gaussians, linear SVM	12905	100	51.9%
	FV-enc. of 32 dim. HOG patches + FV-enc. of colour patches(flipped training images) with 64 gaussians, linear SVM	12905	100	59.6%
[Chen et al., 2012]	Data fusion of SIFT, LBP, colour and gabor descriptors	5000	50	62.7%
	Multi-class Adaboost [Zou et al., 2009] of SIFT, LBP, colour and Gabor descriptors	5000	50	68.3%
[Beijbom et al., 2015]	LLC-encoding, with 1024 words learned via k-means of SIFT, LBP, colour, HOG and MR8 [Varma and Zisserman, 2005] with max-pooling and late fusion with linear SVM	5000	50	77.4%
		646	41	51.2%

Table 4.6: Summary of the results of experiments of combinations of descriptors by feature fusion from researched papers. More combinations of the results from [Anthimopoulos et al., 2014] are illustrated in Figure 4.5.

dimensional SIFT descriptor on an 8 pixel grid, 3100 descriptors are extracted, holding a size of 1550 KiB per image, for a database with 100000 images, that accounts to around 148 GiB of raw descriptor data. Encoding strategies such as BoF and FV help to reduce the size of the information, with the aim of keeping a high descriptiveness of the original information.

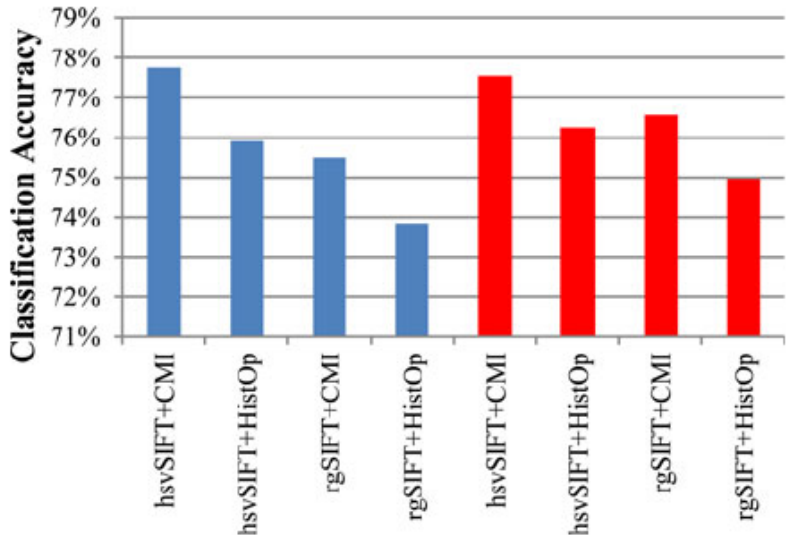


Figure 4.5: Comparisons of raw descriptor concatenation in red. In blue are the results from histogram concatenation (from the BoF-encoding) [Anthimopoulos et al., 2014].

The focus on the encoding methods BoF and FV, is because they are the most used and the most successful in food recognition related works (Chapter 3). Other encoding methods applied in food recognition are sparse coding [Chen et al., 2012] and LLC-encoding (used in [Beijbom et al., 2015]).

In [Chatfield et al., 2014] the choice of the improved[4] FV technique, is argued as *usually* outperforming BoF, LLC and Vector of Locally Aggregated Descriptors (VLAD) [Jegou et al., 2012].

4.2.1 Bag-of-Features (BoF) encoding

The BoF method, also called *Bag-of-Words* or *Bag-of-visual-Words*, originates from the representation of words in textual information retrieval [O'Hara and Draper, 2011]. With Bag-of-Words a text document is represented by a normalised histogram by counting the words. The words appearing in the document form the *dictionary*. All words can be used, or non-informative words such as articles may be excluded, and synonyms may be represented by the same term. The vector that represents the document has the size of the dictionary, forming a sparse histogram vector. Each element of the vector is associated with a word in the dictionary, and the value of that element is the number of times the word appears in the document, normalised by the number of words sampled [O'Hara and Draper, 2011]. It is called a *bag* because the spatial order of the words in the document is

[4]Refers to the description in [Perronnin et al., 2010].

lost. The BoF image representation is analogous to the textual concept. The *dictionary* is constructed by quantization (clustering) of a set of image feature descriptors into a fixed sized set of *visual words*. The impact of the size of the dictionary, i.e. the number of *visual words*, on the classification accuracy is illustrated in Figure 4.6. The images are represented by histograms of the visual dictionary that incorporates assignments of each descriptor to a visual word. As mentioned, the approach is characterised by an orderless collection of image features [O'Hara and Draper, 2011]. In the step of the histogram computation, the spatial information of the location of the keypoint of the descriptor gets discarded. Therefore the information of the absolute location of a keypoint and also the relative locations between the keypoints is lost, and also the scales and orientations of the features are not incorporated in the histogram encoding.

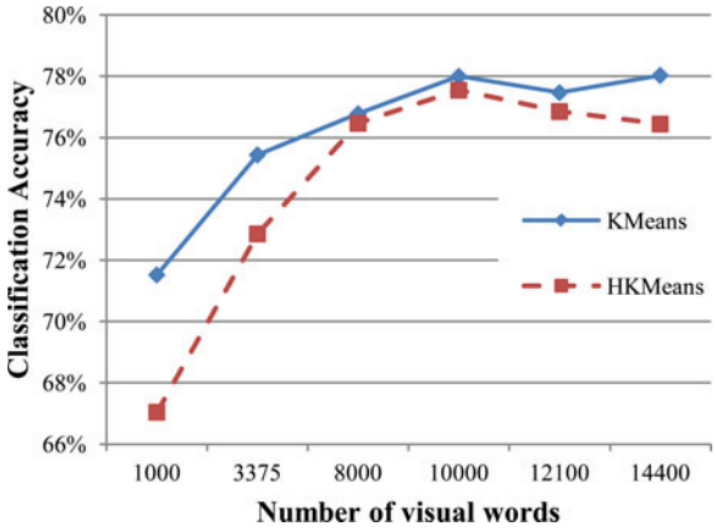


Figure 4.6: Influence of dictionary size on accuracy in experiments with food images in [Anthimopoulos et al., 2014].

There are two advantages that result from encoding to histograms. First the size of the image description is constant, independent from the number of extracted descriptors. This is necessary when using a classifier that only allows input of vectors of equal size. Second, in case of higher dimensional descriptors and/or high extraction sampling densities, the information size gets reduced to the size of the dictionary.

BoF technique consists of five primary steps: [O'Hara and Draper, 2011, Anthimopoulos et al., 2014]:

- Keypoint Sampling: strategies are discussed in Section 4.1.2 on page 48.
- Local Feature Description: extraction of the features of each image.

- Constructing the dictionary (feature space quantization): the extracted features from the training set (or a subset thereof) are clustered into the *dictionary*[5]. Each cluster represents *one visual word* or *term*.
- Term assignment (descriptor quantization): the extracted features of an image get assigned to the entry in the dictionary with the highest similarity to those features.
- Generate histogram: the assignments to each cluster are counted and the histogram gets normalised. This forms the final representation of the BoF-encoding of the image.

The dictionary is constructed only once, all other steps are executed in both training and testing for each image. Figure 4.7 illustrates the steps.

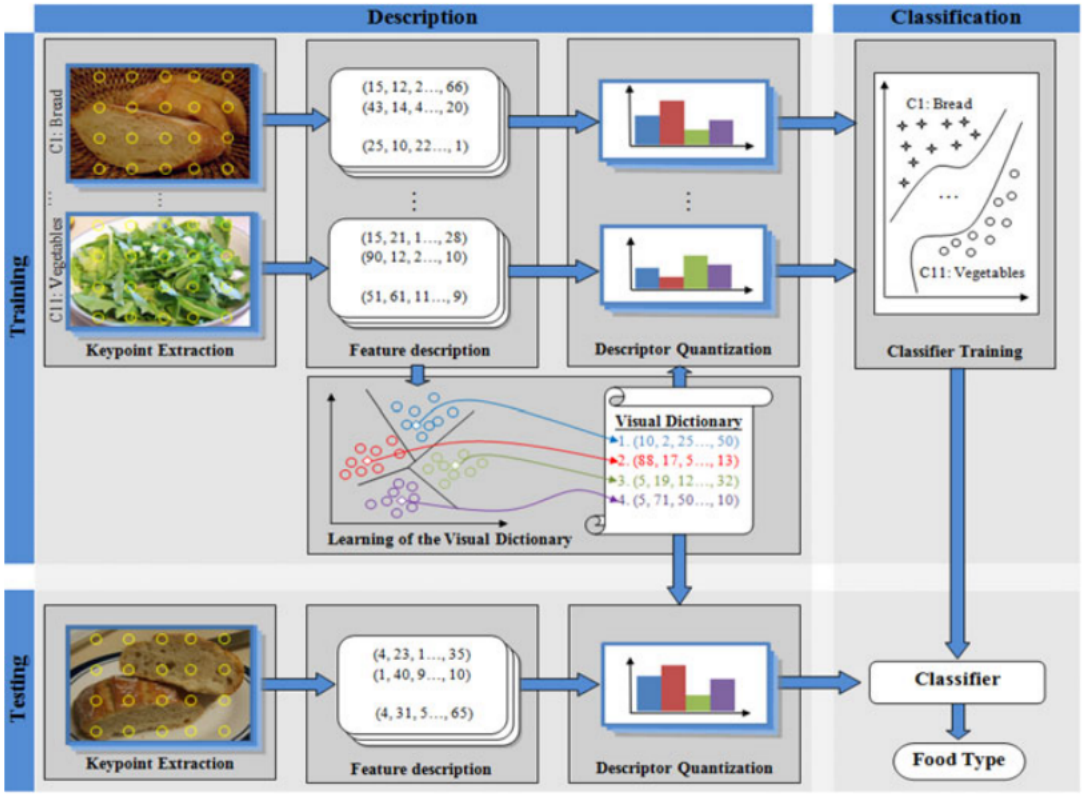


Figure 4.7: Illustration of the steps involved in the BoF-encoding technique [Anthimopoulos et al., 2014, p. 1263].

[5] Also called *visual vocabulary* or *codebook*.

Vector Space and Descriptor Quantization

To be able to construct a dictionary, the descriptor space is quantised into clusters. Many BoF implementations use K-means[6] or a hierarchical version of the algorithm for clustering [Anthimopoulos et al., 2014,Kawano and Yanai, 2015b,Zhu et al., 2015,Lazebnik et al., 2006].

In the next step a strategy is needed to assign a cluster (visual word) to represent actual descriptors. In a *hard assignment* the closest neighbouring cluster-center is assigned (Nearest Neighbour strategy), regardless of the distance to other cluster-centers. Ambiguous descriptors that lie near Voronoi boundaries in between clusters, lead to weak representations by the dictionary [O'Hara and Draper, 2011]. *Soft assignment* approaches compensate with weighing the nearest k-neighbours, so that the nearest cluster gets a higher weight but the k-1 next closest clusters, are also considered. [O'Hara and Draper, 2011] report of works using soft assignments to cause a modest increase in accuracy, but associated with higher computational search time. The reported implementation with k=3, required seven times the number of multiplications compared to the simple assignment.

Spatial Pyramid Matching

In order to overcome the orderless collection of feature descriptors of BoF, [Lazebnik et al., 2006] introduce the Spatial Pyramid Matching (SPM) technique. The technique is inspired by the Pyramid Match Kernel in [Grauman and Darrell, 2005], which is an orderless image representation of **multi-resolution** histograms[7], that allows matching of two collections of feature descriptors in high-dimensional space [Lazebnik et al., 2006]. In contrast, SPM operates in original image space, with **constant resolution** for all levels of the Spatial Pyramid (SP). Histograms from gridded regions[8] of the image are constructed, dividing the image at a different grid on each level of the pyramid. For each region, the descriptors located within that region are encoded into one histogram. The SPM therefore generates r feature vectors (histograms), where r is the number of total regions from all levels of the SP. The histograms are normalised by the total number of feature descriptors contributing to the histogram. All histograms (one for each region) are concatenated to a single feature vector. The result is a much higher dimensional representation, r-times the size of the standard BoF approach. E.g. in [Kawano and Yanai, 2014] the SPM is arranged on three levels with 1×1, 2×2 and 1×3 regions, resulting in eight total regions. When reduced to a single pyramid level consisting of the enitre image, the technique (and feature vector) is equal to standard BoF. The works of [Grauman

[6] standard implementation described in [Lloyd, 1982]

[7] The histograms are constructed from an image pyramid, i.e. the whole image on different image resolutions.

[8] *Region* denotes the areas resulting by splitting the image into a defined grid, e.g. a 2×2 grid splits the image into four regions.

and Darrell, 2005] and [Lazebnik et al., 2006] both develop a matching kernel for their histogram constructions. The SPM suffices in a significant improvement in accuracy over the standard BoF approach [Lazebnik et al., 2006].

Kernel Feature Mapping

[Kawano and Yanai, 2015b] focus on computational cost of the whole recognition process, realizing a system running all its computations on a mobile device. For classification linear SVMs are used. Linear SVMs are capable of being trained linear in time, $O(n)$, with n training samples [Joachims, 2006], whereas the learning of non-linear SVMs scales somewhere between $O(n^2)$ and $O(n^3)$ [Perronnin et al., 2010]. However there have been many studies showing that linear SVMs perform inferior to non-linear SVMs on BoF histograms.

A linear SVM is defined by the inner product $F(x) = \langle w,\ x \rangle$ between a data vector $x \in \mathbb{R}^D$ and a vector of weights $w \in \mathbb{R}^D$ [Vedaldi and Zisserman, 2010]. A non-linear SVM is defined by the expansion $F(x) = \sum_{i=1}^{N} \beta_i K(x,\ x_i)$, where K is a non-linear kernel. $x_1,\ \ldots\ , x_N$ are the support vectors, N feature vectors that represent the training set. Assuming that the computational cost of the inner product and the kernel are comparable, then the evaluation of a non-linear SVM is N-times slower, similarly for the training [Vedaldi and Zisserman, 2010].

Non-linear SVMs can be seen as linear SVMs operating in a higher feature space. The kernel maps the feature into a higher dimensional space. Kernels commonly used in computer vision, such as χ^2, intersection, Hellinger's and Jensen-Shannon kernel are additive combinations of homogeneous kernels [Vedaldi and Zisserman, 2010]. A kernel $K_h : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}$ is homogeneous if $\forall c \geq 0 : K_h(cx,\ cy) = cK_h(x,\ y)$.

The homogeneous feature mapping [Vedaldi and Zisserman, 2010] approximates the kernel, and can be computed explicitly, which accelerates the learning procedure of the SVM. The kernel mapping function $\Psi(x)$ of the data vector x is constructed such that $K(x,\ y) = \langle \Psi(x),\ \Psi(y) \rangle$.

Following the instructions in [Vedaldi and Zisserman, 2010, Kawano and Yanai, 2015b] implement a mapping function that realizes an approximation of the χ^2 kernel. The dimension of the resulting feature vector is three times larger than the original feature vector. The mapping function that was used is defined as:

$$\Psi(x) = \sqrt{x} \begin{pmatrix} 0.8 \\ 0.6\ cos(0.6\ log(x)) \\ 0.6\ sin(0.6\ log(x)) \end{pmatrix}$$

4.2.2 Fisher Vector (FV) encoding

FV-encoding is a high performance method to represent a set of local features [Kawano and Yanai, 2015b]. It can be seen as an extension of the BoF-encoding. FV is not limited to the occurences of the visual words, but also incorporates the distribution of the descriptors [Perronnin et al., 2010]. Through the use of higher order statistics the quanitization error is reduced, compared to the encoding with BoF [Kawano and Yanai, 2015b]. The FV encoding yields higher results than other encoding methods such as BoF and LLC [Kawano and Yanai, 2015b, Yang et al., 2010].

The FV computation bases on an underlying probability distribution estimate of the feature space. A common method in FV-encoding is the use of GMMs, as any continous distribution can be approximated with arbitrary precision [Titterington et al., 1985, Perronnin et al., 2010].

The FV was first presented with application to image classification in [Perronnin and Dance, 2007], improvements were presented in [Perronnin et al., 2010] and a more detailed work on theory and practice for image classification with the FV was presented in [Sanchez et al., 2013].

Gaussian Mixture Model (GMM)

[Sanchez et al., 2013] define a GMM $u_\lambda(x)$ with K components as:

$$u_\lambda(x) = \sum_{k=1}^{K} w_k u_k(x),$$

and its parameters are denoted by $\lambda = \{w_k,\ \mu_k,\ \Sigma_k,\ k = 1,\ \dots\ ,\ K\}$, where w_k, μ_k and Σ_k are the mixture weights, mean vectors and covariance matrices of Gaussian k respectively. To ensure a valid distribution, $\forall k : w_k \geq 0,\ \sum_{k=1}^{K} w_k = 1$ has to hold.

The k^{th} Gaussian component $u_k(x)$ is computed with:

$$u_k(x) = \frac{1}{(2\pi)^{D/2}\,|\Sigma_k|^2} \exp\left\{-\frac{1}{2}(x-\mu_k)'\ \Sigma_k^{-1}\ (x-\mu_k)\right\}$$

For the covariance, a standard assumption is to use the diagonal covariance matrix. σ_k^2 is used to denote the vector that holds the diagonal entries of the covariance matrix Σ_k [Sanchez et al., 2013].

Fisher Vector

The details of the connection between the Fisher Kernel (FK) and the FV are found in [Perronnin and Dance, 2007, Perronnin et al., 2010, Sanchez et al., 2013]. [Jaakkola and

Haussler, 1998] introduce the FK to measure the similarity between two samples [Sanchez et al., 2013]. [Perronnin et al., 2010] state that learning a kernel classifier (such as SVM) using the FK is equivalent to learning a linear classifier on the FV.

In this section the necessary information to compute the FV is presented.

Let X be the set of all extracted D-dimensional features of an image: $X = \{x_t,\ t = 1,\ \ldots,\ T\}$, where T is the number of features and $x_t \in \mathscr{X}$, $\mathscr{X}$ denoting the feature space [Sanchez et al., 2013]. The probability function that models the generative process of the elements in $\mathscr{X}$, is denoted with $u_\lambda(x)$.

The information about the distribution of the descriptors is incorporated into the FV through the gradient of the log-likelihood, given by:

$$G_\lambda^X = \nabla_\lambda\ log\ u_\lambda(X),$$

where $G_\lambda^X \in \mathbb{R}^M$, is depending only on the number of parameters M in λ, and independent of the sample size T. The gradients of each parameter in λ describe the contribution of the parameter to the generative process.

The gradients of a single descriptor x_t, for the three parameters in λ: α_k[9], μ_k and σ_k, are given by:

$$\begin{aligned}
\nabla_{\alpha_k} log\ u_\lambda(x_t) &= \gamma_t(k) - w_k \\
\nabla_{\mu_k} log\ u_\lambda(x_t) &= \gamma_t(k) \left(\frac{x_t - \mu_k}{\sigma_k^2} \right) \\
\nabla_{\sigma_k} log\ u_\lambda(x_t) &= \gamma_t(k) \left[\frac{(x_t - \mu_k)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right]
\end{aligned}$$

where $\gamma_t(k)$ is the posterior probability of Gaussian k, given by:

$$\gamma_t(k) = \frac{w_k u_k(x_t)}{\sum_{j=1}^K w_j u_j(x_t)}$$

The gradients for all T descriptors in an image, are given by:

[9] α_k is a reparameterization of w_k, with $w_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^K \exp(\alpha_j)}$, so that $\forall k : w_k \geq 0,\ \sum_{k=1}^K w_k = 1$.

$$\mathcal{G}^X_{\alpha_k} = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} (\gamma_t(k) - w_k)$$
$$\mathcal{G}^X_{\mu_k} = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{x_t - \mu_k}{\sigma_k} \right)$$
$$\mathcal{G}^X_{\sigma_k} = \frac{1}{\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(k) \frac{1}{\sqrt{2}} \left[\frac{(x_t - \mu_k)^2}{\sigma_k^2} - 1 \right]$$

$\mathcal{G}^X_{\alpha_k}$ is a scalar, $\mathcal{G}^X_{\mu_k}$ and $\mathcal{G}^X_{\sigma_k}$ are D-dimensional vectors, where D is the dimension of the extracted feature. The final FV is a concatenation of the gradients $\mathcal{G}^X_{\alpha_k}$, $\mathcal{G}^X_{\mu_k}$ and $\mathcal{G}^X_{\sigma_k}$, for $k = 1, \ldots, K$. The total dimension of the FV is $E = (2D + 1)K$ [Sanchez et al., 2013].

The gradient of the weights brings little additional information and is not considered for the FV in [Perronnin et al., 2010]. The influence from each parameter in λ is illustrated in Figure 4.8.

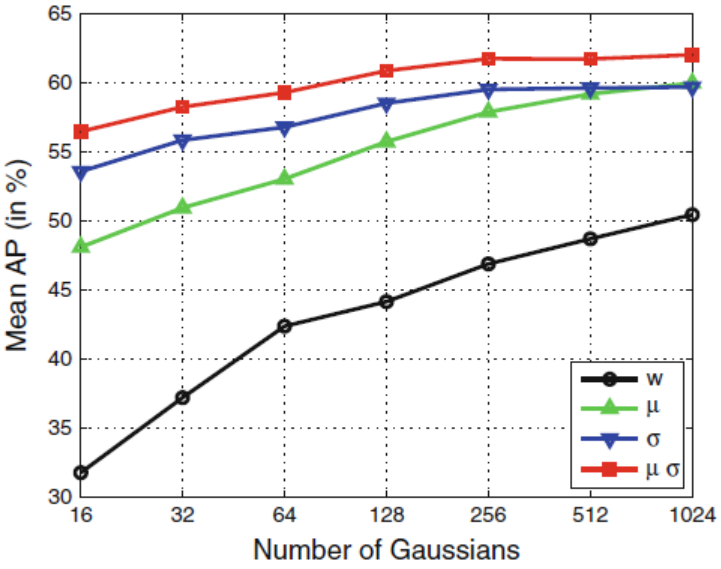


Figure 4.8: Impact on mAP of each parameter in λ [Sanchez et al., 2013]. The experiments were conducted on PASCAL VOC 2007 dataset with SIFT descriptor.

Relevance of Principal Component Analysis (PCA) for Gaussian Mixture Modeling (GMM)

[Sanchez et al., 2013] conduct experiments with and without the use of PCA, the results are illustrated in Figure 4.9. The experiment was conducted using a SIFT descriptor on

the PASCAL VOC 2007 dataset. Without the use of PCA, using the *original* descriptor to compute the FV the accuracy was at 54.5%, while with using PCA the accuracy is above 60% for all reductions to over 48 dimensions.

Informally, this effect is caused by the use of the diagonal covariance matrix for the GMM. The major and minor axes of the ellipses of a GMM with diagonal covariances are parallel to the axes of the coordinate system. Performing a PCA and projecting the data, the axis of the largest variance is employed as the new basis. This is performed for every dimension of the data. Therefore the modelling fits the data better when the axis of the Gaussians are *aligned* with the axes of the variance of the data.

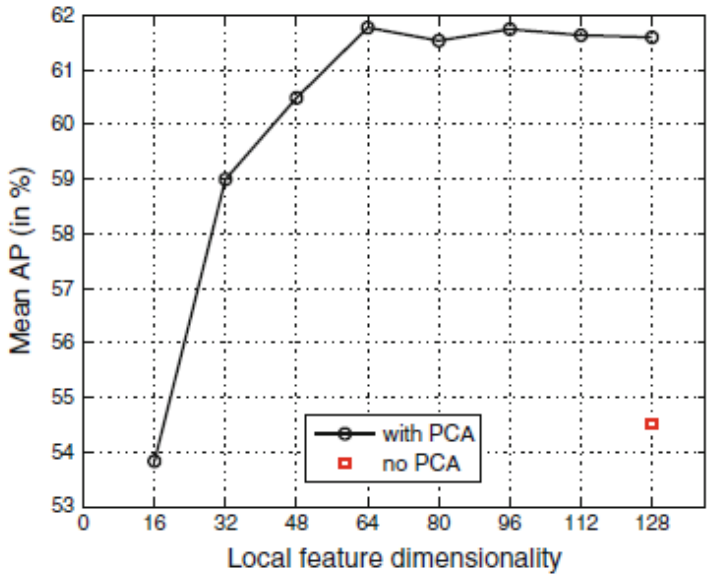


Figure 4.9: Illustrates mean average precision of image classification using the SIFT-descriptor on PASCAL VOC 2007 dataset with FV-encoding, applying various PCA reduction sizes(black) and no result without PCA(red) [Sanchez et al., 2013].

Improvements

l^2-Normalisation [Perronnin et al., 2010, Sanchez et al., 2013] show that applying l^2-normalisation to the FV limits the effect that different ratios of object information to background information between images has. This is the case as the objects sizes vary between the images in the set, as does the background (all non-object information). Through the normalisation the image-independent information is approximately discarded from the FV. Formal argumentations are given in [Perronnin et al., 2010] and in [Sanchez et al., 2013]. The comparisons in [Sanchez et al., 2013] on the PASCAL VOC 2007 dataset show an improvement of 4.6% and 5.5% in accuracy for SIFT and Local Colour

Statistics (LCS) descriptors respectively, using l^2-normalisation over a non-normalised FV (Table 4.7).

Power Normalisation To compute the Power Normalisation (PN), following operation is applied to each dimension in the feature vector:

$$z \leftarrow sign(z)\,|z|^{\rho},$$

with $0 < \rho \leq 1$. The value used for ρ in [Sanchez et al., 2013] is $\rho = \frac{1}{2}$, and referred to as *signed square rooting*. This operation acts like an explicit representation of the Hellinger or Bhattacharyya kernel. The higher the number of Gaussian components in the GMM, the sparser the FV becomes, empirically observed in Figure 4.10 (a), (b) and (c) in [Perronnin et al., 2010]. Through the PN the vector becomes denser (Figure 4.10 (d)), which impacts the dot-product positively [Perronnin et al., 2010]. It is also argued that the PN reduces the influence of descriptors that occur frequently within an image, correcting the incorrect independence assumption [Sanchez et al., 2013].

Spatial Pyramid The SP was introduced for the BoF framework, but can be applied to the FV in the same way [Sanchez et al., 2013]. The FV is computed for each region in the image and the resulting vectors are concatenated. If R is the number of regions, the dimensionality of the FV is $E = (2D + 1)KR$. The described routine in [Sanchez et al., 2013] uses $R = 4$ regions, one of the whole image and three equally splitted horizontal regions (1×1 + 3×1). Similar in [Kawano and Yanai, 2014] a FV is applied on a SP with three levels and $R = 8$ regions (1×1 + 2×2 + 3×1) for food image classification.

An alternative to SPM is a spatial extension of the local descriptors proposed in [Sánchez et al., 2012]. Here, this method will be refered to as Spatially Extended Descriptor (SED). After descriptor extraction, the location is embedded into the feature vector as two extra dimensions[10]. Let $m_t = [m_{x,t}, m_{y,t}]^{\intercal}$, where m are the 2D-coordinates of a local descriptor x_t in the image. The augmented feature vector $\hat{x}_t \in \mathbb{R}^{D+2}$ is defined as:

$$\hat{x}_t = \begin{pmatrix} x_t \\ m_{x,t}/W - 0.5 \\ m_{y,t}/H - 0.5 \end{pmatrix},$$

where W and H are the width and the height of the image respectively. The values are appended to the feature vector after PCA projection but before quantization with the GMM, therefore spatial information is directly captured in the quantization process. Compared to SPM the feature vector is significantly smaller, with a size of $E = (2D+3)K$. The smaller feature vector leads to significant memory reduction over using SPM.

[10] In [Sánchez et al., 2012] the patch scale of the descriptor neighbourhood is also encoded as a third parameter, beside the two location parameters.

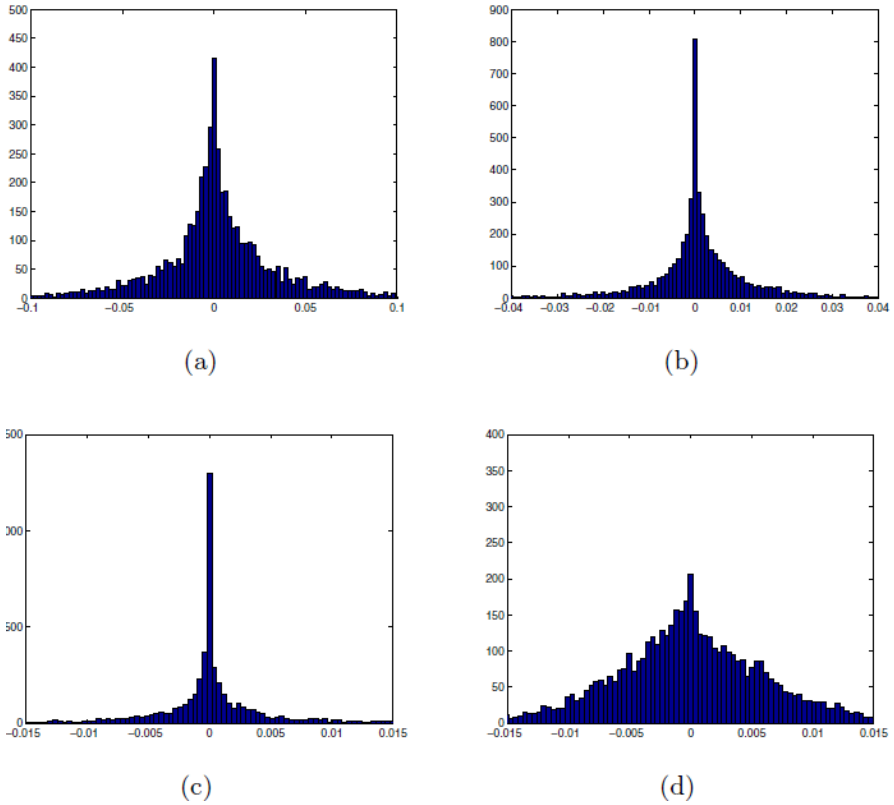


Figure 4.10: (a) (b) and (c) show the distribution of the l^2 normalised FV values of the first dimension from all computed FVs of all 5011 images of the PASCAL VOC 2007 dataset without the use of PN. For (a) 16 Gaussians were used, for (b) 64 and for (c) and (d) 256 Gaussians. For (d) PN was used. Notice the different scales. [Perronnin et al., 2010].

Effect of the improvements In Table 4.7 results of the contribution of each improvement is shown on experiments [Sanchez et al., 2013] conducted on the PASCAL VOC 2007 dataset for the SIFT and the LCS descriptor. Each of the three described improvements were applied separately and in combination with the other improvements. When all improvements are combined the increase in accuracy accounts to 12.2% and 17.4% for the two descriptors respectively.

PN	l^2	SP	SIFT	LCS
No	No	No	49.6	35.2
Yes	No	No	57.9 (+8.3)	47.0 (+11.8)
No	Yes	No	54.2 (+4.6)	40.7 (+5.5)
No	No	Yes	51.5 (+1.9)	35.9 (+0.7)
Yes	Yes	No	59.6 (+10.0)	49.7 (+14.7)
Yes	No	Yes	59.8 (+10.2)	50.4 (+15.2)
No	Yes	Yes	57.3 (+7.7)	46.0 (+10.8)
Yes	Yes	Yes	61.8 (+12.2)	52.6 (+17.4)

Table 4.7: Contribution of each improvement to the accuracy, individually and combined for each improvement. Results are for the SIFT and the LCS desriptors on the PASCAL VOC 2007 dataset [Sanchez et al., 2013].

4.3 Deep Convolutional Neural Networks (DCNNs)

The ILSVRC has been an annual competition since 2010, becoming a benchmark in large-scale object recognition [Russakovsky et al., 2015]. Since the introduction of DCNNs into the ILSVRC competition in 2012 by [Krizhevsky et al., 2012], DCNN is the winning method every year so far [Russakovsky et al., 2015].

4.3.1 Neural Networks (NNs)

The basis of DCNNs are Feedforward Neural Networks (FFNN), also called Multilayer Perceptrons (MLPs). The perceptron is introduced in [Rosenblatt, 1958], an artificial neuron with multiple inputs and an output. The output is computed by summing the weighed inputs, and then compared against a threshold to finally output x_i, $x_i \in \{0, 1\}$. The MLP is an extension that connects multiple perceptrons in a layered structure. An example of a simple MLP is illustrated in Figure 4.11. The decision making of a perceptron following a perceptron is based on the outputs of the perceptrons in the previous layer. Each new layers decisions can be interpreted as a higher level of abstraction in the decision process, than the decisions of its predecessing layer [Nielsen, 2015].

In a FFNN there are no connections in the model that are fed back into itself. Graphically the model relates to an acyclic graph, describing the relations between the functions. They are called *networks* because they consist of many different layered functions. E.g. a network with three functions f^1, f^2 and f^3 form the chain $f^3(f^2(f^1(x)))$. These chain structures are the most common use of Neural Networks (NNs). f^1 is called the first layer, f^2 the second, and so on. The final layer is called output layer. The length of the chains is also referred to as depth of a network. Therefore networks with long chained functions are also called deep networks [Goodfellow et al., 2016].

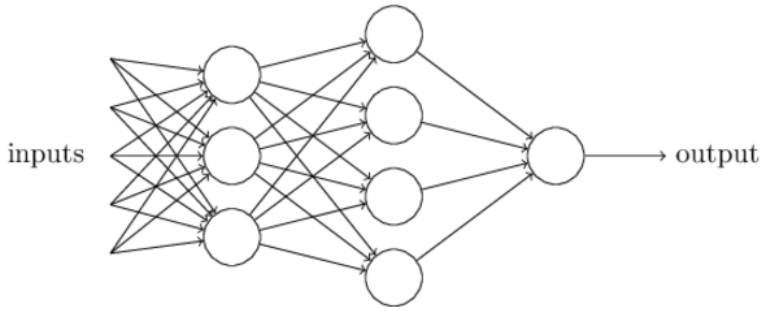


Figure 4.11: A simple example of an MLP with three layers [Nielsen, 2015].

A FFNN maps the input x to the categories y such that $y = f(wx + b)$, where w and b are the learned parameters, for weight and bias respectively, that result in the best approximation of the function [Goodfellow et al., 2016]. Sigmoid Neurons (SNs) are a variation of perceptrons with the addition that the output x_i, $x_i \in \mathbb{R} : 0 \leq x_i \leq 1$ and a very important property for the learning process: small changes in the weights result in small changes in the output of the SN [Nielsen, 2015]. This is accomplished choosing an appropriate output function $f(z) = f(wx + b)$. A common choice for the output function is the sigmoid function:

$$f(z) = \frac{1}{1 + e^{-z}}.$$

A similar function is modeled by the tanh neuron, where $f(z) = tanh(z)$. A different variation of the SN is the Rectified Linear Unit (ReLU), which is given by $f(z) = max(0, z)$. DCNNs with ReLUs train several times faster than networks with the SN or tanh neurons [Krizhevsky et al., 2012].

Learning

For a NN to work, an algorithm is applied to learn the weights and biases of the neurons, such that the output corresponds to the input of the layers [Nielsen, 2015]. To accomplish that, a cost function is defined and minimised. The cost function has a general form of

$$C(w, b) \equiv \frac{1}{2n} \sum |y(x) - a|^2 ,$$

where a is the output vector of the network and y is the vector of the actual categories of the samples in x. When the defined cost function is minimised, the output of the network is close to the true values. The algorithm used to *learn* the parameters to minimize the cost function is called *gradient descent*. To use calculus for finding the global minimum in practice is not feasible as there are millions of parameters in a NN [Nielsen, 2015]. Rather local minimas are estimated. The algorithm repeatedly computes the gradients of the cost function of all parameters and changes the parameters into the direction of the greatest *slope*. Doing this, the value of the cost functions iteratively decreases. The

learning rate parameter η is introduced to control the *speed* of the decrease. A formal definition of the gradient descent update rules are given by

$$w_k \to w'_k = w_k - \eta \frac{\partial C}{\partial w_k}, \text{ and}$$
$$b_l \to b'_l = b_l - \eta \frac{\partial C}{\partial b_l}.$$

To reduce computational cost, the Stochastic Gradient Descent (SGD) is used, where a small subset of x is selected to estimate the gradient [Nielsen, 2015].

The algorithm that computes the gradients through the network is called *backpropagation*. It was introduced in the 1970s, and became popular through the work of [Rumelhart et al., 1988, Nielsen, 2015]. The error is computed backwards through each layer of the network, starting from the final layer.

4.3.2 Convolutional Neural Networks (CNN))

Convolutional Layers (CL)

To consider the spatial structure of images, Convolutional Layers (CLs) are introduced into NNs. The layers are built using local receptive fields, that are slided across the input image. The amount of pixels the field is moved in each step, is called *stride* length. From each local receptive field a neuron is computed in the following hidden layer of the network. The weight parameters are shared by all the receptive fields of the entire layer, with the effect that one layer detects exactly the same feature[11], just at different locations in the input image [Nielsen, 2015]. The weight parameter w is a vector of the size of the receptive field, such that one weight value is learned for each pixel of the window. The bias parameter b is a scalar and also shared by all receptive fields in the entire layer.

The mapping from the image input to the hidden layer is also called *feature map*[12]. It is defined by the shared weights and bias, and can be interpreted as a filter or kernel. Multiple of such maps are computed for each hidden layer, each defining one feature. For the activation the convolution between the image and the weight matrix and the bias is computed in a manner of $a^1 = f(w * a^0 + b)$, where $*$ is the convolution operator.

Pooling Layer (PL)

Another component of most DCNNs are Pooling Layers (PLs). They usually follow CLs and serve the purpose of reducing the output data of a CL, which is input to the

[11] In the context of CNNs the term *feature* denotes a pattern that causes the neuron to activate, e.g. a similar edge or colour pattern at the position of the local receptive field.

[12] In the research literature the term *feature map* is used loosely. Both the weight function and the activation values of the neurons are sometimes called feature maps [Nielsen, 2015].

following CL. To accomplish the reduction, fixed regions of e.g. 2×2 or 3×3 activation neurons are sampled and reduced to a single neuron. Common pooling strategies are *max-pooling*, where the maximum activation of a region represents that region, the other activations are discarded. Informally that results in a reduction of the activations, as only the high activations (which translates in a feature being found) are passed to the next layer, also the exact position gets lost [Nielsen, 2015]. Another strategy commonly used is l^2-pooling, where the square root of the sum of the squares (l^2-norm) of the selected region is passed over to the next layer.

With the additions of CLs and PLs to the NN architecture, an example of a small but complete CNN, with two CLs and two PLs respectively, is illustrated in Figure 4.12.

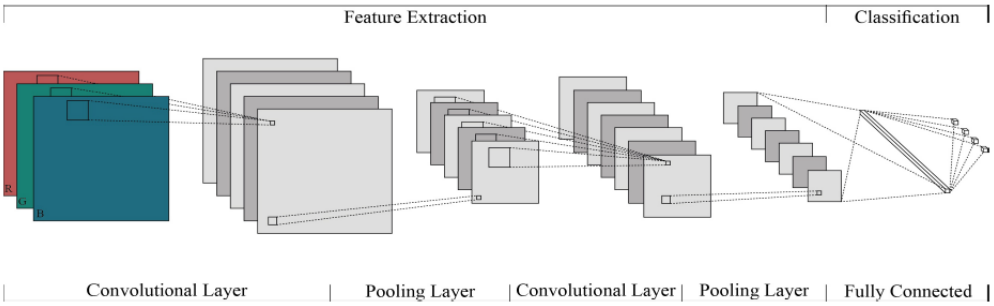


Figure 4.12: Typical DCNN architecture with two feature stages. The layer on the left shows the input layer (the image), connected with the first convolutional layer, containing five feature maps, followed by a pooling layer. The second stage is a convolutional layer with seven feature maps followed again by a pooling layer. The layer on the right is a fully connected layer, serving as the output layer [Christodoulidis et al., 2015].

Until recently, available dataset sizes were in the order of tens of thousands of images [Krizhevsky et al., 2012], e.g. NORB [LeCun et al., 2004], Caltech-101/256 [Griffin et al., 2007], CIFAR-10/100 [Krizhevsky, 2009]. Tasks such as the handwritten digit recognition perform very well with CNNs, achieving an error rate less than 0.3% [Krizhevsky et al., 2012].

In [Krizhevsky et al., 2012] the DCNN technique was applied to the large scale dataset ILSVRC-2010 and ILSVRC-2012. The computation time was reduced by highly optimised parts of the implementation, such as 2D convolution, that were executed on the Graphics Processing Unit (GPU), achieving the best results on the datasets, that were ever reported until 2012.

An explanation for the success of DCNNs lies in the deep structure following the initial feature extraction. The hand-crafted encoding techniques discussed earlier operate on low-level features, such as the SIFT or HOG descriptors, whereas the deep layered structure of a DCNN provide more complex generic mid-level image representations [Oquab et al., 2014].

4.3.3 Pre-training a DCNN

For a DCNN to learn the millions of parameters[13], a large scale dataset is necessary. In cases where only small training data is available, such as currently food recognition, the method of pre-training a network on large-scale generic data has been studied [Donahue et al., 2013, Oquab et al., 2014]. In first experiments of [Kawano and Yanai, 2014], traininig a DCNN with the UEC-FOOD100, they failed to outperform the results of their hand crafted feature approaches (results presented in [Kawano and Yanai, 2015b]), which they determine is due to the small scale of the dataset. To utilize DCNNs for small scale datasets, [Donahue et al., 2013] suggested to pre-train the features with a large scale dataset first in a supervised setting, and then transfer them to the actual data and labels the network is intended for.

In [Oquab et al., 2014] the transfer is executed on the DCNN of [Krizhevsky et al., 2012]. The final layer is removed and the parameters of all the other layers of the network are frozen. An additional fully connected layer and a new output layer(with the appropriate number of categories for the new task) are added to the net, and the parameters of the new layers are learned on the target data.

In the experiments of [Oquab et al., 2014] the AlexNet architecture was used for classification of the PASCAL VOC 2012 dataset. Without pre-training the mAP was 70.9%, with pre-training on the 1000 categories of the ILSVRC-2012 dataset, the mAP increased to 78.7%. With pre-training on the 1000 ILSVRC-2012 plus an additional 500 target task related categories selected from the other ImageNet categories, the mAP achieved was 82.8%.

Fine-tuning In addition to pre-training [Girshick et al., 2013] continue the learning with the SGD of also early layers of the network, with a learning rate of 0.001 ($1/10^{th}$ of the initial learning rate that was used for pre-training) for the early convolutional layers. A small learning rate has the effect that the SGD will try to find little improving changes but the features will not change completely. The *blank* fully connected layers are set to learn at a higher rate, as they are *learned* from scratch from random values to adapt to the new problem of new or different categories.

4.3.4 AlexNet

AlexNet is a commonly used identifier for the architecture of Alex Krizhevsky's 2012 network, described in [Krizhevsky et al., 2012]. The network consists of eight layers, five convolutional and three fully connected layers (illustrated in Figure 4.13). The first CL takes an input image of $224 \times 224 \times 3$, therefore the dataset has to be augmented to this size. The first, second and fifth CLs are followed by a max-pooling unit. As activation function, a ReLU follows each CL and each fully connected layer.

[13] The network in [Krizhevsky et al., 2012] consists of 60 million parameters.

For the ILSVRC-2012 dataset the network achieved a winning top-5 test error rate of 15.3%.

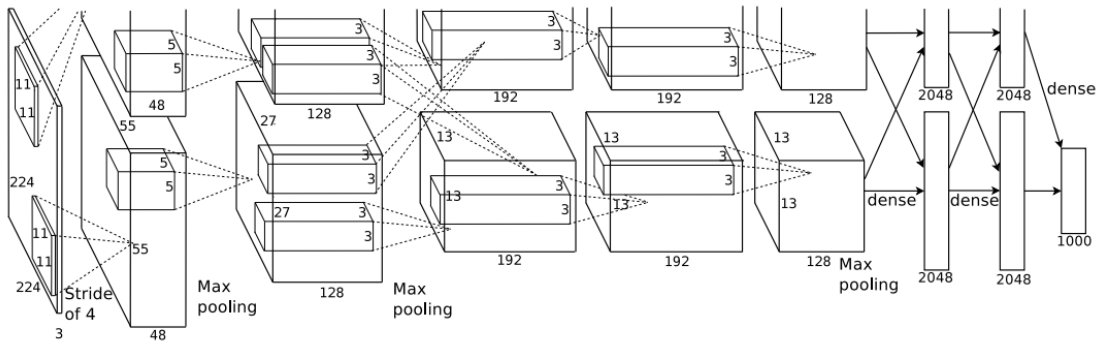


Figure 4.13: The network design from Krizhevsky et al. for two GPUs [Krizhevsky et al., 2012].

4.3.5 GoogLeNet

The network developed by [Szegedy et al., 2014] called *GoogLeNet* was a submission to the ILSVRC-2014 competition. It uses 12 times fewer parameters than Krizhevsky-architecture, while being significantly more accurate. A new structural concept which Szegedy et al. call Inception Module (IM) is introduced, inspired by [Lin et al., 2013].

An IM consists of convolutions of different filter sizes, in case of the GoogLeNet 1×1 convolutions are used to compute reductions followed by 3×3 and 5×5 convolutions and a 3×3 max-pooling layer. The 1×1 convolution reduces the number of feature maps. To accomplish this, the number of 1×1 kernels is set to a smaller value than there are feature maps from the previous output. The outputs of these components are concatenated to form one single vector which is the input to the next IM, illustrated in Figure 4.14. The network consists of multiple IMs, stacked like layers. The GoogLeNet architecture consists of nine such modules stacked on top of each other, resulting in a 22 layered[14] network. The benefits of this design is that various scales are processed simultaneously and then aggregated for the next stage. The dimension reduction of the data, before the convolutions are computed, makes it possible to run parallel convolutions with large patch sizes, allowing to increase the depth and the width of the network [Szegedy et al., 2014].

For the 1000-category ILSVRC-2012 dataset classification task, the top-5 error-rate achieved was 6.67%.

[14]Counting only the layers with parameters (convolutional and the fully connected layers).

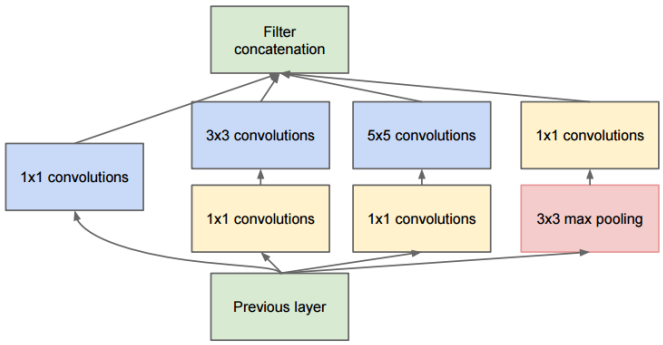


Figure 4.14: IM with the dimension reductions using 1×1 convolution kernels [Szegedy et al., 2014].

Paper	Details	Dataset size	Classes	Accuracy
[Kawano and Yanai, 2014]	DCNN pre-trained with ILSVRC + 1000 food-related categories (*DCNN-FOOD*)	100×100	100	71.80%
	combination of FV of colour and RootHOG and *DCNN-FOOD*	100×100	100	77.35%
	DCNN pre-trained with ILSVRC + 1000 food-related categories (*DCNN-FOOD*)	256×100	256	58.81%
	combination of FV of colour and RootHOG and *DCNN-FOOD*	256×100	256	63.77%
[Yanai and Kawano, 2015]	DCNN pre-trained with ILSVRC + 1000 food-related categories and fine-tuned(*DCNN-FOOD(tf2)*)	100×100	100	78.77%
	DCNN pre-trained with ILSVRC + 1000 food-related categories and fine-tuned (*DCNN-FOOD(ft)*)	256×100	256	67.57%
	DCNN pre-trained with ILSVRC + 1000 food-related categories and fine-tuned (*DCNN-FOOD(ft)*)	101000	101	70.41%
[Bossard et al., 2014]	Random Forest discriminative components mining with IFV(Improved Fisher Vector, 64 dict size) of SURF and Colour	101000	101	50.76%
	DCNN	101000	101	56.40%
[Myers et al., 2015]	GoogLeNet DCNN model, pre-trained on ILSVRC and fine-tuned on FOOD-101	101000	101	79.00%
	GoogLeNet DCNN model, pre-trained on ILSVRC and fine-tuned on FOOD-101	646	41	81.40%
[Wu et al., 2016]	DCNN GoogLeNet architecture	101000	101	69.64%
	DCNN GoogLeNet architecture + semantic hierarchy + label inference	101000	101	72.11%

Table 4.8: Summary of the results of experiments with DCNNs from researched papers.

4.4 Analysis

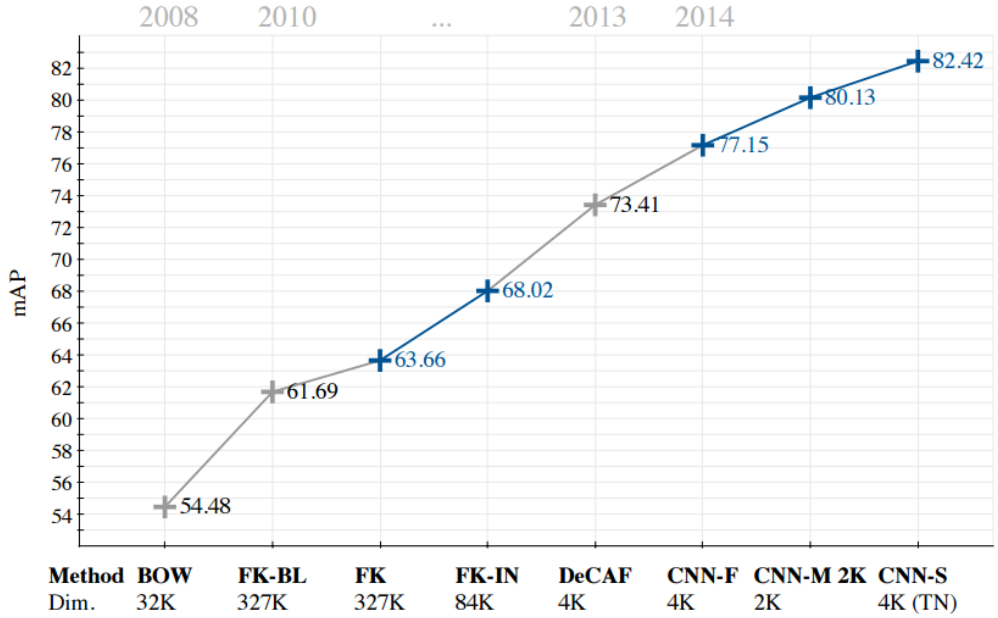


Figure 4.15: Progress made over the years with the corresponding state of the art methods of various BoF, FV and DCNN approaches. The results are for the image classification task of the PASCAL VOC 2007 dataset [Chatfield et al., 2014].

Methods of object recognition and classification have been studied extensively with countless fields of application. [Chatfield et al., 2014] emphasize on the details of the exact implementation[15], in experiments on the PACAL VOC 2007 dataset they compare a handcrafted FV encoding with various DCNN architectures. The aim was to conduct the experiments on a greater common ground, despite the differences of the approaches. E.g. the augmentation of the data including cropping and flipping the images, common for DCNNs, is also applied to the FV encoding, with the result of narrowing the gap between the methods from an original mAP of 63.66% reached with the basic FV, to an mAP of 68.02%[16] This still leaves a gap of a mAP level difference of 14% compared to the best DCNN technique, which reached 82.42% (with pre-training on the ILSVRC-2012 and fine-tuning). The authors also concatenated the feature vectors of both methods, but discovered little difference [Chatfield et al., 2014]. The combination of the DCNN

[15]The previous work [Chatfield et al., 2011], analyse implementation details of encoding techniques such asBoF, LLC and FV.

[16]The best FV-encoding technique (including the improvements from [Perronnin et al., 2010], using the SED technique instead of SPM) on the PASCAL VOC 2007 of a SIFT descriptor. The combination with a LCS descriptor did not improve the result significantly. 512 gaussians were used in the GMM quantization.

and the FV was also done by [Kawano and Yanai, 2014], achieving an improvement of 5.55% for the UEC-FOOD100, and 4.96% for the UEC-FOOD256 dataset.

For the comparison of the methods BoF, FV and DCNN in the experiments on food data in this thesis, the performance is expected to be in similar ranges as in previous object recognition experiments with generic classes, such as ImageNet or PASCAL VOC datasets. Performance comparisons of the three methods on the same dataset can be observed in Figure 4.15 [Chatfield et al., 2014]. Experiments of *FoodCam* (introduced in Section 3.4) support this assumption, they compare FV and DCNNs extensively [Kawano and Yanai, 2014], and also FV and BoF encoding [Hoashi et al., 2010, Kawano and Yanai, 2015b].

Despite all indication of DCNN outperforming a hand-crafted feature approach, the method can not be considered favourable in any case. In [Christodoulidis et al., 2015], where there is minimal training data available[17], through segmentation the image data is structured into overlapping patches of 32×32 pixels. The training and classification is based on the patches. In the specific setup that was used, BoF encoding achieves a classification accuracy of 82.2%, and the best performing DCNN achieves 84.9%, where the DCNN method takes 2.8 times the computation time of the BoF classification. In the following evaluation of their system in [Rhyner et al., 2016], the BoF approach was used. Another factor to consider analysing the close result of the methods is the very limited number of classes.

[17] This has been shown to be overcome with pre-training of the network [Donahue et al., 2013].

CHAPTER 5

Methodology

To the best of our knowledge, no large dataset of food images that includes ground truth data of nutritional information such as calorie content currently exists. [Beijbom et al., 2015] give an example for such a dataset, where dietitians assessed each image of the Menu-Match data for meta information such as calorie values. Through limitations in the context of this thesis, the *problem* is reduced to the core element of every dietary assessment task, the identification of the ingredients. We focus solely on the *computer vision perspective*. For this task, available generic food datasets that map images of food to predefined categories are used to get a deeper understanding of the recognition methods. The experimental focus of this work therefore lies on exploring and comparing state-of-the-art recognition methods on available generic food data.

5.1 Method

Three independent recognition techniques are implemented:

- Bag-of-Features (BoF)-encoding of texture and colour features
- Fisher-Vector (FV)-encoding of texture and colour features
- Deep Convolutional Neural Network (DCNN)

To compare the different approaches, they are evaluated on the same data. Three publicly available datasets are used:

- UEC-FOOD 100

- UEC-FOOD 256
- FOOD-101

The images of the multi-label UEC dataset are cropped to have only one class present on each image. This is done to reduce the problem from a multi-label classification to single-label classification. To accomplish this assumption on real-world data, multiple instances would have to be segmented from each other.

All relevant parameters of each technique are tested with different values to find an optimum. For each descriptor used by the BoF and FV encoding techniques, variations of the parameters are tested on the data and the best found combination is chosen. The general heuristic here is to first vary the sampling size and the descriptor size (neighbourhood). The best combination of the two parameters is then combined with different SP-resolutions. Two spatial sampling techniques are used: SPM and SED. Depending on the resulting total feature-vector size, the dictionary size (BoF) or the number of Gaussians (FV) are also varied.

For the combination of multiple descriptors for the BoF and FV approaches, two general approaches are compared. For *early descriptor fusion* (before classification) the best colour and best texture descriptors are combined. For *late fusion* (by SVM classification) the same procedure is followed, additionally all computed descriptors are combined.

Summary of the general heuristic used for both encoding strategies (BoF and FV):

- Variation of descriptor size and/or sampling size
- Variation of levels of the SP-encoding / SED
- Variation of dictionary size or number of Gaussians
- Variations of combining the descriptors

The evaluation of the results produced by each parameter variation is documented in Chapter 6.

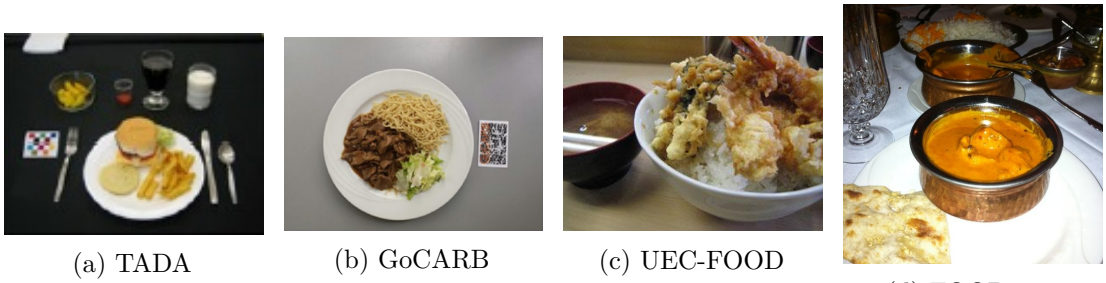
For the DCNN approach, two of the state-of-the-art network architectures are compared. The two architecture are:

- *AlexNet*
- *GoogLeNet*

On both architectures, the effect of using pre-trained network parameters is analysed for all datasets (small scale and large scale). Different configurations of fine-tuning to the food data is tested (e.g. different learning-rates for the individual layers).

5.2 Data

The general distinction in data of a dietary assessment system, is whether or not the images have been generated in a controlled setting. Figures 5.1a and 5.1b illustrate controlled images. Examples of controlled conditions are angle, background, lighting, food composition or fiducial markers used for colour correction and volume estimation. Figures 5.1c and 5.1d illustrate open-world images. The data of this type are images that are not bound by any restrictions other than that the object is at least partly visible on the image.



(a) TADA (b) GoCARB (c) UEC-FOOD (d) FOOD-101

Figure 5.1: Examples of images from the TADA, GoCARB, UEC-FOOD and FOOD-101 databases, illustrating different levels of image control.

5.2.1 Food datasets

In the literature discussed in Chapter 3 numerous sources for data of food images were found. This section provides a summary of some commonly used datasets used in food recognition.

GoCARB

In works of the GoCARB group [GoCARB Project, 2016], image data is used originating from the restaurants at the university hospital *Inselspital*, of the city of Bern, Switzerland. The dataset is small scale with a total of 1620 images. In [Anthimopoulos et al., 2015] the conditions of the data is explained as being restricted to elliptical plates with a flat base, single-dish images, fully visible food items and controlled lighting conditions. In contrast, [Dehais et al., 2015] describe the dataset with varying conditions in angles, plate backgrounds and lighting conditions. Next to the dish a reference card is visible, which is used for volume estimation. The number of food categories vary in the works, though its very limited, between six and eleven broad food categories. In [Anthimopoulos et al., 2015] the categories pasta, potatoes, meat, breaded food, rice, green, salad, mashed potatoes, carrots, and red beans are considered. This broad categorization is designed for their task of estimating carbohydrate content.

UEC-FOOD100/256

The dataset is structured into 100 food classes of a minimum of 100 images per class, and total of 12905 images [Kawano and Yanai, 2015b]. A category is defined as a connected compound of food ingredients, and the dataset is multi-labeled (i.e. an image might have multiple labels). Most categories are Japanese foods, but also international categories like hamburger and pizza etc. are present [Kawano and Yanai, 2015a]. The images were collected from web sources, representing *real world* data. In [Kawano and Yanai, 2015a] the set is expanded by an additional 156 categories[1], leading to a total number of 31394 images. The first 50 categories with example images are listed in Figure 5.2.

Bounding-box information is available for the UEC-FOOD datasets, that identifies the location and size of each dish belonging to a category, present for each image. For the experiments a new dataset is created that consists of solely the cropped regions of the bounding box definitions. This dataset is referred to as UEC-FOOD100/256-BB, creating a single-label dataset of the cropped rectangular regions.

The distribution of number of images per category is not uniform in this dataset. The minimum number of images for each category is 100 images. The non-uniform distribution is a result of the multi-labels, some categories are overrepresented (e.g. bowl of rice is present in many categories as a side dish).

Figure 5.2: The first 50 categories of the UEC-FOOD100 dataset [Kawano and Yanai, 2015b].

[1] Both datasets are publicly available at http://foodcam.mobi/dataset.html, accessed October 10, 2016

FOOD-101

[Bossard et al., 2014] created a real-world food dataset with 101 categories[2]. The images where collected from a social food web platform called foodspotting.com. The categories selected were the most popular and consistently named once on the platform. The dataset has a predefined split of 75% training and 25% test sets, where the test set is cleaned and the training set contains some outliers and wrong labels. The images are rescaled to maximum side length of 512 pixels and smaller images excluded. Each category contains 1000 images leading to a total of 101000 images over the entire dataset. [Myers et al., 2015] construct a variation named *FOOD-201* of 50% of the images of the FOOD-101 dataset, structuring the food into more detailed categories.

Restaurant based datasets

Two examples of restaurant restricted food item datasets are the **Menu-Match** dataset developed in [Beijbom et al., 2015], with 646 images of 41 food items from 3 restaurants. The calorie counts for each item was estimated by a dietitian who had access to ingredients and recipes. The dataset is publicly available[3].

Another dataset from restaurant images is the **Pittsburgh Fast Food Image Dataset** [Chen et al., 2009], which contains images of items from fast food chains, taken under laboratory conditions.

An extensive survey of these and other food datasets can be found in a very recent report by [Ciocca et al., 2017].

5.3 Implementation

The initial goal was an implementation in C++, using only open source tools and open libraries. For the development operating system, Ubuntu Linux was used, running inside a virtual machine. As an Integrated Development Environment (IDE) Eclipse was used. For computer vision and machine learning algorithms, following libraries were used: the Open Computer Vision (OpenCV) library, the Visual Lab Features (VLFeat) library [Vedaldi and Fulkerson, 2010] and SHOGUN [Sonnenburg et al., 2006], and for experiments with DCNN the CAFFE library [Yangqing, J., 2013] was used.

Due to unresolved issues of low performance results of the encoding methods of BoF and FV in the C++ implementation, a prototype in Matlab was implemented for faster and easier error search within the routines. The results from the final experiments (Chapter 6) were produced using the Matlab prototype code for both encoding variants.

[2]The dataset is publicly available at http://www.vision.ee.ethz.ch/datasets/food-101/, accessed October 10, 2016

[3]http://research.microsoft.com/menumatch/data/, accessed October 10, 2016

The hardware that was used for the Matlab experiments, was a server running CentOS Linux Version 7, equipped with 2 Intel® Xeon® CPU E5-2650 v3, with 10 cores of 2.30GHz each, with a total of 40 Threads and 128 GB RAM. The computation of the descriptors is performed on parallel threads. The maximum feature vector dimensions, so that clustering and SVM classification is executable on this hardware, is around 270000 dimensions for the UEC-FOOD datasets, and a maximum of 120000 dimensions for the FOOD101 dataset.

The following sections describe the details of the implementation of each descriptor, encoding method and details of CNN-framework and tools.

5.3.1 Descriptors

For the computation of the **LBP** descriptor, the implementation of the VLFeat library [Vedaldi and Fulkerson, 2010] is used. It follows the implementation of [Ojala et al., 2002], but the implementation is restricted to the 3×3 neighbourhood size and quantifies into 58 uniform patterns. The 3×3 neighbourhood has shown to perform best compared to bigger sizes [Ojala et al., 2002]. The image gets split into equal cells of a defined size and local histograms of the descriptor are computed for each cell.

Two implementations of the **HOG**-descriptor, were compared, both are implemented by the VLFeat library [Vedaldi and Fulkerson, 2010]. The original algorithm from [Dalal and Triggs, 2005] and the variation proposed in [Felzenszwalb et al., 2010], which is based on a parts model implementation. The second variant lead to better results in all tests. For all results presented in Chapter 6, the variant of [Felzenszwalb et al., 2010] is used. The histogram is computed over nine orientations, resulting in 31 total dimensions[4].

The **SIFT** descriptor is computed with the OpenCV library in the C++ implementation. In the MATLAB implementation the VLFeat library [Vedaldi and Fulkerson, 2010] is used. To compute SIFT on multiple scales simultaneously, the library provides the function `vl_phow()`, which computes densely sampled SIFT descriptors of four different neighbourhood sizes where $X \times Y = 4 \times 4,\ 6 \times 6,\ 8 \times 8$ and 10×10, for each sampled location. The library function also provides the computation of the descriptor on RGB, HSV and the Opponent colour space.

Colour Moment Invariants

The implementation of the descriptor follows the invariant generalised colour moments described by [Mindru et al., 2004]. These are rational expressions of the generalised colour moments. The formal definition of generalised colour moments is listed in Section 4.1.1.

The functions of the generalised colour moments that compose the invariant descriptor are:

[4] The exact implementation of the VLFeat library is documented in [VLFeat Library, 2016]

$$S_{02} = \frac{M_{00}^{2} * M_{00}^{0}}{(M_{00}^{1})^{2}}$$

$$D_{02} = \frac{M_{00}^{11} M_{00}^{00}}{M_{00}^{10} M_{00}^{01}}$$

$$S_{12} = \frac{M_{10}^{2} M_{01}^{0} M_{00}^{1} + M_{10}^{1} M_{01}^{2} M_{00}^{0} + M_{10}^{0} M_{01}^{1} M_{00}^{2} - M_{10}^{2} M_{01}^{1} M_{00}^{0} - M_{10}^{1} M_{01}^{0} M_{00}^{2} - M_{10}^{0} M_{01}^{2} M_{00}^{1}}{M_{00}^{2} M_{00}^{1} M_{00}^{0}}$$

$$D_{11} = \frac{M_{10}^{10} M_{01}^{01} M_{00}^{00} + M_{10}^{01} M_{01}^{00} M_{00}^{10} + M_{10}^{00} M_{01}^{10} M_{00}^{01} - M_{10}^{10} M_{01}^{00} M_{00}^{01} - M_{10}^{01} M_{01}^{10} M_{00}^{00} - M_{10}^{00} M_{01}^{01} M_{00}^{10}}{M_{00}^{10} M_{00}^{01} M_{00}^{00}}$$

$$D_{12}^{1} = \frac{M_{10}^{11} M_{01}^{00} M_{00}^{10} + M_{10}^{10} M_{01}^{11} M_{00}^{00} + M_{10}^{00} M_{01}^{10} M_{00}^{11} - M_{10}^{11} M_{01}^{10} M_{00}^{00} - M_{10}^{10} M_{01}^{00} M_{00}^{11} - M_{10}^{00} M_{01}^{11} M_{00}^{10}}{M_{00}^{11} M_{00}^{10} M_{00}^{00}}$$

$$D_{12}^{2} = \frac{M_{10}^{11} M_{01}^{00} M_{00}^{01} + M_{10}^{01} M_{01}^{11} M_{00}^{00} + M_{10}^{00} M_{01}^{01} M_{00}^{11} - M_{10}^{11} M_{01}^{01} M_{00}^{00} - M_{10}^{01} M_{01}^{00} M_{00}^{11} - M_{10}^{00} M_{01}^{11} M_{00}^{01}}{M_{00}^{11} M_{00}^{01} M_{00}^{00}}$$

$$D_{12}^{3} = \frac{M_{10}^{02} M_{01}^{00} M_{00}^{10} + M_{10}^{10} M_{01}^{02} M_{00}^{00} + M_{10}^{00} M_{01}^{10} M_{00}^{02} - M_{10}^{02} M_{01}^{10} M_{00}^{00} - M_{10}^{10} M_{01}^{00} M_{00}^{02} - M_{10}^{00} M_{01}^{020} M_{00}^{10}}{M_{00}^{02} M_{00}^{10} M_{00}^{00}}$$

$$D_{12}^{4} = \frac{M_{10}^{20} M_{01}^{01} M_{00}^{00} + M_{10}^{01} M_{01}^{00} M_{00}^{20} + M_{10}^{00} M_{01}^{20} M_{00}^{01} - M_{10}^{20} M_{01}^{00} M_{00}^{01} - M_{10}^{01} M_{01}^{20} M_{00}^{00} - M_{10}^{00} M_{01}^{01} M_{00}^{20}}{M_{00}^{20} M_{00}^{01} M_{00}^{00}}$$

M_{pq}^{i}, stands for M_{pq}^{i00}, M_{pq}^{0i0} or M_{pq}^{00i}, depending on which of the R, G or B colour channel the moment is computed on.

M_{pq}^{ij} stands for either M_{pq}^{ij0}, M_{pq}^{i0j} or M_{pq}^{0ij}, depending on which 2 colour channels the moment is computed on.

The concatenation of all functions above formulates the descriptor.

Colourpatch

The local colour descriptor is implemented following the description in [Kawano and Yanai, 2015b]. The sampled neighbourhood is divided into a 2×2 grid. The mean and variance values of each grid are computed for each colour channel in the RGB colour space, and the values concatenated into one vector. This results in a descriptor vector of a total of 24 dimensions.

Due to very slow runtime caused by multiple iterations over the images in Matlab, the routine alternatively is implemented within Matlab Executable (MEX)-files. MEX-files are binary compiled files. They provide an interface between Matlab and C++ files. Subroutines are dynamically linked and executed by the MATLAB interpreter.

For all descriptors and both encoding methods, **dense sampling** on a regular grid is the used sampling strategy.

Descriptor combination

For combination of the descriptors, first the *feature fusion* method is implemented. For the selection of the features, the best performing single descriptors for colour and texture are combined.

Combination on the decision level is also implemented, following the approach described in [Beijbom et al., 2015], where the score values from each linear SVM classifier (one per category) from all descriptors are concatenated. The resulting feature vector of one image has the size of number of descriptors times the number of classes.

5.3.2 Classification

For the two hand-crafted descriptor encoding approaches, there are two classifiers implemented for the original C++ implementation, a Random Forest and a linear SVM classifier. The Matlab prototype bases solely on linear SVM classification as it is fast and delivers good results. A single linear SVM is trained for each category and each feature type. All evaluations shown in Sections 6.1 and 6.2 are a result of classification using the SVM implementation from the VLFeat library.

For experiments of the encoding strategies, the training/test split used for the *UEC-FOOD* datasets, is 65 training images and 30 test images for each class. All experiments are conducted on the same set for training images and the same set of images for testing[5]. All images in the dataset are reduced in size to a maximum side-length of 480 pixels, if greater.

5.3.3 Bag-of-Features (BOF)

First 2M random descriptors are extracted from the training set, for the estimation of the dictionary. For clustering the VLFeat implementation of the k-means algorithm is used. For the k-means the *Elkan* strategy [Elkan, 2003] is used, an acceleration of the original algorithm from [Lloyd, 1982].

For the search of the closest cluster, a k-d-tree of the vocabulary is constructed, and then searched for each descriptor of the training and test set, for creating the feature-histogram. For computation of the homogeneous kernel feature map the kernel described in Section 4.2.1 is used. The implementations of the VLFeat library is used.

Features (BoF-histograms) are extracted for each segment of the SP. The SPM is implemented by normalising each histogram with the l^1-norm and concatenating the vectors. The resulting final feature vector is normalised with l^1-norm again.

Algorithm description

The image set is divided into training and test sets X_{train} and X_{test}, according to the used split ratio. The descriptors are extracted from all images. A subset of 2 million random descriptors of X_{train} is used to create the dictionary. The dictionary consists of

[5]For the DCNN classification results, a different split is used, due to different implementations.

the cluster-centers, computed with the *k-means* algorithm. For each image of the training set and the test set, the descriptors from each segment of the SP form a subset X_s. For each subset X_s the descriptors are quantised into a histogram over the dictionary entries with the use of the *KNN* algorithm. The histogram is normalised over the sum of the entries. The computed histograms from all segments of the SP are concatenated and normalised again. The final feature-vector is created by applying the kernel mapping function to the concatenated normalised histograms, to approximate the χ^2 kernel. The linear SVMs are trained with the resulting vectors from the training set. The resulting vectors from the test set are being classified with the computed SVMs. Further the results are evaluated.

5.3.4 Fisher Vector (FV)

The computation of the FV-encoding is implemented in C++ and in Matlab due to the unknown source of the initial erroneous code. For the final results presented in Chapter 6 the implementation from the VLFeat library is used because its execution is faster. The implementation also supports the improvements of the FV from [Perronnin et al., 2010] described in Section 4.2.2. An instruction of a fast computation of the FV can be found in [Sanchez et al., 2013, p. 227].

For the FV-encoding, both spatial strategies, SPM and SED are implemented and the results compared in Chapter 6. For the SPM, the FVs of all segments are concatenated as described in [Sanchez et al., 2013], i.e. each FV is unit-l^2 normalised independently. [Chatfield et al., 2014] report that it is common experience that linear classifiers are sensitive to the normalisation, and that particularly SVMs benefit from an l^2-normalisation.

For the SED strategy the spatial coordinates of the local patches are computed, normalised (following [Sánchez et al., 2012], described in Section 4.2.2) and concatenated with the PCA-projected descriptor before quantization with GMM.

Algorithm description

The image set is divided into training and test sets X_{train} and X_{test}, according to the used split ratio. The descriptors are extracted from all images. A subset X_{dict} of 2 million random descriptors of X_{train} is used to create the dictionary in form of a GMM. The mean μ and standard deviation σ are computed from X_{dict}. Each descriptor in X_{dict} is standardised to zero-mean and unit-variance. A PCA is performed on the standardised vectors. The vectors of X_{dict} are projected into the computed coordinate system. The aligned data is modelled with a GMM as the dictionary for the FV-encoding. For each image of the training set and the test set, the descriptors from each segment of the SP form a subset X_s. For each subset X_s the descriptors are standardised and projected into the PCA space. The FV is computed, power normalised and l^2-normalised. The computed FVs from all segments of the SP are concatenated. The linear SVMs are

trained with the resulting vectors from the training set. The resulting vectors from the test set are being classified with the computed SVMs. Further the results are evaluated.

5.3.5 Deep Convolutional Neural Networks (DCNNs)

For all experiments with DCNN, the deep learning framework CAFFE [Yangqing, J., 2013] is used. The framework was developed by Berkeley AI Research (BAIR) and by community contributors. The benefits of CAFFE are that it provides full support for training, testing, fine-tuning and deployment of networks, it is well documented and designed for exploring DCNNs. At the same time the implementation enables computationally high-performing executions [Jia et al., 2014]. Network models are independent from the framework and are stored as Google Protocol Buffers[6].

For deeper understanding of the framework, first the C++ interface is used. CAFFE also provides bindings for python and Matlab. For the documented benchmark experiments in Chapter 6, the *DIGITS* interface[7], an interactive deep learning GPU training system is used. It supports all necessary configuration for training, testing, pre-training and fine-tuning. For the visualisation of the feature maps of the first convolutional layer a python script is used. Within CAFFE, already ILSVRC-2012 pre-trained nets are downloaded from the **Caffe Model Zoo**[8], for each network architecture.

CAFFE uses a Lightning Memory-Mapped Database (LMDB)-format database for storage of the image data. The advantage of a memory-mapped database is that the requested data is accessed through a pointer that is mapped into application address space, without having to copy the data. The three used datasets FOOD101, UEC-FOOD100BB and UEC-FOOD256BB are converted into LMDB format. The used network architectures require a fixed image size, as both datasets consist of variable image sizes, a preceding image transformation is performed. For the UEC datasets, that represent segmented food dishes, the images are squashed to 256×256 pixels using the Joint Photographic Experts Group (JPEG) format, with a lossy quality setting of 90%. The reason for squashing the images instead of cropping, is to have the same input for the DCNN experiments as used for the encoding techniques, and to not loose information as the images are segmented and therefore do not contain much background information. The images of the FOOD101 are cropped, as described in [Krizhevsky et al., 2012], where the crop is performed on the centre of the image. Another pre-processing step is to subtract the mean RGB values of each pixel over the whole dataset. For the UEC datasets the test set is set to 30% of the images, for the FOOD101 it is 25%, the rest of the images is used for training. In the experiments two network architectures are compared, the *AlexNet* and *GoogLeNet*.

The hardware on which the experiments are carried out, is a single GeForce GTX Titan X GPU with 12GB of memory.

[6] `https://code.google.com/p/protobuf/`, accessed October 10, 2016
[7] `https://developer.nvidia.com/digits`, accessed October 10, 2016
[8] `http://caffe.berkeleyvision.org/model_zoo.html`, accessed October 10, 2016

5.4 Summary

For the experimental part of this thesis three techniques of object recognition for food categories are implemented, BoF, FV-encoding and DCNN. Each of the techniques are tested on three datasets of food images, two of them have 100 food categories, the third one 256. Each technique is analysed for an optimal combination of parameters, such as dictionary size, descriptor size or sampling step size for the hand-crafted descriptor based techniques, or learning rates and fine-tuning for DCNNs. The hand-crafted descriptor techniques are extended with two spatial sampling techniques, SPM and SED. Two DCNN network architectures, *AlexNet* and *GoogLeNet* are used. The results of every step in this process are documented in the next chapter.

CHAPTER 6

Evaluation and Results

To evaluate the isolated methodological performance in a fair comparison, one could compare the algorithms to the human dietitians that perform this task on basis of images. To the best of our knowledge such an evaluation has not yet been conducted. *Menu-Match* [Beijbom et al., 2015] conduct a comparison of their calorie estimation with estimations of a crowd-source system (non experts). The comparative study of [Rhyner et al., 2016], evaluates the estimations of their computational image analysis system in relation to the method of self-reporting of carbohydrate counting. Others, of the few evaluations of dietary assessment systems using image analysis that have been conducted, were in laboratory environments with few participants, comparing the estimations to determined ground-truth values. E.g. [Lee et al., 2012] and [Anthimopoulos et al., 2015] conducted such evaluations. In the scope of this thesis the evaluation is restricted on the accuracy of identification of food categories for all implemented object recognition techniques.

Experiments of three selected recognition methods are conducted. Hand-crafted feature extraction of multiple descriptors is performed and encoded with BoF and FV-encoding. Also classification with two DCNN-architectures is performed. In this chapter the results of the experiments are presented and analysed. First the isolated results of each individual descriptor are presented. This chapter documents the chosen heuristic for finding an optimal result that each descriptor can produce for the data. In particular the effect of the variation of certain parameters such as descriptor size or sampling size. The results of these experiments are not decisive on its own for a conclusion of the respective technique, as each descriptor contributes only in part to the final performance of the technique. The individual descriptors are further combined and the results are presented for all datasets, in Section 6.4 the results of all methods are compared.

Table 6.1 serves as a lookup table for abbreviations of various configurations of SPs. In the following tables presenting the results, each experiment using SPM, is labelled with a reference (A-H) to this table.

conf. name	levels	segments	pyramid structure
(A)	1	1	1×1
(B)	3	8	1×1, 2×2, 1×3
(C)	3	20	1×1, 4×4, 1×3
(D)	3	68	1×1, 8×8, 1×3
(E)	3	148	1×1, 12×12, 1×3
(F)	4	24	1×1, 2×2, 4×4, 1×3
(G)	5	88	1×1, 2×2, 4×4, 8×8, 1×3
(H)	6	344	1×1, 2×2, 4×4, 8×8, 16×16, 1×3

Table 6.1: Configurations of pyramid structures for SPM. This table is a reference for all following experiments for both BoF- and FV-encoding.

6.1 Bag-of-Features

After feature extraction and visual-word assignment the feature vector is expanded with homogeneous feature kernel mapping, and then trained and classified with a linear SVM in a one versus rest manner.

The k-means algorithm is initialised with random start locations for the clusters, therefore the result of the clustering varies on each execution. Due to the extensive runtime of the experiments and the high number of conducted experiments, each experiment is executed only once. The results of the BoF-encoding experiments, presented in following tables succumb to a certain variance. To analyse the variance, following experiments have been executed as a whole 20 times in a row with the same parameter settings, for three different dictionary sizes. The results of the experiment runs are summarised in Table 6.2. The 95% confidence interval is computed as a statistical quantity of the reliability of the results. The result of the variance and the confidence interval show a range of variation that is not significant in the comparison to other approaches.

dictionary size	300	1000	4000
μ	20.9115	22.1390	21.6065
σ	0.6628	0.7930	0.6475
0.95 conf. interval	$\mu \pm 0.2056$	$\mu \pm 0.2942$	$\mu \pm 0.1961$

Table 6.2: Results of 20 runs of the same experiments of BoF encoding of the LBP descriptor, with descriptor size of 4px, on the SP configuration (B), and the UEC-FOOD100BB dataset.

6.1.1 Colourpatch

Results of all experiments for the single descriptor Colourpatch (CP) are summarised in Table 6.3. Each sampling step size of $\{2, 4, 6, 8\}$ is tested in combination with each descriptor size of $\{4, 6, 8, 10, 12, 14, 16\}$, in experiments [a]–[β]. For step sizes 2 and 4 ([a]–[n]), the accuracy stays in a close range of 29.07–31.77% ([h] vs. [c]). For step sizes 6 and 8 the accuracy drops to the range of 24.60–29.43% ([v] vs. [r]). The mean accuracy for all descriptor sizes for each of the step sizes in $\{2, 4, 6, 8\}$ are 31.22, 30.24, 28.44 and 25.91% respectively. For descriptor size 8 e.g., a decrease of the sampling step size from 8 to 2, shows an improvement of around 6% ([x] vs. [c]). The combination with the best result ([c]), is used to experiment with the other parameters. Increasing the dictionary size over 1000, on the same SP configuration (B), does not have significant influence on the recognition accuracy (2.1% improvement) ([c] vs. [ζ]). Adding additional levels and increasing the resolution of the SP however improves the accuracy 4.03% ([ν] vs. [c]). Increasing the dictionary size to 3000 and using a slightly finer grid in the SP, results in an increase of 4.56% ([μ] vs. [c]). Highest accuracy on the UECFOOD100BB dataset with the densely sampled Colourpatch descriptor with BoF-encoding is 36.33%. The descriptor with the same parameters classified the UECFOOD256BB with an accuracy of 28.48% [σ]. Classification of the FOOD101 with the configuration ran out of memory, therefore the parameters are changed to reach a feature vector with a lower dimensionality. An accuracy of 27.15% is reached in [φ].

	sampling grid	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(a)	2px	4px	1000	24000	(B)	UECFOOD100BB	31.70%
(b)	2px	6px	1000	24000	(B)	UECFOOD100BB	30.83%
(c)	2px	8px	1000	24000	(B)	UECFOOD100BB	31.77%
(d)	2px	10px	1000	24000	(B)	UECFOOD100BB	31.07%
(e)	2px	12px	1000	24000	(B)	UECFOOD100BB	31.67%
(f)	2px	14px	1000	24000	(B)	UECFOOD100BB	31.17%
(g)	2px	16px	1000	24000	(B)	UECFOOD100BB	30.30%
(h)	4px	4px	1000	24000	(B)	UECFOOD100BB	29.07%
(i)	4px	6px	1000	24000	(B)	UECFOOD100BB	31.27%
(j)	4px	8px	1000	24000	(B)	UECFOOD100BB	30.80%
(k)	4px	10px	1000	24000	(B)	UECFOOD100BB	29.87%
(l)	4px	12px	1000	24000	(B)	UECFOOD100BB	30.70%
(m)	4px	14px	1000	24000	(B)	UECFOOD100BB	30.30%
(n)	4px	16px	1000	24000	(B)	UECFOOD100BB	29.67%

Continued on next page

Table 6.3 – continued from previous page

	sampling grid	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(o)	6px	4px	1000	24000	(B)	UECFOOD100BB	26.93%
(p)	6px	6px	1000	24000	(B)	UECFOOD100BB	28.53%
(q)	6px	8px	1000	24000	(B)	UECFOOD100BB	28.60%
(r)	6px	10px	1000	24000	(B)	UECFOOD100BB	29.43%
(s)	6px	12px	1000	24000	(B)	UECFOOD100BB	28.63%
(t)	6px	14px	1000	24000	(B)	UECFOOD100BB	28.37%
(u)	6px	16px	1000	24000	(B)	UECFOOD100BB	28.57%
(v)	8px	4px	1000	24000	(B)	UECFOOD100BB	24.60%
(w)	8px	6px	1000	24000	(B)	UECFOOD100BB	25.57%
(x)	8px	8px	1000	24000	(B)	UECFOOD100BB	25.83%
(y)	8px	10px	1000	24000	(B)	UECFOOD100BB	25.57%
(z)	8px	12px	1000	24000	(B)	UECFOOD100BB	27.23%
(α)	8px	14px	1000	24000	(B)	UECFOOD100BB	26.13%
(β)	8px	16px	1000	24000	(B)	UECFOOD100BB	26.47%
(γ)	2px	8px	500	12000	(B)	UECFOOD100BB	30.07%
(δ)	2px	8px	2000	48000	(B)	UECFOOD100BB	32.90%
(ε)	2px	8px	4000	96000	(B)	UECFOOD100BB	33.23%
(ζ)	2px	8px	8000	192000	(B)	UECFOOD100BB	33.87%
(η)	2px	8px	1000	3000	(A)	UECFOOD100BB	25.20%
(ϑ)	2px	8px	2000	6000	(A)	UECFOOD100BB	26.73%
(ι)	2px	8px	4000	12000	(A)	UECFOOD100BB	28.77%
(ϰ)	2px	8px	8000	24000	(A)	UECFOOD100BB	29.03%
(λ)	2px	8px	10000	30000	(A)	UECFOOD100BB	29.20%
(μ)	2px	8px	3000	180000	(C)	UECFOOD100BB	36.33%
(ν)	2px	8px	1000	204000	(D)	UECFOOD100BB	35.80%
(ξ)	2px	8px	500	222000	(E)	UECFOOD100BB	33.77%
(ο)	2px	8px	2000	144000	(F)	UECFOOD100BB	35.97%
(π)	2px	8px	3000	216000	(F)	UECFOOD100BB	36.23%
(ρ)	2px	8px	1000	264000	(G)	UECFOOD100BB	35.70%
(σ)	2px	8px	3000	180000	(C)	UECFOOD256BB	28.48%
(τ)	2px	8px	1000	204000	(D)	UECFOOD256BB	27.42%
(υ)	2px	8px	3000	216000	(F)	UECFOOD256BB	28.24%
(φ)	2px	8px	2000	120000	(C)	FOOD101	27.15%
(χ)	2px	8px	500	102000	(D)	FOOD101	23.40%

Continued on next page

Table 6.3 – continued from previous page

	sampling grid	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(ψ)	2px	8px	1000	72000	(F)	FOOD101	25.07%

Table 6.3: Results of Dense-Colourpatch descriptor with BoF encoding and spatial pyramid sampling.

6.1.2 Colour-Histogram

The results from all experiments conducted with the Colour Histogram (CH) descriptor are presented in Table 6.4. In experiments [a]–[x] in Table 6.4, each sampling step size of $\{2, 4, 6, 8\}$ is tested in combination with each descriptor size of $\{4, 6, 8, 10, 12, 14\}$. The best result is achieved with the smallest values for both parameters [a]. Increasing the dictionary size does not improve the accuracy significantly ([y]–[β]). In experiments [γ] – [η] the SPM is removed. The isolated improvement of SPM with 3 levels and a total of 8 segments, at a dictionary size of 8000 is 4.36% ([α] vs. [ζ]). In [ϑ]–[ν] the SP is increased with various configurations, and the dictionary size adapted so that the total descriptor size is at a maximum of around 250000. The best result is achieved with the settings in [ι], 34.53%. In experiments [ξ]–[ρ] the descriptor is computed on HSV and Opponent colour space (OP) (computation described in Table 4.1), but achieved a worse result compared to the same experiments on RGB colour space ([ϑ] and [ι]). The best result for the UECFOOD256BB is 25.92%, achieved in experiment [σ], and a lower dimensional configuration for the FOOD101 dataset reaches 23.71% in [φ].

	sampling grid	descriptor size	dictionary size	total dims.	colour space	pyramid struct.	dataset	accuracy
(a)	2px	4px	1000	24000	RGB	(B)	UECFOOD100BB	29.70%
(b)	2px	6px	1000	24000	RGB	(B)	UECFOOD100BB	28.97%
(c)	2px	8px	1000	24000	RGB	(B)	UECFOOD100BB	28.77%
(d)	2px	10px	1000	24000	RGB	(B)	UECFOOD100BB	27.43%
(e)	2px	12px	1000	24000	RGB	(B)	UECFOOD100BB	27.43%
(f)	2px	14px	1000	24000	RGB	(B)	UECFOOD100BB	27.73%

Continued on next page

Table 6.4 – continued from previous page

	sampling grid	descriptor size	dictionary size	total dims.	colour space	pyramid struct.	dataset	accuracy
(g)	4px	4px	1000	24000	RGB	(B)	UECFOOD100BB	28.77%
(h)	4px	6px	1000	24000	RGB	(B)	UECFOOD100BB	28.27%
(i)	4px	8px	1000	24000	RGB	(B)	UECFOOD100BB	27.40%
(j)	4px	10px	1000	24000	RGB	(B)	UECFOOD100BB	27.27%
(k)	4px	12px	1000	24000	RGB	(B)	UECFOOD100BB	25.20%
(l)	4px	14px	1000	24000	RGB	(B)	UECFOOD100BB	25.97%
(m)	6px	4px	1000	24000	RGB	(B)	UECFOOD100BB	26.17%
(n)	6px	6px	1000	24000	RGB	(B)	UECFOOD100BB	26.37%
(o)	6px	8px	1000	24000	RGB	(B)	UECFOOD100BB	26.70%
(p)	6px	10px	1000	24000	RGB	(B)	UECFOOD100BB	26.10%
(q)	6px	12px	1000	24000	RGB	(B)	UECFOOD100BB	24.67%
(r)	6px	14px	1000	24000	RGB	(B)	UECFOOD100BB	24.53%
(s)	8px	4px	1000	24000	RGB	(B)	UECFOOD100BB	25.13%
(t)	8px	6px	1000	24000	RGB	(B)	UECFOOD100BB	25.60%
(u)	8px	8px	1000	24000	RGB	(B)	UECFOOD100BB	24.87%
(v)	8px	10px	1000	24000	RGB	(B)	UECFOOD100BB	25.83%
(w)	8px	12px	1000	24000	RGB	(B)	UECFOOD100BB	24.57%
(x)	8px	14px	1000	24000	RGB	(B)	UECFOOD100BB	23.90%
(y)	2px	4px	500	12000	RGB	(B)	UECFOOD100BB	27.47%
(z)	2px	4px	2000	48000	RGB	(B)	UECFOOD100BB	29.30%
(α)	2px	4px	4000	96000	RGB	(B)	UECFOOD100BB	30.03%
(β)	2px	4px	8000	192000	RGB	(B)	UECFOOD100BB	30.23%
(γ)	2px	4px	1000	3000	RGB	(A)	UECFOOD100BB	22.70%
(δ)	2px	4px	2000	6000	RGB	(A)	UECFOOD100BB	24.23%
(ε)	2px	4px	4000	12000	RGB	(A)	UECFOOD100BB	25.60%
(ζ)	2px	4px	8000	24000	RGB	(A)	UECFOOD100BB	25.87%
(η)	2px	4px	10000	30000	RGB	(A)	UECFOOD100BB	26.73%
(ϑ)	2px	4px	3000	180000	RGB	(C)	UECFOOD100BB	34.33%
(ι)	2px	4px	1000	204000	RGB	(D)	UECFOOD100BB	34.53%
(ϰ)	2px	4px	500	222000	RGB	(E)	UECFOOD100BB	33.27%
(λ)	2px	4px	2000	144000	RGB	(F)	UECFOOD100BB	32.97%
(μ)	2px	4px	3000	216000	RGB	(F)	UECFOOD100BB	32.73%
(ν)	2px	4px	1000	264000	RGB	(G)	UECFOOD100BB	33.73%

Continued on next page

Table 6.4 – continued from previous page

	sampling grid	descriptor size	dictionary size	total dims.	colour space	pyramid struct.	dataset	accuracy
(ξ)	2px	4px	3000	180000	HSV	(C)	UECFOOD100BB	30.27%
(ο)	2px	4px	1000	204000	HSV	(D)	UECFOOD100BB	30.23%
(π)	2px	4px	3000	180000	OP	(C)	UECFOOD100BB	28.80%
(ρ)	2px	4px	1000	204000	OP	(D)	UECFOOD100BB	29.47%
(σ)	2px	4px	3000	180000	RGB	(C)	UECFOOD256BB	25.92%
(τ)	2px	4px	1000	204000	RGB	(D)	UECFOOD256BB	25.57%
(υ)	2px	4px	1500	90000	RGB	(C)	FOOD101	23.71%

Table 6.4: Results of Dense-Colour Histogram descriptor with BoF encoding and spatial pyramid sampling.

6.1.3 Colour Moments Invariants

Results of experiments with the CMI descriptor are summarised in Table 6.5. Each sampling step size of $\{2, 4, 6\}$ is tested in combination with each descriptor size of $\{2, 4, 6, 8, 10, 12, 14\}$ in experiments [a]–[u]. The best combination is achieved in experiment [e]. With these parameters the dictionary size is increased in [v]–[y], which brings an improvement of 6% ([e] vs. [y]). Increasing the levels of the SP ([z]–[δ]) achieves the best result in [γ] of 30.50% for the UEC-FOOD100BB dataset. The results for the UEC-FOOD256BB dataset with the same configuration is 25.17%. For the FOOD101 dataset, an accuracy of 18.04% is achieved. The Colourpatch descriptor discriminates the datasets significantly better, 5.83%, 0.75% and 5.67% for the UEC-FOOD100BB, UEC-FOOD256BB and the FOOD101 datasets respectively.

	sampling grid	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(a)	2px	2px	1000	24000	(B)	UECFOOD100BB	17.70%
(b)	2px	4px	1000	24000	(B)	UECFOOD100BB	21.03%
(c)	2px	6px	1000	24000	(B)	UECFOOD100BB	22.10%
(d)	2px	8px	1000	24000	(B)	UECFOOD100BB	22.80%
(e)	2px	10px	1000	24000	(B)	UECFOOD100BB	23.97%
(f)	2px	12px	1000	24000	(B)	UECFOOD100BB	23.90%
(g)	2px	14px	1000	24000	(B)	UECFOOD100BB	23.73%
(h)	4px	2px	1000	24000	(B)	UECFOOD100BB	15.30%
(i)	4px	4px	1000	24000	(B)	UECFOOD100BB	15.80%
(j)	4px	6px	1000	24000	(B)	UECFOOD100BB	18.13%
(k)	4px	8px	1000	24000	(B)	UECFOOD100BB	19.03%
(l)	4px	10px	1000	24000	(B)	UECFOOD100BB	20.60%
(m)	4px	12px	1000	24000	(B)	UECFOOD100BB	20.27%
(n)	4px	14px	1000	24000	(B)	UECFOOD100BB	20.47%
(o)	6px	2px	1000	24000	(B)	UECFOOD100BB	12.03%
(p)	6px	4px	1000	24000	(B)	UECFOOD100BB	14.47%
(q)	6px	6px	1000	24000	(B)	UECFOOD100BB	15.27%
(r)	6px	8px	1000	24000	(B)	UECFOOD100BB	16.77%
(s)	6px	10px	1000	24000	(B)	UECFOOD100BB	17.43%
(t)	6px	12px	1000	24000	(B)	UECFOOD100BB	18.23%
(u)	6px	14px	1000	24000	(B)	UECFOOD100BB	18.23%
(v)	2px	10px	500	12000	(B)	UECFOOD100BB	20.90%
(w)	2px	10px	2000	48000	(B)	UECFOOD100BB	25.57%
(x)	2px	10px	4000	96000	(B)	UECFOOD100BB	27.63%
(y)	2px	10px	8000	192000	(B)	UECFOOD100BB	29.97%
(z)	2px	10px	3000	180000	(C)	UECFOOD100BB	30.40%
(α)	2px	10px	1000	204000	(D)	UECFOOD100BB	28.70%
(β)	2px	10px	500	222000	(E)	UECFOOD100BB	25.37%
(γ)	2px	10px	3000	216000	(F)	UECFOOD100BB	30.50%
(δ)	2px	10px	1000	264000	(G)	UECFOOD100BB	30.10%
(ε)	2px	10px	3000	216000	(F)	UECFOOD256BB	25.17%
(ζ)	2px	10px	1500	90000	(C)	FOOD101	18.04%

Table 6.5: Results of Colour Moment Invariants descriptor with BoF encoding and spatial pyramid sampling.

6.1.4 SIFT

The implementation that is used (VLFeat Library), supports the computation on multiple descriptor sizes. First single descriptor sizes of $\{2, 4, 6, 8, 10\}$ are tested with each sampling size of $\{2, 4, 6\}$ ([a]–[o]). The best result achieves an accuracy of 38.17% [b]. Combining two of the descriptor sizes does not improve the accuracy ([p]–[u]). The combination of 4 descriptor sizes is tested on sampling step sizes of $\{2, 4, 6, 8, 10\}$, achieving the best result with step size 4: 39.83% [w]. Increasing the dictionary size ([β]–[ζ]), achieves an improvement of another 3% ([ζ] vs [w]). Increasing the SP levels is only feasible if the dictionary size is lowered, to avoid the usage of more memory than available. The maximum feature descriptor successfully classified on the described hardware (Section 5.3) for 100 classes had a dimensionality of 270336 per image (using 4 Bytes for float storage for each dimension, the whole UEC-FOOD100BB dataset representation takes up 9.6GiB of memory). The best result achieved for the UEC-FOOD100BB is 43.97% in [η]. For the UEC-FOOD256BB only one descriptor size is used [μ]. Sampling descriptors on many sizes produces a feature space that takes many days to compute the SVM for the 256 classes. The exact structure of the feature space is not investigated further. With the configuration in [μ] an accuracy of 33.50% is achieved. On the FOOD101 the descriptor achieved 35.74%.

	sampling grid	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(a)	2	2	1000	24000	(B)	UECFOOD100BB	36.47%
(b)	2	4	1000	24000	(B)	UECFOOD100BB	38.17%
(c)	2	6	1000	24000	(B)	UECFOOD100BB	35.70%
(d)	2	8	1000	24000	(B)	UECFOOD100BB	34.17%
(e)	2	10	1000	24000	(B)	UECFOOD100BB	1.00%
(f)	4	2	1000	24000	(B)	UECFOOD100BB	28.23%
(g)	4	4	1000	24000	(B)	UECFOOD100BB	35.77%
(h)	4	6	1000	24000	(B)	UECFOOD100BB	35.07%
(i)	4	8	1000	24000	(B)	UECFOOD100BB	32.83%
(j)	4	10	1000	24000	(B)	UECFOOD100BB	1.00%
(k)	6	2	1000	24000	(B)	UECFOOD100BB	22.80%
(l)	6	4	1000	24000	(B)	UECFOOD100BB	30.37%
(m)	6	6	1000	24000	(B)	UECFOOD100BB	32.40%
(n)	6	8	1000	24000	(B)	UECFOOD100BB	29.63%

Continued on next page

Table 6.6 – continued from previous page

	sampling grid	descriptor sizes	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(o)	6	10	1000	24000	(B)	UECFOOD100BB	1.00%
(p)	4	4, 6	1000	24000	(B)	UECFOOD100BB	37.77%
(q)	4	4, 8	1000	24000	(B)	UECFOOD100BB	38.00%
(r)	4	4, 10	1000	24000	(B)	UECFOOD100BB	37.83%
(s)	4	4, 12	1000	24000	(B)	UECFOOD100BB	38.00%
(t)	4	6, 8	1000	24000	(B)	UECFOOD100BB	35.97%
(u)	4	6, 10	1000	24000	(B)	UECFOOD100BB	37.17%
(v)	4	4, 6, 8	1000	24000	(B)	UECFOOD100BB	39.17%
(w)	4	4, 6, 8, 10	1000	24000	(B)	UECFOOD100BB	39.83%
(x)	2	4, 6, 8, 10	1000	24000	(B)	UECFOOD100BB	39.20%
(y)	6	4, 6, 8, 10	1000	24000	(B)	UECFOOD100BB	37.30%
(z)	8	4, 6, 8, 10	1000	24000	(B)	UECFOOD100BB	34.90%
(α)	10	4, 6, 8, 10	1000	24000	(B)	UECFOOD100BB	32.30%
(β)	4	4, 6, 8, 10	200	4800	(B)	UECFOOD100BB	31.07%
(γ)	4	4, 6, 8, 10	500	12000	(B)	UECFOOD100BB	35.90%
(δ)	4	4, 6, 8, 10	2000	48000	(B)	UECFOOD100BB	41.53%
(ε)	4	4, 6, 8, 10	4000	96000	(B)	UECFOOD100BB	42.73%
(ζ)	4	4, 6, 8, 10	8000	192000	(B)	UECFOOD100BB	42.87%
(η)	4	4, 6, 8, 10	3000	180000	(C)	UECFOOD100BB	43.97%
(ϑ)	4	4, 6, 8, 10	1000	204000	(D)	UECFOOD100BB	40.50%
(ι)	4	4, 6, 8, 10	500	222000	(E)	UECFOOD100BB	38.20%
(ϰ)	4	4, 6, 8, 10	3000	216000	(F)	UECFOOD100BB	43.63%
(λ)	4	4, 6, 8, 10	1000	264000	(G)	UECFOOD100BB	41.03%
(μ)	4	4	3000	180000	(C)	UECFOOD256BB	33.50%
(ν)	4	4, 6, 8, 10	2000	120000	(C)	FOOD101	35.74%

Table 6.6: Results of SIFT descriptor with BoF encoding.

6.1.5 LBP

Variations in descriptor size ([a]–[h]), achieve best accuracy at 4px ([c]). Decreasing the dictionary size from 1000 to 200 ([c] vs. [k]), decreases the accuracy only 2.9%. With

the limitations of a total vector size of around 250000[1] dimensions, the small feature vector can be sampled on a higher number of SP-levels ([u]–[z]), compared to experiments with the previous descriptors. Experiment [z] samples the descriptors on a total of 344 segments on six SP levels, achieving an accuracy of 29.03% on the UEC-FOOD100BB dataset. The same configuration achieves 20.83% on the UEC-FOOD256BB dataset. On the FOOD101 a lower dimensional configuration achieves 15.83%.

	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(a)	2px	1000	24000	(B)	UECFOOD100BB	20.97%
(b)	3px	1000	24000	(B)	UECFOOD100BB	20.60%
(c)	4px	1000	24000	(B)	UECFOOD100BB	22.80%
(d)	5px	1000	24000	(B)	UECFOOD100BB	22.13%
(e)	6px	1000	24000	(B)	UECFOOD100BB	21.73%
(f)	8px	1000	24000	(B)	UECFOOD100BB	20.83%
(g)	10px	1000	24000	(B)	UECFOOD100BB	20.23%[2]
(h)	12px	1000	24000	(B)	UECFOOD100BB	19.03%
(i)	4px	50	1200	(B)	UECFOOD100BB	13.37%
(j)	4px	100	2400	(B)	UECFOOD100BB	17.23%
(k)	4px	200	4800	(B)	UECFOOD100BB	19.93%
(l)	4px	300	7200	(B)	UECFOOD100BB	21.47%[2]
(m)	4px	400	9600	(B)	UECFOOD100BB	20.90%
(n)	4px	500	12000	(B)	UECFOOD100BB	21.73%
(o)	4px	2000	48000	(B)	UECFOOD100BB	22.03%
(p)	4px	4000	96000	(B)	UECFOOD100BB	22.00%
(q)	4px	8000	192000	(B)	UECFOOD100BB	21.53%
(r)	5px	1000	3000	(A)	UECFOOD100BB	12.70%
(s)	5px	5000	15000	(A)	UECFOOD100BB	13.60%
(t)	5px	10000	30000	(A)	UECFOOD100BB	OOM[3]
(u)	4px	4000	240000	(C)	UECFOOD100BB	23.83%
(v)	4px	1000	204000	(D)	UECFOOD100BB	27.90%
(w)	4px	500	222000	(E)	UECFOOD100BB	28.87%
(x)	4px	3000	216000	(F)	UECFOOD100BB	25.13%
(y)	4px	1000	264000	(G)	UECFOOD100BB	28.30%

Continued on next page

[1]Due to the hardware (described in Section 5.3) and time limitations.

[2]These experiments have been analysed for its variance, and executed 20 times. Results are listed in Table 6.2

[3]Produced an out of memory error.

Table 6.7 – continued from previous page

	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(z)	4px	200	206400	(H)	UECFOOD100BB	29.03%
(α)	4px	200	206400	(H)	UECFOOD256BB	20.83%
(β)	4px	500	102000	(D)	FOOD101	15.83%

Table 6.7: Results of LBP descriptor with BoF encoding.

6.1.6 RootHOG

The results of experiments with the Root-HOG descriptor are presented in Table 6.8. First the descriptor size is increased step by step in [a]–[h], resulting in a decrease in accuracy, for values higher than 4 [b]. Increasing the dictionary size ([i]–[o]), leads to an increase up to a dictionary size of 1000 words [b], then decreases further. Increasing the SP-levels over the used standard configuration E ([p]–[x]), does not have a significant effect on the accuracy, leading to slightly worse results than in [b]. On the UEC-FOOD100BB dataset the best result is 23.10%, achieved in [b]. For this dataset, increasing the spatial information or the granularity of the dictionary does not affect the discrimination of the data. On the UEC-FOOD256BB dataset the configurations achieve a significant improvement over configuration [b] ([z] and [α] vs. [y]), with an accuracy of 15.63% in [y]. On the FOOD101 dataset the descriptor achieves an accuracy of 11.12% [γ].

	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(a)	2	1000	24000	(B)	UECFOOD100BB	18.97%
(b)	4	1000	24000	(B)	UECFOOD100BB	23.10%
(c)	6	1000	24000	(B)	UECFOOD100BB	22.30%
(d)	8	1000	24000	(B)	UECFOOD100BB	22.53%
(e)	10	1000	24000	(B)	UECFOOD100BB	20.00%
(f)	12	1000	24000	(B)	UECFOOD100BB	20.27%
(g)	16	1000	24000	(B)	UECFOOD100BB	18.23%

Continued on next page

Table 6.8 – continued from previous page

	descriptor size	dictionary size	total dims.	pyramid struct.	dataset	accuracy
(h)	20	1000	24000	(B)	UECFOOD100BB	16.73%
(i)	4	50	1200	(B)	UECFOOD100BB	16.77%
(j)	4	100	2400	(B)	UECFOOD100BB	18.63%
(k)	4	200	4800	(B)	UECFOOD100BB	20.73%
(l)	4	500	12000	(B)	UECFOOD100BB	22.87%
(m)	4	2000	48000	(B)	UECFOOD100BB	22.77%
(n)	4	4000	96000	(B)	UECFOOD100BB	22.10%
(o)	4	8000	192000	(B)	UECFOOD100BB	21.73%
(p)	4	1000	3000	(A)	UECFOOD100BB	8.17%
(q)	4	1000	60000	(C)	UECFOOD100BB	18.90%
(r)	4	3000	180000	(C)	UECFOOD100BB	19.10%
(s)	4	1000	204000	(D)	UECFOOD100BB	19.97%
(t)	4	500	222000	(E)	UECFOOD100BB	21.63%
(u)	4	1000	72000	(F)	UECFOOD100BB	21.40%
(v)	4	3000	216000	(F)	UECFOOD100BB	18.47%
(w)	4	1000	264000	(G)	UECFOOD100BB	22.97%
(x)	4	200	206400	(H)	UECFOOD100BB	21.83%
(y)	4	1000	24000	(B)	UECFOOD256BB	10.35%
(z)	4	500	222000	(E)	UECFOOD256BB	14.45%
(α)	4	1000	264000	(G)	UECFOOD256BB	15.63%
(β)	4	1000	24000	(B)	FOOD101	6.99%
(γ)	4	1500	108000	(F)	FOOD101	11.12%

Table 6.8: Results of Root-HOG descriptor with BoF encoding.

6.1.7 Descriptor combinations

Feature Fusion

For the feature fusion (early fusion) the resulting histograms of the descriptor extraction and subsequent BoF-encoding are concatenated. The best performing colour descriptor and the best performing texture descriptor are selected. The results are presented

in Table 6.9. For the UEC-FOOD100BB dataset 50.87% is achieved. For the UEC-FOOD256BB, the SIFT descriptor is sampled on only one size per sampling location, to prevent classification issues with the resulting feature space. The exact cause is not investigated further. The result of the feature fusion on the UEC-FOOD256BB dataset is 42.20% and the FOOD101 dataset 42.39%.

descriptor	sampling grid	descriptor sizes	dictionary size	total dimensions	pyramid structure	dataset	accuracy
SIFT	4	4, 6, 8, 10	4000	96000	(B)	UECFOOD100BB	50.87%
CP	2	8	1500	108000	(F)		
SIFT	4	4	4000	96000	(B)	UECFOOD256BB	42.20%
CP	2	8	1500	108000	(F)		
SIFT	4	4, 6, 8, 10	1500	36000	(B)	FOOD101	42.39%
CP	2	8	1500	36000	(B)		

Table 6.9: Results of Feature Combinations with BoF encoding by histogram concatenation.

Decision Fusion

For fusion at decision level (*late fusion*), all computed descriptors have been combined by classifying the scores of the individual classifiers. The results are listed in Table 6.10. For the UEC-FOOD100BB dataset an accuracy of 52.30% is achieved, for the UEC-FOOD256BB dataset 44.32% and for the FOOD101 dataset 44.16%.

descriptor	sampling grid	descriptor sizes	dictionary size	total dimensions	pyramid structure	dataset	accuracy
SIFT	4	4, 6, 8, 10	1000	24000	(B)	UEC-100BB	48.10%
CP	2	8	1000	24000	(B)		
SIFT	4	4, 6, 8, 10	3000	216000	(F)	UEC-100BB	52.30%
CP	2	8	3000	180000	(C)		
LBP	–	4	200	206400	(H)		
Root-HOG	–	4	1000	264000	(G)		
CMI	2	10	3000	216000	(F)		
CH[a]	2	4	1000	204000	(D)		
SIFT	4	4	3000	180000	(C)	UEC-256BB	44.32%
CP	2	8	3000	180000	(C)		
LBP	–	4	200	206400	(H)		
Root-HOG	–	4	1000	264000	(G)		
CMI	2	10	3000	216000	(F)		
CH[a]	2	4	3000	180000	(C)		
SIFT		4, 6, 8, 10	2000	120000	(C)	FOOD101	44.16%
CP	2	8	2000	120000	(C)		
LBP	–	4	500	36000	(F)		
Root-HOG	–	4	1500	108000	(F)		
CMI	2	10	1500	90000	(C)		
CH[a]	2	4	1500	90000	(C)		

[a]RGB colour space.

Table 6.10: Results of Feature Combinations with BoF encoding by late fusion.

6.1.8 Summary

The best results for each descriptor and each dataset, and the results from both fusion strategies are summarised in Table 6.11. The best result from a single descriptor is achieved with the SIFT descriptor for all datasets. For the UEC-FOOD100BB an accuracy of 43.97% is reached, for the UEC-FOOD256BB 33.50% and the FOOD101 35.74%. The improvement of feature fusion compared to result of the single descriptor is 6.9%, 8.7% and 6.7% for the three datasets respectively, and the improvements of late fusion (of all descriptors) compared to the best performing descriptor are 8.33%, 10.82% and 8.42%.

The best results achieved with BoF-encoding are 52.30%, 44.32% and 44.16% for the three datasets. This results are compared to the other object recognition approaches at

method	UEC-FOOD100BB	UEC-FOOD256BB	FOOD101
CP	36.33	28.48	27.15
CH	34.53	25.92	23.71
CMI	30.50	25.17	18.04
SIFT	43.97	33.50	35.74
LBP	29.03	20.83	15.83
Root-HOG	23.10	15.63	11.12
feature fusion	50.87	42.20	42.39
late fusion	52.30	44.32	44.16

Table 6.11: A summary of the best results from all BoF-encoding experiments, for each descriptor and both fusion strategies and all three datasets.

the end of this chapter.

6.2 Fisher Vector

After feature extraction all descriptors of each image are encoded into a FV as a feature vector. The training and classification is performed with linear SVMs in a one versus rest manner. First the colour descriptors are discussed in Sections 6.2.1–6.2.3, the texture descriptors in Sections 6.2.4–6.2.6, and the combinations of the descriptors in Section 6.2.7.

6.2.1 Colourpatch

In experiments [a]–[e] the descriptor size is increased step by step, and in [f]–[i] the sampling size increased in combination with the best performing descriptor size. The best combination is the one of experiment [d]. Increasing the number of Gaussians of the GMM ([j]–[m]), increases the accuracy 4.34% ([d] vs. [m]), achieving 52.37%. This result supports the report of [Yanai and Kawano, 2015], who achieve an accuracy of 53.04% with the CP descriptor with FV-encoding on the same SP configuration on the UEC-FOOD100BB dataset. In [n]–[s] the SP configuration is varied. The number of Gaussians are decreased to keep the total feature vector size at around 250000. The combinations do not improve the accuracy compared to [m]. The precision of the model of the descriptor space, has a greater impact on the discrimination than the spatial information of the descriptors. Also, this descriptor performs better with the SED than with the SPM, for all other descriptors tested, this is not the case. The results for the UEC-FOOD256BB and the FOOD101 are 45.08% and 44.46% respectively ([u] and [v]).

From [w]–[ω] the descriptor was extended with the spatial coordinates. In [w]–[π] every combination of sampling step sizes in $\{2, 4, 6, 8\}$ and descriptor sizes in $\{4, 6, 8, 10, 12\}$

is tested, achieving the best result with the parameters in [y]. In [ρ]–[χ] the number of Gaussians in the GMM is increased, achieving 54.10% for the UEC-FOOD100BB dataset in [χ], the best result of all single descriptors tested, for both encoding variations. The same configuration achieved 45.38% and 48.38% for the UEC-FOOD256BB and the FOOD101 datasets respectively ([ψ] and [ω]).

	spatial pooling	sampling grid	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(a)	SPM	4px	4px	64	24	24576	(B)	UECFOOD100BB	44.13%
(b)	SPM	4px	6px	64	24	24576	(B)	UECFOOD100BB	47.50%
(c)	SPM	4px	8px	64	24	24576	(B)	UECFOOD100BB	47.87%
(d)	SPM	4px	10px	64	24	24576	(B)	UECFOOD100BB	48.03%
(e)	SPM	4px	12px	64	24	24576	(B)	UECFOOD100BB	47.50%
(f)	SPM	2px	10px	64	24	24576	(B)	UECFOOD100BB	47.97%
(g)	SPM	6px	10px	64	24	24576	(B)	UECFOOD100BB	46.03%
(h)	SPM	8px	10px	64	24	24576	(B)	UECFOOD100BB	45.33%
(i)	SPM	10px	10px	64	24	24576	(B)	UECFOOD100BB	42.40%
(j)	SPM	4px	10px	32	24	12288	(B)	UECFOOD100BB	44.27%
(k)	SPM	4px	10px	128	24	49152	(B)	UECFOOD100BB	50.13%
(l)	SPM	4px	10px	256	24	98304	(B)	UECFOOD100BB	51.77%
(m)	SPM	4px	10px	512	24	196608	(B)	UECFOOD100BB	52.37%
(n)	SPM	4px	10px	128	24	122880	(C)	UECFOOD100BB	51.47%
(o)	SPM	4px	10px	256	24	245760	(C)	UECFOOD100BB	51.80%
(p)	SPM	4px	10px	64	24	208896	(D)	UECFOOD100BB	48.73%
(q)	SPM	4px	10px	32	24	227328	(E)	UECFOOD100BB	44.97%
(r)	SPM	4px	10px	128	24	147456	(F)	UECFOOD100BB	51.60%
(s)	SPM	4px	10px	64	24	270336	(G)	UECFOOD100BB	49.47%
(t)	SPM	4px	10px	256	24	98304	(B)	UECFOOD256BB	44.41%
(u)	SPM	4px	10px	512	24	196608	(B)	UECFOOD256BB	45.08%
(v)	SPM	4px	10px	180	24	69120	(B)	FOOD101	44.46%
(w)	SED	2px	4px	64	24	3328	(x,y)	UECFOOD100BB	42.40%
(x)	SED	2px	6px	64	24	3328	(x,y)	UECFOOD100BB	43.43%
(y)	SED	2px	8px	64	24	3328	(x,y)	UECFOOD100BB	44.63%
(z)	SED	2px	10px	64	24	3328	(x,y)	UECFOOD100BB	44.03%
(α)	SED	2px	12px	64	24	3328	(x,y)	UECFOOD100BB	43.00%

Continued on next page

Table 6.12 – continued from previous page

	spatial pooling	sampling grid	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(β)	SED	4px	4px	64	24	3328	(x,y)	UECFOOD100BB	40.33%
(γ)	SED	4px	6px	64	24	3328	(x,y)	UECFOOD100BB	44.13%
(δ)	SED	4px	8px	64	24	3328	(x,y)	UECFOOD100BB	44.00%
(ε)	SED	4px	10px	64	24	3328	(x,y)	UECFOOD100BB	43.20%
(ζ)	SED	4px	12px	64	24	3328	(x,y)	UECFOOD100BB	43.27%
(η)	SED	6px	4px	64	24	3328	(x,y)	UECFOOD100BB	37.50%
(ϑ)	SED	6px	6px	64	24	3328	(x,y)	UECFOOD100BB	41.00%
(ι)	SED	6px	8px	64	24	3328	(x,y)	UECFOOD100BB	42.50%
(ϰ)	SED	6px	10px	64	24	3328	(x,y)	UECFOOD100BB	42.83%
(λ)	SED	6px	12px	64	24	3328	(x,y)	UECFOOD100BB	41.60%
(μ)	SED	8px	4px	64	24	3328	(x,y)	UECFOOD100BB	35.60%
(ν)	SED	8px	6px	64	24	3328	(x,y)	UECFOOD100BB	38.33%
(ξ)	SED	8px	8px	64	24	3328	(x,y)	UECFOOD100BB	40.50%
(ο)	SED	8px	10px	64	24	3328	(x,y)	UECFOOD100BB	41.13%
(π)	SED	8px	12px	64	24	3328	(x,y)	UECFOOD100BB	40.67%
(ρ)	SED	2px	8px	128	24	6656	(x,y)	UECFOOD100BB	47.37%
(σ)	SED	2px	8px	256	24	13312	(x,y)	UECFOOD100BB	49.93%
(τ)	SED	2px	8px	512	24	26624	(x,y)	UECFOOD100BB	51.63%
(υ)	SED	2px	8px	1024	24	53248	(x,y)	UECFOOD100BB	53.23%
(φ)	SED	2px	8px	2048	24	106496	(x,y)	UECFOOD100BB	54.07%
(χ)	SED	2px	8px	4096	24	212992	(x,y)	UECFOOD100BB	54.10%
(ψ)	SED	2px	8px	1024	24	53248	(x,y)	UECFOOD256BB	45.38%
(ω)	SED	2px	8px	1024	24	53248	(x,y)	FOOD101	48.38%

Table 6.12: Results of Colourpatch descriptor with FV encoding.

6.2.2 Colour-Histogram

The FV encoding of the CH descriptor do not lead to promising results, expecting an improvement compared to the BoF encoding results. Classification with the linear SVM is extensively more computationally expensive compared to the other descriptor classifications. A probable cause is that the seperability of the computed feature space is more complex, for each of the two-class problems (one versus rest). A similar issue occurred at classification of the SIFT descriptor with BoF encoding with descriptor size 10 (Table 6.6), and for classification of the FOOD101 dataset with FV-encoding of the CMI descriptor. Another possible cause might be an issue with the SVM implementation of the VLFeat library, but this is not verified. A different classifier is not tested on the mentioned experiments. To rule out the normalisation effect of the data before classification, the concatenated histograms (each histogram is l^1 normalised) are l^1 normalised again, in a second test l^2 normalised and also no additional normalisation is also tested. The l^2 normalisation achieved significantly higher results (as expect with FV, more information in Section 4.2.2), but still low results compared to BoF encoding or compared to other descriptors. The runtime is also not reduced by different normalisations of the feature vector.

	sampling grid	descriptor size	Gaussians	PCA reduction dims.	total dims.	colour space	pyramid struct.	dataset	accuracy
(a)	2px	4px	64	24	24576	RGB	(B)	UECFOOD100BB	13.20%
(b)	2px	6px	64	24	24576	RGB	(B)	UECFOOD100BB	12.73%
(c)	2px	8px	64	24	24576	RGB	(B)	UECFOOD100BB	12.67%
(d)	2px	10px	64	24	24576	RGB	(B)	UECFOOD100BB	12.10%
(e)	2px	12px	64	24	24576	RGB	(B)	UECFOOD100BB	14.50%
(f)	4px	4px	64	24	24576	RGB	(B)	UECFOOD100BB	12.70%
(g)	4px	6px	64	24	24576	RGB	(B)	UECFOOD100BB	12.37%
(h)	4px	8px	64	24	24576	RGB	(B)	UECFOOD100BB	14.40%
(i)	4px	10px	64	24	24576	RGB	(B)	UECFOOD100BB	12.20%
(j)	4px	12px	64	24	24576	RGB	(B)	UECFOOD100BB	13.93%
(k)	6px	4px	64	24	24576	RGB	(B)	UECFOOD100BB	12.40%
(l)	6px	6px	64	24	24576	RGB	(B)	UECFOOD100BB	11.90%
(m)	6px	8px	64	24	24576	RGB	(B)	UECFOOD100BB	12.30%
(n)	6px	10px	64	24	24576	RGB	(B)	UECFOOD100BB	12.73%
(o)	6px	12px	64	24	24576	RGB	(B)	UECFOOD100BB	11.33%
(p)	8px	4px	64	24	24576	RGB	(B)	UECFOOD100BB	11.63%

Continued on next page

Table 6.13 – continued from previous page

	sampling grid	descriptor size	Gaussians	PCA reduction dims.	total dims.	colour space	pyramid struct.	dataset	accuracy
(q)	8px	6px	64	24	24576	RGB	(B)	UECFOOD100BB	11.60%
(r)	8px	8px	64	24	24576	RGB	(B)	UECFOOD100BB	12.60%
(s)	8px	10px	64	24	24576	RGB	(B)	UECFOOD100BB	9.77%
(t)	8px	12px	64	24	24576	RGB	(B)	UECFOOD100BB	11.47%

Table 6.13: Results of Dense-Colour Histogram descriptor with FV encoding.

6.2.3 CMI

In experiments [a]–[u] each sampling step size in $\{2, 4, 6\}$ is tested in combination with each descriptor size of $\{2, 4, 6, 8, 10, 12, 14\}$. The best result is achieved in [e]. Increasing the number of Gaussians of the GMM ([v]–[y]), achieves a slight increases with 512 Gaussians of 1.36% compared to [e]. Variations of the configuration of the SP increase the accuracy 3.23% compared to [e] in experiment [δ], achieving 29.10%. For the UEC-FOOD256BB dataset a result of 23.71% is reached [ε]. Execution on the FOOD101 dataset was cancelled due to the long runtime for classification (several days). With a slightly different parameter setting of 64 Gaussians and pyramid configuration C, resulting in slightly smaller dimensionality the same issue occurred. The reason for the cause and the feature space is not explored any further, as the spatial extension of the descriptor achieved acceptable results [ψ], and is used for the descriptor combination (late fusion).

In experiments [η]–[ψ], the descriptor is extended with the spatial coordinates, achieving a result of 27.80% in [φ], 1.3% less than the best SPM configuration in [δ], with 19.7% of the feature vector size. For the UEC-FOOD256BB 22.92% are achieved and for the FOOD101 19.62%.

	spatial pooling	sampling grid	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(a)	SPM	2px	2px	64	24	24576	(B)	UECFOOD100BB	17.87%
(b)	SPM	2px	4px	64	24	24576	(B)	UECFOOD100BB	23.80%
(c)	SPM	2px	6px	64	24	24576	(B)	UECFOOD100BB	23.20%
(d)	SPM	2px	8px	64	24	24576	(B)	UECFOOD100BB	25.67%
(e)	SPM	2px	10px	64	24	24576	(B)	UECFOOD100BB	25.87%
(f)	SPM	2px	12px	64	24	24576	(B)	UECFOOD100BB	25.80%
(g)	SPM	2px	14px	64	24	24576	(B)	UECFOOD100BB	24.07%
(h)	SPM	4px	2px	64	24	24576	(B)	UECFOOD100BB	12.47%
(i)	SPM	4px	4px	64	24	24576	(B)	UECFOOD100BB	21.47%
(j)	SPM	4px	6px	64	24	24576	(B)	UECFOOD100BB	23.30%
(k)	SPM	4px	8px	64	24	24576	(B)	UECFOOD100BB	23.83%
(l)	SPM	4px	10px	64	24	24576	(B)	UECFOOD100BB	24.77%
(m)	SPM	4px	12px	64	24	24576	(B)	UECFOOD100BB	23.93%
(n)	SPM	4px	14px	64	24	24576	(B)	UECFOOD100BB	23.70%
(o)	SPM	6px	2px	64	24	24576	(B)	UECFOOD100BB	13.00%
(p)	SPM	6px	4px	64	24	24576	(B)	UECFOOD100BB	16.87%
(q)	SPM	6px	6px	64	24	24576	(B)	UECFOOD100BB	20.17%
(r)	SPM	6px	8px	64	24	24576	(B)	UECFOOD100BB	19.13%
(s)	SPM	6px	10px	64	24	24576	(B)	UECFOOD100BB	22.57%
(t)	SPM	6px	12px	64	24	24576	(B)	UECFOOD100BB	22.93%
(u)	SPM	6px	14px	64	24	24576	(B)	UECFOOD100BB	22.47%
(v)	SPM	2px	10px	32	24	12288	(B)	UECFOOD100BB	23.37%
(w)	SPM	2px	10px	128	24	49152	(B)	UECFOOD100BB	25.63%
(x)	SPM	2px	10px	256	24	98304	(B)	UECFOOD100BB	25.73%
(y)	SPM	2px	10px	512	24	196608	(B)	UECFOOD100BB	27.23%
(z)	SPM	2px	10px	256	24	245760	(C)	UECFOOD100BB	29.03%
(α)	SPM	2px	10px	64	24	208896	(D)	UECFOOD100BB	28.50%
(β)	SPM	2px	10px	32	24	227328	(E)	UECFOOD100BB	26.57%
(γ)	SPM	2px	10px	128	24	147456	(F)	UECFOOD100BB	28.57%
(δ)	SPM	2px	10px	64	24	270336	(G)	UECFOOD100BB	29.10%
(ε)	SPM	2px	10px	128	24	147456	(F)	UECFOOD256BB	23.71%

Continued on next page

Table 6.14 – continued from previous page

	spatial pooling	sampling grid	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(ζ)	SPM	2px	10px	180	24	69120	(B)	FOOD101	–%[4]
(η)	SED	2px	4px	64	24	3328	(x,y)	UECFOOD100BB	20.10%
(ϑ)	SED	2px	6px	64	24	3328	(x,y)	UECFOOD100BB	23.07%
(ι)	SED	2px	8px	64	24	3328	(x,y)	UECFOOD100BB	23.83%
(ϰ)	SED	2px	10px	64	24	3328	(x,y)	UECFOOD100BB	22.33%
(λ)	SED	2px	12px	64	24	3328	(x,y)	UECFOOD100BB	23.00%
(μ)	SED	4px	4px	64	24	3328	(x,y)	UECFOOD100BB	12.90%
(ν)	SED	4px	6px	64	24	3328	(x,y)	UECFOOD100BB	21.53%
(ξ)	SED	4px	8px	64	24	3328	(x,y)	UECFOOD100BB	22.37%
(ο)	SED	4px	10px	64	24	3328	(x,y)	UECFOOD100BB	23.47%
(π)	SED	4px	12px	64	24	3328	(x,y)	UECFOOD100BB	21.17%
(ρ)	SED	2px	8px	32	24	1664	(x,y)	UECFOOD100BB	20.63%
(σ)	SED	2px	8px	128	24	6656	(x,y)	UECFOOD100BB	24.73%
(τ)	SED	2px	8px	256	24	13312	(x,y)	UECFOOD100BB	25.80%
(υ)	SED	2px	8px	512	24	26624	(x,y)	UECFOOD100BB	27.47%
(φ)	SED	2px	8px	1024	24	53248	(x,y)	UECFOOD100BB	27.80%
(χ)	SED	2px	8px	1024	24	53248	(x,y)	UECFOOD256BB	22.92%
(ψ)	SED	2px	8px	1024	24	53248	(x,y)	FOOD101	19.62%

Table 6.14: Results of CMI descriptor with FV encoding.

6.2.4 SIFT

The descriptor is extracted on four sizes on each sampling location for all experiments. In [a]–[g] the sampling step size is increased step by step, achieving the best result in [a]. Reducing the dimensionality with PCA in [h]–[k] does not reduce the accuracy significantly between 64 and 128 dimensions, with a drop of 0.6% ([a] vs. [j]). The dimensionality of 80 [i] is chosen for the following experiments. In [l]–[n] the number of Gaussians of the GMM is increased to up to 128 Gaussians, slightly increasing the

[4]Experiment was cancelled due to a estimated classification runtime of several hours per classifier (one for each category).

accuracy to 43.77% on the UEC-FOOD100BB dataset [n]. Increasing the levels of the SP, with the cost of cutting back on the GMM precision, does not improve the results ([o] and [p]). For the UEC-FOOD256BB and the FOOD101 dataset the results are 36.63% and 37.46% respectively.

In experiments [s]–[ϑ], the descriptor is extended with the spatial coordinates, achieving a result of 35.50% in [t] for the UEC-FOOD100BB dataset, 8.27% less than the best SPM configuration in [n]. For the UEC-FOOD256BB 26.63% are achieved, and 36.02% for the FOOD101 dataset.

	spatial pooling	sampling grid	descriptor sizes	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(a)	SPM	2	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	41.93%
(b)	SPM	4	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	41.67%
(c)	SPM	6	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	40.80%
(d)	SPM	8	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	40.47%
(e)	SPM	10	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	39.53%
(f)	SPM	12	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	38.47%
(g)	SPM	14	4, 6, 8, 10	64	128	131072	(B)	UECFOOD100BB	36.90%
(h)	SPM	2	4, 6, 8, 10	64	100	102400	(B)	UECFOOD100BB	41.53%
(i)	SPM	2	4, 6, 8, 10	64	80	81920	(B)	UECFOOD100BB	42.30%
(j)	SPM	2	4, 6, 8, 10	64	64	65536	(B)	UECFOOD100BB	41.33%
(k)	SPM	2	4, 6, 8, 10	64	32	32768	(B)	UECFOOD100BB	39.60%
(l)	SPM	2	4, 6, 8, 10	16	80	20480	(B)	UECFOOD100BB	38.00%
(m)	SPM	2	4, 6, 8, 10	32	80	40960	(B)	UECFOOD100BB	41.27%
(n)	SPM	2	4, 6, 8, 10	128	80	163840	(B)	UECFOOD100BB	43.77%
(o)	SPM	2	4, 6, 8, 10	64	80	204800	(C)	UECFOOD100BB	43.73%
(p)	SPM	2	4, 6, 8, 10	64	80	245760	(F)	UECFOOD100BB	43.77%
(q)	SPM	2	4, 6, 8, 10	128	80	163840	(B)	UECFOOD256BB	36.63%
(r)	SPM	2	4, 6, 8, 10	64	80	81920	(B)	FOOD101	37.46%
(s)	SED	2	4, 6, 8, 10	64	128	16640	(x,y)	UECFOOD100BB	34.27%
(t)	SED	4	4, 6, 8, 10	64	128	16640	(x,y)	UECFOOD100BB	35.50%
(u)	SED	6	4, 6, 8, 10	64	128	16640	(x,y)	UECFOOD100BB	33.70%
(v)	SED	8	4, 6, 8, 10	64	128	16640	(x,y)	UECFOOD100BB	32.87%
(w)	SED	10	4, 6, 8, 10	64	128	16640	(x,y)	UECFOOD100BB	31.47%
(x)	SED	4	4, 6, 8, 10	64	100	13056	(x,y)	UECFOOD100BB	34.17%

Continued on next page

Table 6.15 – continued from previous page

	spatial pooling	sampling grid	descriptor sizes	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(y)	SED	4	4, 6, 8, 10	64	80	10496	(x,y)	UECFOOD100BB	34.57%
(z)	SED	4	4, 6, 8, 10	64	64	8448	(x,y)	UECFOOD100BB	33.90%
(α)	SED	4	4, 6, 8, 10	64	32	4352	(x,y)	UECFOOD100BB	30.10%
(β)	SED	4	4, 6, 8, 10	32	128	8320	(x,y)	UECFOOD100BB	34.40%
(γ)	SED	4	4, 6, 8, 10	128	128	33280	(x,y)	UECFOOD100BB	33.03%
(δ)	SED	4	4, 6, 8, 10	256	128	66560	(x,y)	UECFOOD100BB	32.20%
(ε)	SED	4	4, 6, 8, 10	512	128	133120	(x,y)	UECFOOD100BB	32.27%
(ζ)	SED	4	4, 6, 8, 10	1024	128	266240	(x,y)	UECFOOD100BB	31.20%
(η)	SED	4	4, 6, 8, 10	64	128	16640	(x,y)	UECFOOD256BB	26.63%
(ϑ)	SED	4	4, 6, 8, 10	64	128	16640	(x,y)	FOOD101	36.02%

Table 6.15: Results of SIFT descriptor with FV encoding.

6.2.5 LBP

The results from all experiments with the FV encoding of the LBP descriptor are presented in Table 6.16. In experiments [a]–[i] the descriptor size is increased step by step, reaching the best result in [d]. In [j]–[m] the number of Gaussians used for the GMM is increased, without affecting the accuracy. Increasing the levels of the SP shows an improvement of 3.3% ([n] vs. [d]), resulting in 29.40% for UEC-FOOD100BB. For the UEC-FOOD256BB and FOOD101 datasets, 23.59% and 22.39% are reached.

In [r]–[ι] the descriptor is extended with the spatial coordinates instead of using SPM. The best result for the UEC-FOOD100BB is 3.83% less accurate than the best result from using SPM ([η] vs. [n]). For the UEC-FOOD256BB dataset the result is 2.85% lower, and for the FOOD101 dataset 2.52%.

	spatial pooling	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(a)	SPM	2px	64	58	59392	(B)	UECFOOD100BB	23.53%
(b)	SPM	4px	64	58	59392	(B)	UECFOOD100BB	24.53%
(c)	SPM	5px	64	58	59392	(B)	UECFOOD100BB	25.43%
(d)	SPM	6px	64	58	59392	(B)	UECFOOD100BB	26.10%
(e)	SPM	8px	64	58	59392	(B)	UECFOOD100BB	25.57%
(f)	SPM	10px	64	58	59392	(B)	UECFOOD100BB	25.77%
(g)	SPM	12px	64	58	59392	(B)	UECFOOD100BB	24.20%
(h)	SPM	14px	64	58	59392	(B)	UECFOOD100BB	23.87%
(i)	SPM	16px	64	58	59392	(B)	UECFOOD100BB	23.27%
(j)	SPM	6px	16	58	14848	(B)	UECFOOD100BB	24.63%
(k)	SPM	6px	32	58	29696	(B)	UECFOOD100BB	24.97%
(l)	SPM	6px	128	58	118784	(B)	UECFOOD100BB	25.90%
(m)	SPM	6px	192	58	178176	(B)	UECFOOD100BB	26.43%
(n)	SPM	6px	64	58	148480	(C)	UECFOOD100BB	29.40%
(o)	SPM	6px	64	58	178176	(F)	UECFOOD100BB	29.00%
(p)	SPM	6px	64	58	148480	(C)	UECFOOD256BB	23.59%
(q)	SPM	6px	32	58	74240	(C)	FOOD101	22.39%
(r)	SED	4px	64	58	7680	(x,y)	UECFOOD100BB	21.50%
(s)	SED	6px	64	58	7680	(x,y)	UECFOOD100BB	22.93%
(t)	SED	8px	64	58	7680	(x,y)	UECFOOD100BB	22.80%
(u)	SED	10px	64	58	7680	(x,y)	UECFOOD100BB	21.97%
(v)	SED	12px	64	58	7680	(x,y)	UECFOOD100BB	21.70%
(w)	SED	14px	64	58	7680	(x,y)	UECFOOD100BB	21.33%
(x)	SED	16px	64	58	7680	(x,y)	UECFOOD100BB	19.43%
(y)	SED	6px	64	50	6656	(x,y)	UECFOOD100BB	21.93%
(z)	SED	6px	64	40	5376	(x,y)	UECFOOD100BB	20.20%
(α)	SED	6px	64	30	4096	(x,y)	UECFOOD100BB	20.13%
(β)	SED	6px	16	58	1920	(x,y)	UECFOOD100BB	16.77%
(γ)	SED	6px	32	58	3840	(x,y)	UECFOOD100BB	20.67%
(δ)	SED	6px	128	58	15360	(x,y)	UECFOOD100BB	24.20%
(ε)	SED	6px	256	58	30720	(x,y)	UECFOOD100BB	24.90%
(ζ)	SED	6px	512	58	61440	(x,y)	UECFOOD100BB	25.50%
(η)	SED	6px	1024	58	122880	(x,y)	UECFOOD100BB	25.57%

Continued on next page

Table 6.16 – continued from previous page

	spatial pooling	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(ϑ)	SED	6px	1024	58	122880	(x,y)	UECFOOD256BB	20.74%
(ι)	SED	6px	512	58	61440	(x,y)	FOOD101	19.87%

Table 6.16: Results of LBP descriptor with FV encoding.

6.2.6 Root-HOG

In experiments [a]–[h], the descriptor size is increased, achieving the best result in [c]. In [i]–[l] the number of Gaussians of the GMM is varied, which does not improve the result from [c]. For the experiments using between 32 and 256 Gaussians the accuracy stays within a range of 1.4% ([i] – [l] and [c]). In [m]–[p] the SP configuration is varied achieving the best result of 32.93% in [o], an increase of 4.23% compared to [c]. The same configuration achieved 23.11% and 21.77% on the UEC-FOOD256BB and the FOOD101 respectively ([q] and [r]).

For experiments with spatially extending the descriptor ([s] – [ε]), the achieved accuracies are worse than using SPM for all variations of parameters and all three datasets. For the UEC-FOOD100BB the accuracy drops 7.56%, for the UEC-FOOD256BB 3.96%, and for the FOOD101 dataset 4.45%.

	spatial pooling	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(a)	SPM	2px	64	31	31744	(B)	UECFOOD100BB	25.03%
(b)	SPM	3px	64	31	31744	(B)	UECFOOD100BB	27.03%
(c)	SPM	4px	64	31	31744	(B)	UECFOOD100BB	28.70%
(d)	SPM	6px	64	31	31744	(B)	UECFOOD100BB	28.67%
(e)	SPM	8px	64	31	31744	(B)	UECFOOD100BB	27.53%

Continued on next page

Table 6.17 – continued from previous page

	spatial pooling	descriptor size	Gaussians	PCA reduction dims.	total dims.	pyramid struct.	dataset	accuracy
(f)	SPM	10px	64	31	31744	(B)	UECFOOD100BB	26.97%
(g)	SPM	12px	64	31	31744	(B)	UECFOOD100BB	24.93%
(h)	SPM	14px	64	31	31744	(B)	UECFOOD100BB	24.80%
(i)	SPM	4px	16	31	7936	(B)	UECFOOD100BB	24.73%
(j)	SPM	4px	32	31	15872	(B)	UECFOOD100BB	27.30%
(k)	SPM	4px	128	31	63488	(B)	UECFOOD100BB	27.87%
(l)	SPM	4px	256	31	126976	(B)	UECFOOD100BB	28.03%
(m)	SPM	4px	128	31	158720	(C)	UECFOOD100BB	30.47%
(n)	SPM	4px	32	31	134912	(D)	UECFOOD100BB	29.37%
(o)	SPM	4px	64	31	95232	(F)	UECFOOD100BB	32.93%
(p)	SPM	4px	128	31	190464	(F)	UECFOOD100BB	29.87%
(q)	SPM	4px	64	31	95232	(F)	UECFOOD256BB	23.11%
(r)	SPM	4px	64	31	79360	(C)	FOOD101	21.77%
(s)	SED	2px	64	31	4224	(x,y)	UECFOOD100BB	18.43%
(t)	SED	4px	64	31	4224	(x,y)	UECFOOD100BB	21.80%
(u)	SED	6px	64	31	4224	(x,y)	UECFOOD100BB	22.17%
(v)	SED	8px	64	31	4224	(x,y)	UECFOOD100BB	21.60%
(w)	SED	10px	64	31	4224	(x,y)	UECFOOD100BB	21.00%
(x)	SED	12px	64	31	4224	(x,y)	UECFOOD100BB	20.43%
(y)	SED	6px	16	31	1056	(x,y)	UECFOOD100BB	15.40%
(z)	SED	6px	32	31	2112	(x,y)	UECFOOD100BB	18.73%
(α)	SED	6px	128	31	8448	(x,y)	UECFOOD100BB	23.60%
(β)	SED	6px	256	31	16896	(x,y)	UECFOOD100BB	24.47%
(γ)	SED	6px	512	31	33792	(x,y)	UECFOOD100BB	25.37%
(δ)	SED	6px	512	31	33792	(x,y)	UECFOOD256BB	19.15%
(ε)	SED	6px	512	31	33792	(x,y)	FOOD101	17.32%

Table 6.17: Results of Root-HOG descriptor with FV encoding.

6.2.7 Descriptor combinations

Feature Fusion

For the feature fusion (early fusion) the resulting FVs of the individual descriptors are concatenated. The results are presented in Table 6.18. The best performing colour descriptor and the best performing texture descriptor is selected. The results for the UEC-FOOD100BB dataset is 56.90%. Combining FVs resulting from both SPM and SED, does not lead to good results, therefore the spatial strategies are not mixed. For the UEC-FOOD100BB, the accuracy reached with the SED strategy takes up less than half of the dimensionality of the feature vector than the SPM vector, with an accuracy drop of only 1.47%. For the UEC-FOOD256BB dataset, the SPM strategy is used and a result of 51.52% is reached. For the FOOD101 dataset, the SED strategy is used due to the good results in relation to a smaller descriptor size with an accuracy of 51.18%.

descriptor	sampling grid	descriptor sizes	Gaussians	pca dimensions	pyramid structure	dimensions	total dimensions	dataset	accuracy
SIFT	2	4, 6, 8, 10	64	80	(B)	81920	155648	UEC-100BB	56.90%
CP	2	10	64	24	(F)	73728			
SIFT	2	4, 6, 8, 10	128	80	(x,y)	20992	74240	UEC-100BB	55.43%
CP	2	10	1024	24	(x,y)	53248			
SIFT	2	4, 6, 8, 10	64	80	(B)	81920	180224	UEC-256BB	51.52%
CP	4	10	256	24	(B)	98304			
SIFT	2	4, 6, 8, 10	128	80	(x,y)	20992	74240	FOOD101	51.18%
CP	2	10	1024	24	(x,y)	53248			

Table 6.18: Results of Feature Combinations with FV encoding by feature vector concatenation.

Decision Fusion

For fusion at decision level (*late fusion*), all computed descriptors are being combined by classifying the scores of the individual classifiers. The results are listed in Table 6.19. For the UEC-FOOD100BB dataset an accuracy of 58.33% is achieved, for the UEC-FOOD256BB dataset 53.14% and for the FOOD101 dataset 55.62%.

descriptor	sampling grid	descriptor sizes	Gaussians	pca dimensions	pyramid structure	dimensions	dataset	accuracy
SIFT	2	4, 6, 8, 10	128	80	(B)	163840		
CP	2	8	2048	24	(x,y)	106496		
CMI	2	8	1024	24	(x,y)	53248	UEC-100BB	58.33%
LBP	–	6	64	58	(C)	148480		
Root-HOG	–	4	64	31	(F)	95232		
SIFT	2	4, 6, 8, 10	128	80	(B)	163840		
CP	2	8	1024	24	(x,y)	53248		
CMI	2	10	128	24	(F)	147456	UEC-256BB	53.14%
LBP	–	6	64	58	(C)	148480		
Root-HOG	–	4	64	31	(F)	95232		
SIFT	2	4, 6, 8, 10	64	80	(B)	81920		
CP	2	8	1024	24	(x,y)	53248		
CMI	2	8	1024	24	(x,y)	53248	FOOD101	55.62%
LBP	–	6	32	58	(C)	74240		
Root-HOG	–	4	64	31	(C)	79360		

Table 6.19: Results of Feature Combinations with FV encoding by late fusion.

6.2.8 Summary

The best results for each descriptor and each dataset, and the results from both fusion strategies are summarised in Table 6.20. The CH descriptor is not included in the summary, due to the unsatisfying results. The best result from a single descriptor is achieved with the CP descriptor for all datasets. For the UEC-FOOD100BB an accuracy of 54.10% is reached, for the UEC-FOOD256BB 45.38% and the FOOD101 48.38%. The improvement of feature fusion is 2.8%, 6.14% and 2.8% for the three datasets respectively, and the improvements of late fusion (of all descriptors) over the use of the best performing descriptor is 4.23%, 7.76% and 7.24%.

The best results achieved with BoF-encoding are 58.33%, 53.14% and 55.62% for the three datasets. This results are compared to the other object recognition approaches at the end of this chapter.

method	UEC-FOOD100BB	UEC-FOOD256BB	FOOD101
CP (SED)	54.10	45.38	48.38
CMI	29.10	23.71	– [a]
SIFT	43.77	36.63	37.46
LBP	29.40	23.59	22.39
Root-HOG	32.93	23.11	21.77
feature fusion	56.90	51.52	51.18
late fusion	58.33	53.14	55.62

[a] Was cancelled due too long execution time.

Table 6.20: A summary of the best results from all FV-encoding experiments, of each descriptor for both fusion strategies and all three datasets.

6.3 Deep Convolutional Neural Nets

For experiments with DCNNs, two network architectures are selected, the *AlexNet* and the *GoogLeNet*. Various combinations of pre-training and fine-tuning are compared for both architectures. The results are presented in Tables 6.21 and 6.23 in the following sections.

6.3.1 AlexNet

Results from experiments with the AlexNet DCNN architecture (described in Section 4.3.4) are presented in Table 6.21. For the UEC-FOOD100BB dataset the AlexNet achieves an accuracy of 48.63% without any pretraining [a]. Pre-training the network with the 1000 categories from the ILSVRC-2012 results in an improvement of 18.66% [b]. Fine-tuning on the UEC-FOOD100BB dataset achieves an improvement of another 6.99% [e].

For fine-tuning, each layer is applied with a multiplier for the learning rate. A value of 0 for the multiplier would stop the particular layer from further learning, a value of 0.5 would let the particular layer learn with half of the current learning rate, and a value of 1 would apply the full learning rate to the particular layer. In [c]–[f], various learning rate-multiplier configurations for each individual layer have been tested. All configurations are listed with the learning rate-multipliers for each layer in Table 6.22. The best result is achieved by having the last convolutional layer and the last fully connected layer learn at the full learning rate, and each preceding layer learns at half the rate as the its following layer. The configuration is denoted as *all layers exp.*

In [g]–[j], the network from [y] is used as the pre-trained network. This network has been pre-trained with the 1000 ILSVRC-2012 categories and then fine-tuned on the 101000 food images from the FOOD101 dataset. Again the network is trained without

	pretrained	finetuning config.	final training epoch	dataset	accuracy top-1	accuracy top-5
(a)	–	–	90	UEC-100BB	48.63%	72.99%
(b)	ImageNet	–	72	UEC-100BB	67.29%	89.84%
(c)	ImageNet	all layers const.	81	UEC-100BB	72.25%	92.36%
(d)	ImageNet	all layers step	88	UEC-100BB	73.26%	92.78%
(e)	ImageNet	all layers exp.	79	UEC-100BB	74.28%	92.69%
(f)	ImageNet	all layers exp2.	87	UEC-100BB	72.89%	92.22%
(g)	ImageNet & FOOD101 [y]	–	72	UEC-100BB	67.48%	90.67%
(h)	ImageNet & FOOD101 [y]	all layers const.	90	UEC-100BB	72.22%	93.06%
(i)	ImageNet & FOOD101 [y]	all layers step	90	UEC-100BB	73.63%	93.24%
(j)	ImageNet & FOOD101 [y]	all layers exp.	77	UEC-100BB	74.31%	93.61%
(k)	–	–	84	UEC-256BB	38.53%	64.17%
(l)	ImageNet	–	86	UEC-256BB	54.94%	80.73%
(m)	ImageNet	all layers const.	82	UEC-256BB	61.61%	85.11%
(n)	ImageNet	all layers step	60	UEC-256BB	61.79%	85.49%
(o)	ImageNet	all layers exp.	76	UEC-256BB	62.92%	85.97%
(p)	ImageNet & FOOD101 [y]	–	86	UEC-256BB	56.09%	81.59%
(q)	ImageNet & FOOD101 [y]	all layers const.	67	UEC-256BB	62.37%	86.04%
(r)	ImageNet & FOOD101 [y]	all layers step	80	UEC-256BB	61.79%	85.49%
(s)	ImageNet & FOOD101 [y]	all layers exp.	77	UEC-256BB	64.19%	87.35%
(t)	–	–	90	FOOD101	52.36%	77.43%
(u)	ImageNet	–	86	FOOD101	51.43%	78.18%
(v)	ImageNet	all layers const.	82	FOOD101	67.36%	88.26%
(x)	ImageNet	all layers step	85	FOOD101	66.50%	87.34%
(y)	ImageNet	all layers exp.	85	FOOD101	68.58%	88.77%

Table 6.21: Results with various training configurations of the AlexNet. The initial learning rate is set to 0.01 for all experiments and an exponential decay function is applied throughout the training.

fine-tuning first [g], and then fine-tuned on the configurations from Table 6.22. The additional adaptation to the FOOD101 data, does not improve the results compared to the experiments in [b]–[f]. The best result achieved on the UEC-FOOD100BB dataset is 74.31%.

For the UEC-FOOD256BB dataset the same procedure is repeated in [k]–[s]. In the case of this dataset the use of the ILSVRC pre-trained and FOOD101-fine-tuned network [y], increases the accuracy compared to the network only pre-trained with the ILSVRC. The improvement is 1.15%, for the non-fine-tuned, and 1.27% for the best fine-tuning variant. The best result achieved on the UEC-FOOD256BB is 64.19%.

In [t]–[y] the FOOD101 dataset is classified with the AlexNet. First without pre-training reaching 52.36% in [t]. The result is higher then the result achieved by applying the ILSVRC-pre-trained AlexNet, without fine-tuning. The cause is that each category in the FOOD101 dataset has 1000 images, which is the same amount of data used for training the parameters of the ILSVRC net. When the pre-trained network is fine-tuned on the FOOD101 data though, the accuracy increases 16.22%, reaching 68.58% classification accuracy.

conf. name	conv1	conv2	conv3	conv4	conv5	fc6	fc7	fc8
no fine-tuning	0	0	0	0	0	0	0	1
all layers const.	0.05	0.05	0.05	0.05	0.05	0.05	0.05	1
all layers step	0.05	0.05	0.05	0.05	0.5	0.5	0.5	1
all layers exp.	0.05	0.1	0.2	0.5	1	0.2	0.5	1
all layers exp. 2	0.05	0.1	0.2	0.5	1	1	1	1

Table 6.22: Configuration of learning rates of the weights of the individual layers of the AlexNet architecture, that are used in the fine-tuning process to the food data. The value zero means, no that the corresponding layer does not continue learning the parameters of the pre-trained model, therefore the layer is not fine-tuned. The learning rates of the biases are consistently set to the double value of the weight learning rates.

6.3.2 GoogLeNet

Results from experiments with the GoogLeNet architecture (described in Section 4.3.5) are presented in Table 6.23. For the UEC-FOOD100BB dataset the GoogLeNet achieves an accuracy of 53.97% without any pretraining [a]. Pre-training the network with the 1000 categories from the ILSVRC-2012 results in an improvement of 17.22% [b]. Fine-tuning on the UEC-FOOD100BB dataset achieves an improvement of another 7.95% [e].

Similar as with the AlexNet, for fine-tuning each layer is applied with a multiplier for the learning rate. Due to the high number of layers only constant multipliers for all layers(except the output layer, on which the unchanged learning rate is applied on) are used. The values 0.01, 0.05, 0.1 and 0.2 are tested. The rate of 0.2 achieves the best result in all experiments with the exception of experiments [f], where a slightly better result is achieved in [e].

In [g]–[j], the network from [z] is used as the pre-trained network. This network has been pre-trained with the 1000 ILSVRC-2012 categories and is then fine-tuned on the 101000 food images from the FOOD101 dataset. The network is trained without fine-tuning first [g], and then fine-tuned on the learning rate multipliers of 0.05, 0.1 and 0.2 for all layers, except the output layer. The additional pre-training on the FOOD101 data, does improve the result from [e] to 80.34% [j], an increase of 1.2%.

	pretrained	finetuning config.	final training epoch	dataset	accuracy top-1	accuracy top-5
(a)	–	–	100	UEC-100BB	53.97%	80.00%
(b)	ImageNet	–	97	UEC-100BB	71.19%	91.82%
(c)	ImageNet	all layers const. 0.01	77	UEC-100BB	76.14%	94.59%
(d)	ImageNet	all layers const. 0.05	93	UEC-100BB	78.65%	95.33%
(e)	ImageNet	all layers const. 0.1	87	UEC-100BB	79.14%	95.66%
(f)	ImageNet	all layers const. 0.2	100	UEC-100BB	79.07%	95.86%
(g)	FOOD101 [z]	–	33	UEC-100BB	71.31%	92.19%
(h)	FOOD101 [z]	all layers const. 0.05	88	UEC-100BB	78.02%	95.89%
(i)	FOOD101 [z]	all layers const. 0.1	59	UEC-100BB	79.30%	96.14%
(j)	FOOD101 [z]	all layers const. 0.2	55	UEC-100BB	80.34%	96.42%
(k)	–	–	100	UEC-256BB	49.33%	76.44%
(l)	ImageNet	–	99	UEC-256BB	58.07%	83.47%
(m)	ImageNet	all layers const. 0.01	89	UEC-256BB	65.94%	88.22%
(n)	ImageNet	all layers const. 0.05	92	UEC-256BB	68.48%	89.78%
(o)	ImageNet	all layers const. 0.1	83	UEC-256BB	69.29%	90.68%
(p)	ImageNet	all layers const. 0.2	70	UEC-256BB	69.93%	90.03%
(q)	FOOD101 [z]	–	85	UEC-256BB	61.11%	85.27%
(r)	FOOD101 [z]	all layers const. 0.05	74	UEC-256BB	69.82%	90.67%
(s)	FOOD101 [z]	all layers const. 0.1	64	UEC-256BB	70.04%	90.90%
(t)	FOOD101 [z]	all layers const. 0.2	83	UEC-256BB	71.10%	91.40%
(u)	–	–	99	FOOD101	68.96%	89.02%
(v)	ImageNet	–	100	FOOD101	58.73%	83.66%
(w)	ImageNet	all layers const. 0.01	96	FOOD101	75.54%	93.37%
(x)	ImageNet	all layers const. 0.05	85	FOOD101	78.64%	94.36%
(y)	ImageNet	all layers const. 0.1	97	FOOD101	78.97%	94.19%
(z)	ImageNet	all layers const. 0.2	73	FOOD101	79.39%	94.27%

Table 6.23: Results with various training configurations of the GoogLeNet. The initial learning rate is set to 0.01 for all experiments and an exponential decay function is applied throughout the training.

For the UEC-FOOD256BB dataset the same procedure is repeated in [k]–[t]. The accuracy of the network without any pre-training is 49.33% [k]. An increase of 8.74% is achieved by using the ILSVRC pre-trained GoogLeNet [l]. Another 11.86% increase in accuracy is achieved by fine-tuning this network to the UEC-FOOD256BB [p]. The final additional pre-training on the FOOD101 data improves the result to 71.10% [t], a further increase of 1.17% ([t] v.s [p]).

In [u]–[z] the FOOD101 dataset is classified with the GoogLeNet. First without pre-

training reaching 68.96% in [u]. Same as for the AlexNet, the result is higher then the result achieved by applying the ILSVRC-pre-trained GoogLeNet, without fine-tuning. The cause is that each category in the FOOD101 dataset has 1000 images, which is the same amount of data used for training the parameters of the ILSVRC net. When the pre-trained network is fine-tuned on the FOOD101 data though, the accuracy increases 10.43% ([u] vs. [z]), reaching 79.39% classification accuracy.

6.4 Comparisons

Table 6.24 shows a summary of the best results of all methods. For the BoF and the FV encoding techniques, the best result of a single descriptor and the best result for each descriptor combination is selected. Also the best results of each DCNN-network architecture for learning without pre-training and with pre-training are selected. In Figure 6.1 the best results of each method and for each dataset are visualised in direct comparison.

method	details	UEC-100BB	UEC-256BB	FOOD101
BoF single descriptor	SIFT	43.97%	33.50%	35.74%
BoF descriptor fusion	CP & SIFT	50.87%	42.20%	42.39%
BoF late fusion	all descriptors	52.30%	44.32%	44.16%
FV single descriptor	CP (SED)	54.10%	45.38%	48.38%
FV descriptor fusion	CP & SIFT	56.90%	51.52%	51.18%[a]
FV late fusion	all descriptors	58.33%	53.14%	55.62%
DCNN AlexNet	no pre-training	48.63%	38.53%	52.36%
DCNN GoogLeNet	no pre-training	53.97%	49.33%	68.96%
DCNN AlexNet	pretrained & finetuned	74.31%	64.19%	68.58%
DCNN GoogLeNet	pretrained & finetuned	80.34%	71.10%	79.39%

[a] SED method is used for both descriptors.

Table 6.24: Summary of best results of all methods for all three datasets.

The improvement of FV over BoF is 6.03%, 8.82% and 11.46% and the improvement of DCNN over FV is 22.01%, 17.96% and 23.77%, for the UEC-FOOD100BB, UEC-FOOD256BB and the FOOD101 datasets respectively. Compared to the DCNNs without any pre-training, all results of the FV encoding are better than the DCNN results for all datasets, except for the GoogLeNet classification of the FOOD101 dataset. Showing that for small scale datasets the FV encoding can outperform DCNN, but for larger scale datasets (1000 images and more per category) DCNN will outperform FV encoding.

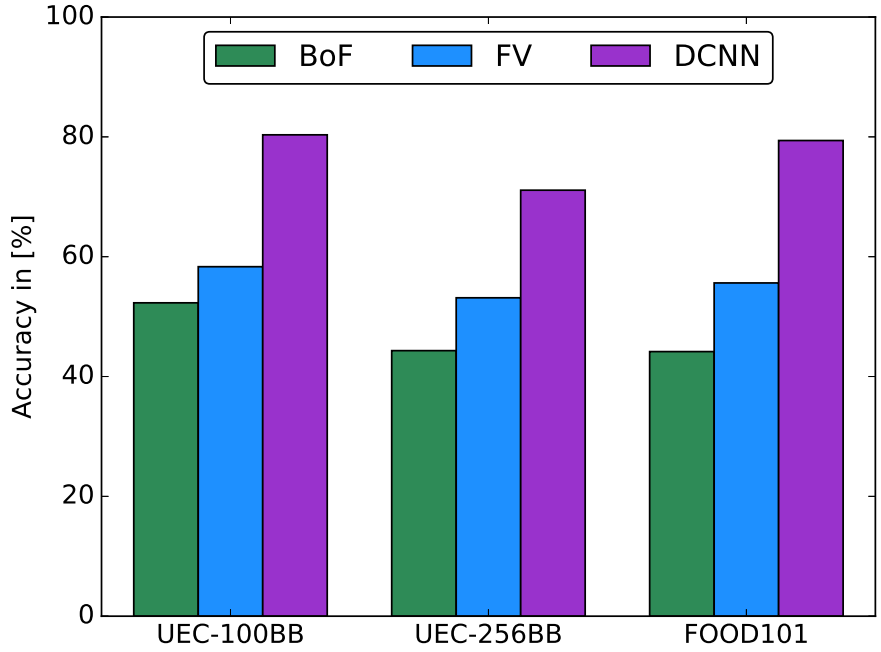


Figure 6.1: Top results of the three recognition strategies for all three datasets.

An interesting finding is the similar recognition rate for the UEC-FOOD100BB and the FOOD101 datasets, 80.34% and 79.39%. This results show two capabilities of the GoogLeNet DCNN architecture: the invariance of the computed features to background information if there is enough data, and the ability of adapting to datasets of small scale through fine-tuning.

CHAPTER 7

Conclusion

An analysis of state-of-the-art food recognition approaches identify following methods as the most promising:

- Bag-of-Features (BoF)-encoding of texture and colour features
- Fisher-Vector (FV)-encoding of texture and colour features
- Deep Convolutional Neural Network (DCNN)

Experiments with the three selected object recognition techniques on three of the biggest publicly available datasets for food images are in the expected ranges of accuracy (compared to previous observations from related work): the FV-encoding technique outperforms the BoF encoding and the DCNN technique outperforms the FV encoding in the classification task.

Analysing the results in the context of dietary assessment, a fully automated recognition can not be advised for a satisfying performance with the current methods. Considering the best result of around 80% recognition accuracy in a 100-category problem with real-world data that does not underlie any assumptions, and around 70% for 256-categories with images from bounding-box segmented objects containing less background information. The results achieved on the datasets are supported by experiments with similar results by [Myers et al., 2015, Yanai and Kawano, 2015]. A real-world system could include many hundreds or even thousands of fine-graded food categories. Depending on the application, the identification of the dish is followed by further modelling and/or assumptions of the data, e.g. for segmentation or volume estimation.

Computer vision does not provide a one-fits-all solution, but has the potential to contribute to an improvement of the dietary assessment process, increase its usability and lower

its costs [Sharp and Allman-Farinelli, 2014]. To improve traditional dietary assessment methods, computer vision techniques can play an assisting role in the data collection. The results for the top-5 recognition rate were accuracies in the range of 90-96% for all three datasets (GoogLeNet-DCNN), which could be used for interactive suggestions saving the user time, compared to text-searching for items. For identifying information that is not included on the imagery data, such as exact preparation details, brands of ingredients, used oils and so on, combinations with traditional assessment techniques such as the 24HR and DR methods could achieve improvements.

Limiting the application to sub-problems, achieve promising results. [Rhyner et al., 2016] conducted a comparative study with self-reporting results, [Lee et al., 2012] and [Anthimopoulos et al., 2015] conducted comparative studies with weighted ground truth values. All three applications were conducted in a laboratory setting of limited categories and assumptions on the data, such as images without any occlusions of food ingredients. Natural limitations on basis of image data, such as occlusion, limits the informational content of the assessment. Such limitations are inherently relevant for computational vision analysis systems. Improvements that can be achieved over traditional dietary assessment methods on basis of imagery data, could be investigated in further research in form of comparative evaluations with weighted ground truth information of the contained nutrients. Also comparative studies of computer vision systems with self-reportings of users and/or estimations of dietitian experts are of interest, considering the shortcomings of traditional dietary assessment methods.

Acronyms

24HR 24-Hour Recall. 2, 8, 9, 11, 12, 14, 124

ANN Artificial Neural Network. 23

BAIR Berkeley AI Research. 86

BoF Bag-of-Features. 5, 19–21, 23, 26, 27, 30, 35, 36, 43, 48, 49, 51–58, 62, 72, 73, 76, 80, 83, 84, 89–92, 94, 96, 98, 99, 101–103, 106, 116, 121, 123

CAFFE Convolutional Architecture For Fast Feature Embedding. 25, 27, 80, 86

CH Colour Histogram. 92, 102, 103, 106, 116

CL Convolutional Layer. 66, 67, 69

CMI Colour Moment Invariant. 41, 43, 48, 51, 52, 94, 102, 103, 106, 107, 109, 116, 117

CNN Convolutional Neural Network. 24, 66, 67, 80

CP Colourpatch. 90, 101–103, 115–117, 121

CPU Central Processing Unit. 27

CRF Conditional Random Field. 29

DCNN Deep Convolutional Neural Network. 5, 24, 25, 27–30, 35, 36, 39, 64, 65, 67, 68, 71–73, 76, 80, 82, 86, 89, 117, 121, 123, 124

DH Dietary History. 9, 14

DoG Difference of Gaussians. 49, 50

DR Dietary Record. 8–10, 14, 15, 124

EFD Fractal Dimension estimation. 21, 45

FFNN Feedforward Neural Networks. 64, 65

FFQ Food Frequency Questionnaires. 2, 9, 11, 14, 34, 35

FK Fisher Kernel. 58, 59

FNDDS Food and Nutrient Database for Dietary Studies. 33

FV Fisher Vector. 5, 27, 28, 35, 36, 42, 43, 47–49, 51–53, 58–63, 71–73, 76, 80, 83, 85, 89, 103, 106, 107, 109, 111, 113–117, 121, 123

GCM Generalised Colour Moment. 41

GFD Gabor-based image decomposition and Fractal Dimension estimation. 21, 45

GMM Gaussian Mixture Model. 27, 58, 61, 62, 72, 83, 103, 104, 107, 110, 111, 113

GOSDM Gradient Orientation Spatial-Dependence Matrix. 21, 45

GPU Graphics Processing Unit. 67, 69, 86

HOG Histogram of Oriented Gradients. 26, 27, 30, 39, 42, 47, 48, 51, 52, 68, 71, 80, 99, 101–103, 113, 114, 116, 117

IDE Integrated Development Environment. 80

ILSVRC ImageNet Large-Scale Visual Recognition Challenge. 5, 27, 29, 64, 67–72, 86, 117–121

IM Inception Module. 69, 70

JPEG Joint Photographic Experts Group. 86

KNN K-Nearest-Neighbour. 21

LBP Local Binary Pattern. 21, 23, 25, 30, 39, 44–46, 48, 52, 80, 90, 98, 99, 102, 103, 111, 113, 116, 117

LCS Local Colour Statistics. 61, 63, 64, 72

LLC Locality-constrained Linear Coding. 30, 35, 36, 52, 53, 58, 72

LMDB Lightning Memory-Mapped Database. 86

mAP mean Average Precision. 29, 60, 68, 72

MEX Matlab Executable. 82

MKL Multiple Kernel Learning. 26

MLP Multilayer Perceptron. 64, 65

mpFR mobile phone Food Record. 19–21, 36

NIN Network In Network. 28

NN Neural Network. 64–67

NNDB National Nutrient Database. 32, 33

OpenCV Open Computer Vision. 80

ORA Overall Recognition Accuracy. 23

PCA Principal Component Analysis. 27, 52, 60–62, 83, 110

PL Pooling Layer. 67

PN Power Normalisation. 62–64

RANSAC Random Sample Consensus. 24

RBF Radial Basis Function. 20, 23

ReLU Rectified Linear Unit. 65, 69

RF Random Forest. 23

SED Spatially Extended Descriptor. 62, 72, 83, 104, 105, 109–115, 117, 121

SGD Stochastic Gradient Descent. 66, 68

SIFT Scale Invariant Feature Transform. 20, 21, 23, 26, 30, 39, 41, 47–52, 60, 61, 63, 64, 68, 72, 80, 81, 96, 98, 101–103, 106, 109, 111, 115–117, 121

SN Sigmoid Neuron. 65

SP Spatial Pyramid. 56, 62, 64, 76, 83, 89–91, 93, 95, 96, 98, 99, 103, 107, 110, 111, 113

SPM Spatial Pyramid Matching. 56, 57, 62, 72, 83, 89, 93, 104, 105, 108–115

SRG Seeded Region Growing. 24

SURF Speeded-Up Robust Feature. 20, 27, 48, 52

SVM Support Vector Machine. 20, 21, 23, 25–27, 30, 48, 51, 52, 57, 59, 76, 80, 82, 83, 90, 96, 103, 106

TADA Technology Assisted Dietary Assessment. 19

USDA United States Department of Agriculture. 32

VLAD Vector of Locally Aggregated Descriptors. 53

VLFeat Visual Lab Features. 80, 82, 83, 96, 106

Bibliography

[Anthimopoulos et al., 2013] Anthimopoulos, M., Dehais, J., Diem, P., and Mougiakakou, S. (2013). Segmentation and recognition of multi-food meal images for carbohydrate counting. In *2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 1–4.

[Anthimopoulos et al., 2015] Anthimopoulos, M., Dehais, J., Shevchik, S., Ransford, B. H., Duke, D., Diem, P., and Mougiakakou, S. (2015). Computer vision-based carbohydrate estimation for type 1 patients with diabetes using smartphones. *Journal of Diabetes Science and Technology*, 9(3):507–515.

[Anthimopoulos et al., 2014] Anthimopoulos, M., Gianola, L., Scarnato, L., Diem, P., and Mougiakakou, S. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE Journal of Biomedical and Health Informatics*, 18(4):1261–1271.

[Arandjelović and Zisserman, 2012] Arandjelović, R. and Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918.

[Arbelaez et al., 2011] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916.

[Baranowski et al., 2014] Baranowski, T., Islam, N., Douglass, D., Dadabhoy, H., Beltran, A., Baranowski, J., Thompson, D., Cullen, K. W., and Subar, A. F. (2014). Food intake recording software system, version 4 (firsst4): A self-completed 24 hour dietary recall for children. *Journal of Human Nutrition and Dietetics: The Official Journal of the British Dietetic Association*, 27(0 1):66–71. 22616645[pmid].

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

[Beijbom et al., 2015] Beijbom, O., Joshi, N., Morris, D., Saponas, S., and Khullar, S. (2015). Menu-match: Restaurant-specific food logging from images. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 844–851.

[Bettadapura et al., 2015] Bettadapura, V., Thomaz, E., Parnami, A., Abowd, G. D., and Essa, I. A. (2015). Leveraging Context to Support Automated Food Recognition in Restaurants. *Computing Research Repository*, abs/1510.02078.

[Borji and Itti, 2014] Borji, A. and Itti, L. (2014). Human vs. computer in scene and object recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–120.

[Bosch et al., 2011a] Bosch, M., Schap, T., Zhu, F., Khanna, N., Boushey, C. J., and Delp, E. J. (2011a). Integrated database system for mobile dietary assessment and analysis. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6.

[Bosch et al., 2011b] Bosch, M., Zhu, F., Khanna, N., Boushey, C., and Delp, E. (2011b). Combining global and local features for food identification in dietary assessment. In *2011 18th IEEE International Conference on Image Processing (ICIP)*, pages 1789–1792.

[Bosch et al., 2011c] Bosch, M., Zhu, F., Khanna, N., Boushey, C. J., and Delp, E. J. (2011c). Food texture descriptors based on fractal and local gradient information. In *Proceedings of the 19th European Signal Processing Conference, EUSIPCO 2011, Barcelona, Spain, August 29 - Sept. 2, 2011*, pages 764–768.

[Bossard et al., 2014] Bossard, L., Guillaumin, M., and Van Gool, L. (2014). Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*.

[Boushey et al., 2015] Boushey, J. C., Harray, J. A., Kerr, A. D., Schap, E. T., Paterson, S., Aflague, T., Bosch Ruiz, M., Ahmad, Z., and Delp, J. E. (2015). How willing are adolescents to record their dietary intake? the mobile food record. *Journal of Medical Internet Research*, 3(2):e47.

[Branson et al., 2010] Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., and Belongie, S. (2010). *Visual Recognition with Humans in the Loop*, pages 438–451. Springer Berlin Heidelberg.

[Burke, 1947] Burke, B. (1947). The diet history as a tool in research. *Journal of the American Dietetic Association*, 23:1041–1046.

[Burrows et al., 2010] Burrows, T. L., Martin, R. J., and Collins, C. E. (2010). A systematic review of the validity of dietary assessment methods in children when compared with the method of doubly labeled water. *Journal of the American Dietetic Association*, 110(10):1501 – 1510.

[Carter et al., 2012] Carter, M. C., Burley, V. J., Nykjaer, C., and Cade, J. E. (2012). 'My Meal Mate' (MMM): validation of the diet measures captured on a smartphone application to facilitate weight loss. *British Journal of Nutrition*, 109(3):539–546.

[Chatfield et al., 2011] Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*.

[Chatfield et al., 2014] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*.

[Chen et al., 2014] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *Computing Research Repository*, abs/1412.7062.

[Chen et al., 2009] Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R., and Yang, J. (2009). PFID: Pittsburgh fast-food image dataset. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 289–292.

[Chen et al., 2012] Chen, M.-Y., Yang, Y.-H., Ho, C.-J., Wang, S.-H., Liu, S.-M., Chang, E., Yeh, C.-H., and Ouhyoung, M. (2012). Automatic chinese food identification and quantity estimation. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, pages 29:1–29:4, New York, NY, USA. ACM.

[Christodoulidis et al., 2015] Christodoulidis, S., Anthimopoulos, M., and Mougiakakou, S. (2015). *Food Recognition for Dietary Assessment Using Deep Convolutional Neural Networks*, pages 458–465. Springer International Publishing.

[Ciocca et al., 2017] Ciocca, G., Napoletano, P., and Schettini, R. (2017). Food recognition: A new dataset, experiments, and results. *IEEE Journal of Biomedical and Health Informatics*, 21(3):588–598.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA. IEEE Computer Society.

[Dehais et al., 2015] Dehais, J., Anthimopoulos, M., and Mougiakakou, S. (2015). *Dish Detection and Segmentation for Dietary Assessment on Smartphones*, pages 433–440. Springer International Publishing.

[Dehais et al., 2013] Dehais, J., Shevchik, S., Diem, P., and Mougiakakou, S. G. (2013). Food volume computation for self dietary assessment applications. In *2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 1–4.

[Donahue et al., 2013] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013). DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *Computing Research Repository*, abs/1310.1531.

[Eigen and Fergus, 2014] Eigen, D. and Fergus, R. (2014). Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. *Computing Research Repository*, abs/1411.4734.

[Elkan, 2003] Elkan, C. (2003). Using the Triangle Inequality to Accelerate k-Means. In Fawcett, T. and Mishra, N., editors, *International Conference on Machine Learning*, pages 147–153. AAAI Press.

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645.

[Felzenszwalb and Huttenlocher, 1998] Felzenszwalb, P. F. and Huttenlocher, D. P. (1998). Image segmentation using local variation. In *1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998. Proceedings.*, pages 98–104.

[Forouhi and Wareham, 2014] Forouhi, N. G. and Wareham, N. J. (2014). Epidemiology of diabetes. *Medicine (Abingdon)*, 42(12):698–702. S1357-3039(14)00271-0[PII].

[Forster et al., 2016] Forster, H., Walsh, M. C., Gibney, M. J., Brennan, L., and Gibney, E. R. (2016). Personalised nutrition: the role of new dietary assessment methods. *Proceedings of the Nutrition Society*, 75(1):96–105.

[Freeman and Adelson, 1991] Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906.

[Gemming et al., 2015] Gemming, L., Utter, J., and Mhurchu, C. N. (2015). Image-assisted dietary assessment: A systematic review of the evidence. *Journal of the Academy of Nutrition and Dietetics*, 115(1):64 – 77.

[Girshick et al., 2013] Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *Computing Research Repository*, abs/1311.2524.

[GoCARB Project, 2016] GoCARB Project (2016). Type 1 diabetes self-management and carbohydrate counting: A computer vision based approach. URL: `http://gocarb.eu/gocarb-project/`, [accessed 2016-10-03].

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Graff et al., 2000] Graff, M. R., Gross, T. M., Juth, S. E., and Charlson, J. (2000). How well are individuals on intensive insulin therapy counting carbohydrates? *Diabetes Research and Clinical Practice*, 50:238 – 239.

[Grauman and Darrell, 2005] Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision*, pages 1458–1465.

[Griffin et al., 2007] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology.

[Haralick et al., 1973] Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6).

[Hoashi et al., 2010] Hoashi, H., Joutou, T., and Yanai, K. (2010). Image recognition of 85 food categories by feature fusion. In *Proceedings of the 2010 IEEE International Symposium on Multimedia*, ISM '10, pages 296–301, Washington, DC, USA. IEEE Computer Society.

[Illner et al., 2012] Illner, A.-K., Freisling, H., Boeing, H., Huybrechts, I., Crispim, S., and Slimani, N. (2012). Review and evaluation of innovative technologies for measuring diet in nutritional epidemiology. *International Journal of Epidemiology*, 41(4):1187–1203.

[International Diabetes Federation, 2016] International Diabetes Federation (2016). IDF Diabetes Atlas, 7th Edition. URL: `http://www.diabetesatlas.org/`, [accessed 2016-09-15].

[Jaakkola and Haussler, 1998] Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press.

[Jegou et al., 2012] Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., and Schmid, C. (2012). Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(9):1704–1716.

[Jia et al., 2014] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA. ACM.

[Joachims, 2006] Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 217–226, New York, NY, USA. ACM.

[Kagaya et al., 2014] Kagaya, H., Aizawa, K., and Ogawa, M. (2014). Food detection and recognition using convolutional neural network. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 1085–1088, New York, NY, USA. ACM.

[Karl and Roberts, 2014] Karl, J. P. and Roberts, S. B. (2014). Energy density, energy intake, and body weight regulation in adults. *Advances in Nutrition*, 5(6):835–850. 007112[PII].

[Kawano and Yanai, 2014] Kawano, Y. and Yanai, K. (2014). Food image recognition using deep convolutional features pre-trained with food-related categories. *Department of Informatics, The University of Electro-Communications 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 JAPAN.*

[Kawano and Yanai, 2015a] Kawano, Y. and Yanai, K. (2015a). Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In *Computer Vision - ECCV 2014 Workshops*, volume 8927 of *Lecture Notes in Computer Science*, pages 3–17. Springer International Publishing.

[Kawano and Yanai, 2015b] Kawano, Y. and Yanai, K. (2015b). Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, 74(14):5263–5287.

[Kikunaga et al., 2007] Kikunaga, S., Tin, T., Ishibashi, G., Wang, D.-H., and Kira, S. (2007). The Application of a Handheld Personal Digital Assistant with Camera and Mobile Phone Card (Wellnavi) to the General Population in a Dietary Survey. *Journal of Nutritional Science and Vitaminology*, 53(2):109–116.

[Knez and Šajn, 2015] Knez, S. and Šajn, L. (2015). *Food Object Recognition Using a Mobile Device: State of the Art*, pages 366–374. Springer International Publishing, Cham.

[Krizhevsky, 2009] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA. IEEE Computer Society.

[LeCun et al., 2004] LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 97–104, Washington, DC, USA. IEEE Computer Society.

[Lee et al., 2012] Lee, C. D., Chae, J., Schap, T. E., Kerr, D. A., Delp, E. J., Ebert, D. S., and Boushey, C. J. (2012). Comparison of known food weights with image-based

portion-size automated estimation and adolescents' self-reported portion size. *Journal of Diabetes Science and Technology*, 6(2):428–434. 22538157[pmid].

[Li et al., 2010] Li, Z., Liu, G., Qian, X., and Wang, C. (2010). Scale and rotation invariant gabor texture descriptor for texture classification.

[Lin et al., 2013] Lin, M., Chen, Q., and Yan, S. (2013). Network In Network. *Computing Research Repository*, abs/1312.4400.

[Lloyd, 1982] Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–136.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Mangai et al., 2010] Mangai, U. G., Samanta, S., Das, S., and Chowdhury, P. R. (2010). A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical Review*, 27:293–307.

[Martin et al., 2012] Martin, C. K., Correa, J. B., Han, H., Allen, H. R., Rood, J., Champagne, C. M., Gunturk, B. K., and Bray, G. A. (2012). Validity of the Remote Food Photography Method (RFPM) for estimating energy and nutrient intake in near real-time. *Obesity (Silver Spring)*, 20(4):891–899. 22134199[pmid].

[Martin et al., 2009] Martin, C. K., Han, H., Coulon, S. M., Allen, H. R., Champagne, C. M., and Anton, S. D. (2009). A novel method to remotely measure food intake of free-living people in real-time: The Remote Food Photography Method (RFPM). *The British journal of nutrition*, 101(3):446–456. 18616837[pmid].

[Matas et al., 2002] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press. doi:10.5244/C.16.36.

[McAllister et al., 2009] McAllister, E. J., Dhurandhar, N. V., Keith, S. W., Aronne, L. J., Barger, J., Baskin, M., Benca, R. M., Biggio, J., Boggiano, M. M., Eisenmann, J. C., Elobeid, M., Fontaine, K. R., Gluckman, P., Hanlon, E. C., Katzmarzyk, P., Pietrobelli, A., Redden, D. T., Ruden, D. M., Wang, C., Waterland, R. A., Wright, S. M., and Allison, D. B. (2009). Ten putative contributors to the obesity epidemic. *Critical Reviews in Food Science and Nutrition*, 49(10):868–913. 19960394[pmid].

[Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[Mindru et al., 2004] Mindru, F., Tuytelaars, T., Van Gool, L., and Moons, T. (2004). Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, 94(1-3):3–27.

[Myers et al., 2015] Myers, A., Johnston, N., Rathod, V., Balan, A. K., Gorban, A. N., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J., and Murphy, K. (2015). Im2calories: Towards an automated mobile vision food diary. In *International Conference on Computer Vision*.

[Nabi et al., 2015] Nabi, J., Doddamadaiah, A. R., and Lakhotia, R. (2015). Smart dietary monitoring system. In *2015 IEEE International Symposium on Nanoelectronic and Information Systems*, pages 207–212.

[Nielsen, 2015] Nielsen, M. (2015). Neural networks and deep learning.

[Noronha et al., 2011] Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. (2011). Platemate: Crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 1–12, New York, NY, USA. ACM.

[O'Hara and Draper, 2011] O'Hara, S. and Draper, B. A. (2011). Introduction to the bag of features paradigm for image classification and retrieval. *Computing Research Repository*, abs/1101.3354.

[Ojala et al., 2002] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

[Oquab et al., 2014] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1717–1724, Washington, DC, USA. IEEE Computer Society.

[Perronnin and Dance, 2007] Perronnin, F. and Dance, C. R. (2007). Fisher kernels on visual vocabularies for image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society.

[Perronnin et al., 2010] Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg. Springer-Verlag.

[Pouladzadeh et al., 2013] Pouladzadeh, P., Shirmohammadi, S., and Arici, T. (2013). Intelligent SVM based food intake measurement system. In *2013 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 87–92.

[Puri et al., 2009] Puri, M., Zhu, Z., Yu, Q., Divakaran, A., and Sawhney, H. (2009). Recognition and volume estimation of food intake using a mobile device. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–8.

[Rahman et al., 2011] Rahman, M. H., Pickering, M. R., and Frater, M. R. (2011). Scale and rotation invariant gabor features for texture retrieval. In *2011 International Conference on Digital Image Computing: Techniques and Applications*, pages 602–607.

[Rhyner et al., 2016] Rhyner, D., Loher, H., Dehais, J., Anthimopoulos, M., Shevchik, S., Botwey, R. H., Duke, D., Stettler, C., Diem, P., and Mougiakakou, S. (2016). Carbohydrate estimation by a mobile phone-based system versus self-estimations of individuals with type 1 diabetes mellitus: A comparative study. *Journal of Medical Internet Research*, 18(5):e101. v18i5e101[PII].

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

[Rother et al., 2004] Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM Special Interest Group on Graphics and Interactive Techniques 2004 Papers*, SIGGRAPH '04, pages 309–314, New York, NY, USA. ACM.

[Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Sánchez et al., 2012] Sánchez, J., Perronnin, F., and De Campos, T. (2012). Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*, 33(16):2216–2223.

[Sanchez et al., 2013] Sanchez, J., Perronnin, F., Mensink, T. E. J., and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*.

[Shannon, 1948] Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423.

[Sharp and Allman-Farinelli, 2014] Sharp, D. B. and Allman-Farinelli, M. (2014). Feasibility and validity of mobile phones to assess dietary intake. *Nutrition*, 30(11–12):1257 – 1266.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905.

[Shim et al., 2014] Shim, J.-S., Oh, K., and Kim, H. C. (2014). Dietary assessment methods in epidemiologic studies. *Epidemiology and Health*, 36:e2014009. epih-36-e2014009[PII].

[Shin et al., 2014] Shin, S., Park, E., Sun, D. H., You, T.-K., Lee, M.-J., Hwang, S., Paik, H. Y., and Joung, H. (2014). Development and evaluation of a web-based computer-assisted personal interview system (capis) for open-ended dietary assessments among koreans. *Clinical Nutrition Research*, 3(2):115–125. 25136539[pmid].

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computing Research Repository*, abs/1409.1556.

[Small et al., 2009] Small, L., Sidora-Arcoleo, K., Vaughan, L., Creed-Capsel, J., Chung, K.-Y., and Stevens, C. (2009). Validity and reliability of photographic diet diaries for assessing dietary intake among young children. *ICAN: Infant, Child, & Adolescent Nutrition*, 1(1):27–36.

[Smart et al., 2013] Smart, C. E., Evans, M., O'Connell, S. M., McElduff, P., Lopez, P. E., Jones, T. W., Davis, E. A., and King, B. R. (2013). Both dietary protein and fat increase postprandial glucose excursions in children with type 1 diabetes, and the effect is additive. *Diabetes Care*, 36(12):3897–3902.

[Smart et al., 2012] Smart, C. E., King, B. R., McElduff, P., and Collins, C. E. (2012). In children using intensive insulin therapy, a 20-g variation in carbohydrate amount significantly impacts on postprandial glycaemia. *Diabetic Medicine, a Journal of the British Diabetic Association*, 29(7):e21–24.

[Sonnenburg et al., 2006] Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research (JMLR)*, 7:1531–1565.

[Stumbo, 2013] Stumbo, P. J. (2013). New technology in dietary assessment: a review of digital methods in improving food record accuracy. *Proceedings of the Nutrition Society*, 72(1):70–76.

[Subar et al., 2003] Subar, A. F., Kipnis, V., Troiano, R. P., Midthune, D., Schoeller, D. A., Bingham, S., Sharbaugh, C. O., Trabulsi, J., Runswick, S., Ballard-Barbash, R., Sunshine, J., and Schatzkin, A. (2003). Using intake biomarkers to evaluate the extent of dietary misreporting in a large sample of adults: The open study. *American Journal of Epidemiology*, 158(1):1–13.

[Sun et al., 2010] Sun, M., Fernstrom, J. D., Jia, W., Hackworth, S. A., Yao, N., Li, Y., Li, C., Fernstrom, M. H., and Sclabassi, R. J. (2010). A wearable electronic system for objective dietary assessment. *Journal of the American Dietetic Association*, 110(1):45. 20102825[pmid].

[Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. *Computing Research Repository*, abs/1409.4842.

[Tamura et al., 1978] Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473.

[Titterington et al., 1985] Titterington, D., Smith, A., and Makov, U. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York.

[Tola et al., 2010] Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830.

[United States Department of Agriculture, 2016] United States Department of Agriculture (2016). USDA Food Composition Databases. URL: `https://ndb.nal.usda.gov/`, [accessed 2016-11-20].

[USDA, 2016] USDA (2016). United States Department of Agriculture (USDA), National Nutrient Database (NNDB), Standard Release 27. URL: `http://www.ars.usda.gov/ba/bhnrc/ndl`, [accessed 2016-11-20].

[Varma and Zisserman, 2005] Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1):61–81.

[Vedaldi and Fulkerson, 2010] Vedaldi, A. and Fulkerson, B. (2010). Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1469–1472, New York, NY, USA. ACM.

[Vedaldi and Zisserman, 2010] Vedaldi, A. and Zisserman, A. (2010). Efficient additive kernels via explicit feature maps. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3546.

[VLFeat Library, 2016] VLFeat Library (2016). VLFeat Reference Documentation. URL: `http://www.vlfeat.org/doc.html`, [accessed 2016-10-03].

[Wu et al., 2016] Wu, H., Merler, M., Uceda-Sosa, R., and Smith, J. R. (2016). Learning to make better mistakes: Semantics-aware visual food recognition. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pages 172–176, New York, NY, USA. ACM.

[Yanai and Kawano, 2015] Yanai, K. and Kawano, Y. (2015). Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops 2015, Turin, Italy, June 29 - July 3, 2015*, pages 1–6.

[Yanai et al., 2016] Yanai, K., Tanno, R., and Okamoto, K. (2016). Efficient mobile implementation of a cnn-based object recognition system. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pages 362–366, New York, NY, USA. ACM.

[Yang et al., 2009] Yang, J., Yu, K., Gong, Y., and Huang, T. S. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1794–1801.

[Yang et al., 2010] Yang, S., Chen, M., Pomerleau, D., and Sukthankar, R. (2010). Food recognition using statistics of pairwise local features. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2249–2256.

[Yangqing, J., 2013] Yangqing, J. (2013). Caffe: An open source convolutional architecture for fast feature embedding. URL: `http://caffe.berkeleyvision.org`, [accessed 2016-10-03].

[Zhu et al., 2015] Zhu, F., Bosch, M., Khanna, N., Boushey, C., and Delp, E. (2015). Multiple hypotheses image segmentation and classification with application to dietary assessment. *IEEE Journal of Biomedical and Health Informatics*, 19(1):377–388.

[Zhu et al., 2011] Zhu, F., Bosch, M., Schap, T., Khanna, N., Ebert, D. S., Boushey, C. J., and Delp, E. J. (2011). Segmentation assisted food classification for dietary assessment.

[Zhu et al., 2010] Zhu, F., Bosch, M., Woo, I., Kim, S., Boushey, C., Ebert, D., and Delp, E. (2010). The use of mobile devices in aiding dietary assessment and evaluation. *IEEE Journal of Selected Topics in Signal Processing*, 4(4):756–766.

[Zou et al., 2009] Zou, H., Zhu, J., Rosset, S., and Hastie, T. (2009). Multi-class adaboost. *Statistics and its Interface*, 2:349–360.