LEARNING BASED IMAGE ANALYSIS WITH APPLICATION IN DIETARY ASSESSMENT AND EVALUATION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Yu Wang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2017

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Edward J Delp, Co-chair

School of Electrical and Computer Engineering

Dr. Fengqing Zhu, Co-chair

School of Electrical and Computer Engineering

Dr. Jan P Allebach

School of Electrical and Computer Engineering

Dr. Zygmunt Pizlo

Department of Psychological Sciences

Approved by:

Dr. Venkataramanan Balakrishnan

Head of the School Graduate Program

To mom and dad
who always believe in me
encourage me to go on every adventure.

ACKNOWLEDGMENTS

Words cannot express how much I appreciate the guidance, sticks or carrots, from my doctoral adviser, Professor Edward J. Delp. He is not only the definition of a great engineer but also a talented manager. He has always motivated me to challenge myself to the extent that “my head will not explode” and picked me up whenever I feel stressed with the research. He taught me how to become a servant leader and a better communicator. I am grateful to him for offering many opportunities to attend conferences and meet other researchers and for the confidence he has shown in me. I feel proud and privileged to have worked with him. I have enjoyed all our discussions over various matters from which I have learned valuable lessons of life.

I would like to thank my co-advisor, Professor Fengqing Zhu for her helpful discussion and critical thinking. Her analytical approach to solving a complex problem and attention to the details has pushed me past my limits. I would like to thank Professor Carol J. Boushey for the passion and energy that she put into the TADA project. She has helped me understand the importance of the project from a nutritionists point of view. It has always been a great pleasure working with her, especially when she took Ziad and me to one of the Indiana Reservations in Seattle. I would also like to thank Jim, Professor Bousheys husband, for his kindness and the generous gift of two books on the economy. I thank my other advisory committee members, Professor Jan Allebach and Professor Zygmunt Pizlo for their valuable suggestions and insight.

It has been a great pleasure being part of the TADA team. I thank Ms. Chang Liu, Mr. Sri Yarlagad and Mr. Shaobo Fang for being wonderful teammates. I also wish to give special thanks to the senior members who I have worked with: Dr. Chang Xu, Dr. Ye He, Dr. Ziad Ahmad for laying a solid foundation for the project and being extremely helpful to solve system level issues. I want to thank Professor Deborah Kerr for her insight in the development of the TADA mobile applications, and I truly appreciate the fundings from her and Curtin University. I would also like to thank the National Institutes of Health for their generous sponsorship that makes this dissertation possible.

I was very fortunate to have learned ECE 662 Computer Vision from Professor Avinash Kak, ECE637/641 Image Processing from Professor Charles Bouman, ECE580 Optimization Methods from Professor Stanislaw H. Zak, ECE638 Digital Color Imaging Systems from Professor Jan Allebach and ECE538 Signal Processing from Professor Michael Zoltowski. Hats off to all the professors who have devoted their effort to make the courses so much fun and I would love to recommend these classes to anyone who is interested in image processing and computer vision.

I would like to thank Dr. Carla Zoltowski of Engineering Projects In Community Service (EPICS) for offering me the position of a graduate teaching assistant. I would also like to thank Professor Raymond DeCarlo, Hung-Yi Lo, and Nikhil Jali. We have had a great time in the lab of ECE202 Linear Circuit.

Video and Image Processing Laboratory (VIPER) is an ideal place to work on challenging signal processing problems with a group of talented engineers. I have been very fortunate to be a part of this vibrant research environment. I wish to thank Khalid Tahboub and Joonsoo Kim for being good “buddyboys” along the road to graduation and being open to any discussion regarding our research and future career. I am also thankful to Mr. Daniel Mas, Mr. Javier Ribera, Mr. David Guera Cobo, Dr. Albert Parra Pozo, Mr. Soonam Lee, Mr. Yuhao Chen and Mr. Chichen Fu for the exuberance they have consistently shown. I would like to thank my other VIPER colleagues: Ms. Qingchaung Chen, Mr. Jiaju Yue, Ms. Jeehyun Choe, Ms. Dahjung Chung, Dr. Neeraj Gadgil, Ms. Blanca Delgado, Ms. Thitiporn Pramoun, Ms. Kharittha Thongkur and Dr. Bin Zhao.

Last but not least, I want to thank my parents for their unconditional love. I thank them for raising me to a humble and optimistic person. I am blessed by the bonds we share. I also want to thank Ms. Yan Gu for always supporting me and helping me to become a better person.

This work was sponsored by the US National Institutes of Health under grant NIH/NCI 1U01CA130784-01 and NIH/NIDDK 2R56DK073711-04,1R01-DK073711-01A1.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

3D	Three Dimentional
ACE	Automatic Color Equalization
BD	Blind Deconvolution
BoF	Bag of Features
BTTB	Block Toeplitz with Toeplitz Blocks
CAM	Class Activation Map
CHAT	Connecting Health and Technology
CNN	Convolutional Neural Networks
CRF	Conditional Random Fields
DCD	Dominant Color Descriptor
EFD	Entropy-based categorization and Fractal Dimension
E-TADA	Experiments TADA Database
FNDDS	USDA Food and Nutrient Database for Dietary Studies
GAP	Global Average Pooling
GFD	Gabor-Based Image Decomposition and Fractal Dimension Estimation
GMAP	Global Max-Average Pooling
GMP	Global Max Pooling
HoG	histogram of oriented gradient
PSF	Point Spread Function
HTTPS	Hyper Text Transfer Protocol Secure
HVS	Human Visual System
I-TADA	Image TADA Database
ILSVRC	ImageNet Large Scale Visual Recognition Competition

KNN	K Nearest Neighbors
LBP	Local Binary Pattern
LDC	Linear Distance Coding
MDSIFT	Multi-scale Dense SIFT
NCut	Normalized Cut
ODS	Optimal Dataset Scale
OIS	Optimal Image Scale
PRI	Probabilistic Rand Index
RBF	Radial Basis Function
RCT	randomised controlled trial
REST	Representational state transfer
SCD	Scalable Color Descriptor
SIFT	Scale Invariant Feature Transforms
SPP	Spatial Pyramid Pooling
SVM	Support Vector Machine
TADA	Technology Assisted Dietary Assessment
VoI	Variation of Information
VT	Vocabulary Tree

ABSTRACT

Wang, Yu. Ph.D., Purdue University, August 2017. Learning Based Image Analysis With Application In Dietary Assessment and Evaluation. Major Professors: Edward J. Delp and Fengqing Zhu.

Mobile devices will transform the healthcare industry by increasing accessibility to quality care and wellness management. Accurate methods to assess food and nutrient intake are essential. We have developed a dietary assessment system, known as the mobile Food Record (mFR) to automatically estimate food type, nutrients and energy from a food image captured by a mobile device. Color information is of great importance in our mFR system and it serves as a key feature to identify foods. Thus, a preprocessing step including color correction and image deblurring is necessary to ensure that we can utilize the image for the further analysis. We present an image quality enhancement technique combining saliency based image deblurring and color correction using LMS color space.

The accurate estimate of nutrients is essentially dependent on the correctly labelled food items and sufficiently well-segmented regions. Since food recognition also largely relies on the interest region detection or segmentation, image segmentation plays a critical role in our mFR system. We propose a generic segmentation method that combines normalized cut and superpixels. Experimental results suggest that the proposed method using multiple simple features is effective for food segmentation.

To achieve high classification accuracy in food images is challenging due to large number of food categories, lighting and pose variations, background noise and occlusion. Deep learning with big data has shown its dominance in various object detection tasks. In this thesis, we compare deep features with the handcrafted features in terms of classification performance and we also introduce a weakly supervised segmentation method based on class activation maps using only the label of the input image to deal with sparsity of ground-truth masks or bounding boxes. Furthermore, a 3-stage food localization and identification technique using end-to-end deep networks is proposed. Finally, we integrate contextual information into our mFR system and introduce the personalized learning model to further improve the food recognition accuracy. The result indicates that our contextual models are promising and further investigation is warranted.

1. INTRODUCTION

1.1 Problem Formulation and Research Goals

The objective of the work is to provide a user-friendly application to record daily food intake and improve the accuracy of dietary assessment by learning from different sources of data. Traditional dietary assessment mainly relies on the written and orally reported methods. It generally requires a nutritionist to lead and complete a survey. In the modern society, such methods are considered inefficient and not feasible for everyday dietary monitoring [1].

The ubiquitous mobile devices have gradually changed the landscape of healthcare industry and research. Recent reports suggest that in 2015 nearly 66 percent of American adults are smartphone owners and almost 20% of Americans rely on a smartphone to stay connected to the world [2]. More and more companies start to develop tools to help to monitor people's health conditions. Smartphone with its inherent capability of photography, light-load computation, and accessibility to the Internet becomes the most popular tool for younger generations to keep track of their fitness. For example, Tuingle [3], is designed to help users to record what they have eaten and estimate calories based on users' input. However, there are not many user-friendly and scientifically verified applications that automatically analyze food intake from pictures. Some studies [4, 5] highlights the importance of using eating occasion images to record and estimate dietary intake versus classical handwritten approaches.

The Technology Assisted Dietary Assessment (TADA) system aims to provide valuable insights for fitness monitoring as well as mounting intervention programs for chronic diseases, such as diabetes. The TADA system, which includes a web interface, a mobile application, and a backend image analysis server, is designed to automatically estimate food type, volume, nutrients, and energy from a food image captured by a mobile device. The analysis process starts with capturing a pair of before and after images of food and beverages consumed using the TADA mobile application (available both on iOS and Android). These images are asynchronously uploaded to our backend server whenever the Internet connection is established. The users are instructed to include a fiducial marker while taking images. The fiducial marker serves as a reference to the known dimension and color space [6–8]. On the backend server, a sequence of image processing techniques, such as color correction, image segmentation, is used to identify food and estimate the food portion [9–11]. Finally, the energy and nutrients of a food image are estimated based on the USDA Food and Nutrient Database for Dietary Studies (FNDDS) [12, 13].

One of the outstanding designs in the TADA system is that the automatic analysis is based on a single image. It means no other forms of user input are necessary. On the contrary, the approach described in [14] requires the user to take multiple pictures of a meal scene. In addition, we made our mobile application so intuitive and easy to use that even 3-year old child can take a good food image [15]. However, the idea of putting as little burden on the user as possible leaves us various challenges. The following part of this section will discuss those challenges in greater details. First of all, there is no regulation on how to prepare foods. Thus, even the same food or dish

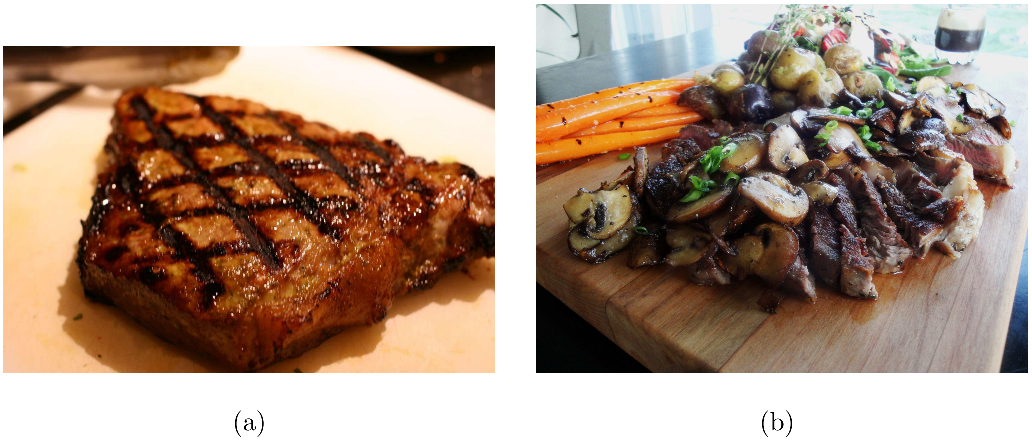
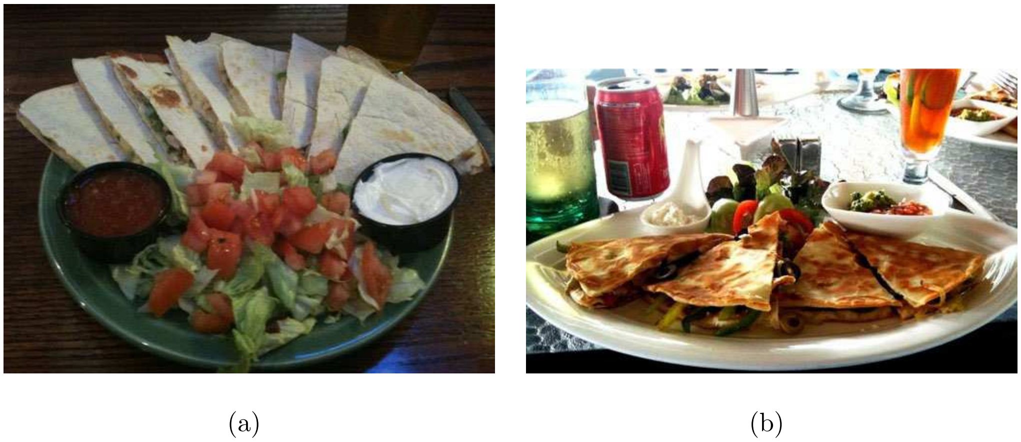


Fig. 1.1.: Images of foods with different cooking style.

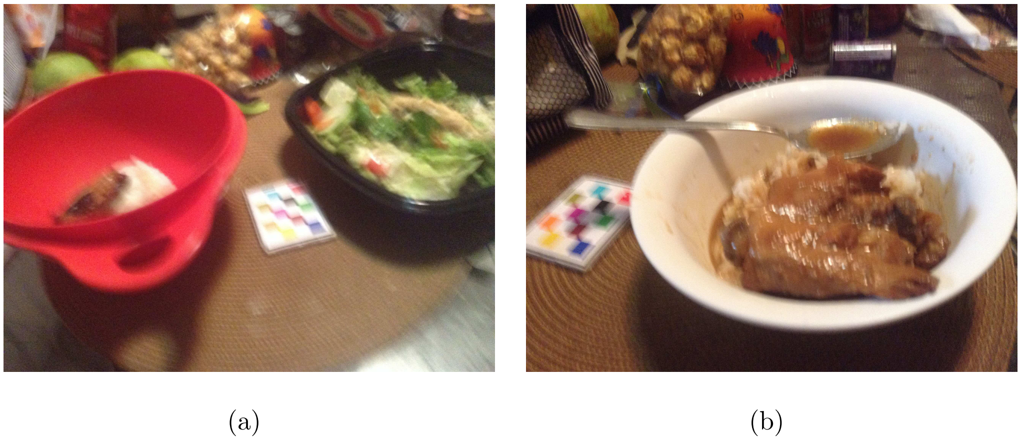
varies from time to time and person to person. Besides, diverse cooking style and personal preference make similar foods or dishes look entirely different. For example, some people prefer the steak with mushroom and onion on it while some prefer it plain (see Figure 1.1). Secondly, various lighting conditions and different camera sensors make food images more complicated. People may take pictures of food in restaurants, at home or even outdoor. Lighting conditions have a huge impact on the food appearances (see Figure 1.2). Even if we assume lighting conditions are exactly same, different cameras may reproduce colors that are off the true tone. Thirdly, even



(a)

(b)

Fig. 1.2.: Images of foods taken in different lighting conditions.



(a)

(b)

Fig. 1.3.: Two examples of blurry food images.

though we have implemented an image quality checker on the mobile phone, users may still take and send blurry images (see Figure 1.3). There are cases where the user may be reluctant to retake the picture of his/her meal or the image on the small mobile telephone screen appears to be good enough.

Last but not the least, noisy background and occlusion by utensils or other objects impose more challenges on finding robust segmentation methods to obtain the objects of interest. Food and drinks in opaque containers are much harder to analyze. Two examples of distracting utensils and opaque containers are illustrated in Figure 1.4.

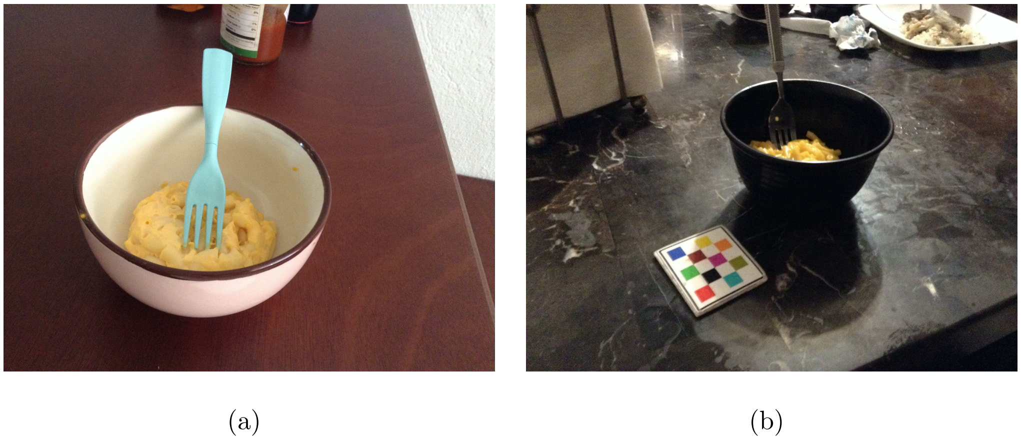


Fig. 1.4.: Images of foods which are held in an opaque container or partly occluded by utensil.

1.2 Overview of Image-Based Dietary Assessment

In recent years, image-based “nutrition” systems and services have become increasingly popular, especially the ones making use of the mobile devices. To some extent, these systems are capable of taking images of foods eaten at different eating occasions and using the images as part of a food diary to assist users in recording their diets. These systems include but not limit to *FoodLog* [16], *EButton* [17], *Tuingle* [3], *FoodCam* [18], *DietCam* [14]. DietCam [14] uses images acquired from multiple views to do the analysis. FoodLog [16] provides both a mobile application and cloud service that allows users to record daily dietary intake by acquiring images of food. A user must first identify the names and quantities of food items and then the nutrient values are estimated, which places a large portion of the dietary assessment on the user (or on a human analyst). EButton is a wearable device with limited image capturing and processing capabilities. It is particularly designed for the visually impaired individuals to record their diets. Due to the ubiquity and accessibility of smartphones, it might hardly be an option for the others. Tuingle [3] can recognize food items automatically, but it only works for images with a single food item and requires users to manually input portion size to complete the energy and nutrition assessment.

Besides the aforementioned systems, extensive research and studies have been conducted in the area of food image analysis. In 2009, the Pittsburgh Fast-Food Image Dataset (PFID) [19], containing 4545 still images and 606 stereo image pairs was released. The data was collected by obtaining three instances of 101 foods from 11 popular fast food chains and capturing images and videos in both restaurant conditions and a controlled laboratory setting. The authors [19] proposed two baseline recognition methods based on the PFID, and they used a Support Vector Machine (SVM) to classify color and Scale Invariant Feature Transforms (SIFT) features. As the combination of SVM and simple features performed poorly on the PFID, the paper mainly discussed why the baseline approach failed and promoted the dataset.

A multi-kernel learning (MKL) based food recognition system was later introduced [18]. The system used a multiple kernel SVM to integrate a color histogram feature, a Gabor feature and a bag of SIFT features. T. Joutou et al. [18] reported the classification accuracy of 61.34% for 50 kinds of foods. They also tested the prototype system on 166 food images acquired in real-world condition and obtained 37.55% accuracy.

S. Yang et al. [20] proposed a food recognition method which was specialized for American fast food such as hamburger, pizza, and tacos. They defined eight basic food materials such as bread, beef and cheese, and recognized them and their relative position in a food image. Finally, they classified images into one of 61 categories using detected materials and their relations. They achieve the 28.2% classification rate on PFID. Zong et al. [21] also proposed a food recognition system employing SIFT detector and Local Binary Pattern (LBP). They achieved the better classification rate than Yang's method [20] on PFID.

In 2014, M.M. Anthimopoulos et al. [22] proposed another food recognition approach based on an optimized bag of features (BoF). A dedicated dataset consisting of roughly 5000 food images of 11 categories was created. The system achieved 78% classification accuracy using a hierarchical k-means approach and a linear SVM. However, images from the dataset were acquired under strictly controlled conditions. Thus the reported result does not reflect the real-world performance of the system.

In fact, all of the works assumed that one food image contained only one food item, and the food item should occupy a major part of the image. However, we cannot make such assumption in the real world. Y. Matsuda et al. [23] described a two-step method to analyze multi-food images. Object detectors like a deformable part model (DPM) [24], a circle detector and the JSEG region segmentation [25] were first used to obtain candidate regions. Then similar to [18], MKL was applied on each candidate region to classify a fusion of various hand-engineered features, such as SIFT, Color-SIFT and histogram of oriented gradient (HoG). The method was tested on a dataset containing 9060 single-food or multi-food images of 100 food classes. The authors reported 55.8% Top 10 classification rate for multi-food images. In 2014, Y. Kawano et al. [26] extended their work on multi-food image analysis by integrating Fisher Vectors and testing on a larger scale dataset i.e. UEC-FOOD256 dataset. They achieved the 74.4% classification rate for the top 5 category candidates. In the same year, another large-scale food dataset was introduced. The authors denoted the dataset including 101,000 images of 101 food classes as "Food-101" [27]. Concomitantly, a Random Forests (RF) based approach was proposed to mine discriminative visual components, and it achieved an average accuracy of 50.76% on the Food-101 dataset. Some researchers [28, 29] focused on foods from restaurants, and they analyzed multiple features and classifiers which were more effective in their applications.

As of 2015, deep learning has already shown its dominance on various computer vision tasks [30–32]. CNNs also gradually penetrates in the field of food image analysis. Some works [33–35] used either an end-to-end deep neural network or deep features with variations of SVM to obtain better results. In [33], the authors trained a CNN on the Food101 dataset [27] with 101 food categories and used domain adaption to improve the classification performance. [34] reports achieving the top 1 classification accuracy, 78.77% and 67.57% on the UEC-FOOD100/256 dataset.

Due to the complexity of food images (e.g. occlusion and cluttered background), food segmentation remains an open research problem. Most of the food classification work mentioned above only deal with single food images. Thus image segmentation is not necessary. For example, [28] focused on restaurant foods and utilize GPS information to get restaurant menus. With the correct restaurant information, they can readily map the detected dish to the corresponding nutrient table. In [23], ten types of food with containers were examined using a deformable part model and a circle detector to constrain the food region of interest. Bettadapura et al. [29] used a hierarchical segmentation method in their implementation. *Im2Calories* [35] is an end-to-end system that utilizes multiple deep networks. In *Im2Calories*, GoogLeNet [36] was used for classification tasks and DeepLab [37] was used for the semantic segmentation.

Lacking a large and multi-purpose food image dataset is another problem haunting the dietary assessment community. Before Food-101 and UEC-FOOD256 being broadly adopted, every research group was creating and using its own dataset. Therefore, sometimes the reported results are hardly comparable from group to group. More efforts are definitely needed for creating large-scale food image datasets for both classification and segmentation [27, 35, 38, 39]. To deal with the sparsity of data for a particular food, J. Zheng et al. [40] proposed a superpixel-based Linear Distance Coding (LDC) framework. Their approach demonstrated a promising result in both accuracy and robustness on a challenging small food image dataset where only 12 training images are available per category [40].

We define portion size estimation as the process of determining how much food (in cm^3 or grams) is present in the food image. Food volume estimation (or portion size estimation) is a challenging problem, since the food preparation process and the way food is consumed can cause large variation in food shape and appearance. Many existing image-based work on food volume estimation require either modifying the mobile device such as 3D range finding [41], acquiring multiple images [42, 43], or video [44] which is not desirable for users trying to collect information about their diets and can contribute to poor compliance with these methods.

1.3 Contributions of This Thesis

In this thesis, we extend our previous work by introducing deblurring and color correction in the preprocessing and proposing a personalized learning model to improve classification accuracy as a postprocessing step. Color information is of great importance in our mFR system and it serves as an essential feature to identify foods. Thus, a preprocessing step including color correction and image deblurring is necessary to ensure that we can utilize the image for the further analysis. We present an image quality enhancement technique combining saliency based image deblurring and color correction using LMS color space. We also propose a generic segmentation method that combines normalized cut and superpixels to replace the segmentation refinement scheme.

To achieve high classification accuracy in food images is challenging due to a large number of food categories, lighting and pose variations, background noise and occlusion. Deep learning with big data has shown its dominance in various object detection tasks. In this thesis, we compare deep features with the handcrafted features regarding classification performance, and we also introduce a weakly supervised segmentation method based on class activation maps using only the label of the input image to deal with the sparsity of ground-truth masks or bounding boxes. Furthermore, a 3-stage food localization and identification technique using end-to-end deep networks is proposed.

The main contributions of this thesis are listed as follows:

- We propose a polynomial model based color correction method using LMS color space. The proposed method requires the fiducial marker present in the scene and then it computes a color correction matrix based on the detected 11 colors and the pre-measured ground-truth colors. Based on the experimental results, it demonstrates more accuracy compared to other color correction models using CIELAB or sRGB space.

- We introduce a de-blurring scheme using a saliency map that runs up to 5 times faster than its counterpart. Since our users are required to include a fiducial marker when they take pictures, it is reasonable to make use of its appearance. From the testing results, the fiducial marker almost always is detected as a salient region. And since it possesses visually recognizable features like corners, edges and contrast color patches, we use the corresponding salient region to estimate blur kernel.

- We propose a segmentation method based on the normalized cut (Ncut) and superpixels. The idea is to mine higher level features from superpixels and reduce the size of the affinity matrix in Ncut. The method relies on color and texture features for fast computation and efficient use of memory. We also introduce an object based segmentation evaluation method, especially for multi-food images. Our method achieves competitive results using the Berkeley Segmentation Dataset and outperforms some of the most popular techniques in a food image dataset.

- Successful methods for object segmentation rely on a large amount of labeled data on the pixel level. However, such training data are not yet available for food images and expensive to obtain. We describe a weakly supervised convolutional neural network (CNN) which only requires image level annotation. We propose a graph-based segmentation method which uses the class activation maps trained on food datasets as a top-down saliency model. We evaluate the proposed method for both classification and segmentation tasks. We achieve competitive classification and segmentation accuracy compared to the previously reported results.

- We investigate end-to-end CNN structures and train a dedicated food localizer using a region proposal based network. We adopt the deformable convolutional layer to improve food localization accuracy. Furthermore, we propose a three-stage food analysis pipeline. Finally, we attack the proposed system using adversarial examples and use them to investigate overfitting in the domain of food images.
- We propose to incorporate temporal context to improve food classification accuracy. We use recursive Bayesian estimation to achieve active learning from users' feedback through our mobile applications. Three auxiliary datasets were created to simulate users' dietary records in one month. We also propose a context based image analysis system that integrates food co-occurrence pattern and temporal context into a personalized learning model. Experimental results showed the classification accuracy was improved by 15.56% on average compared to the automatic image analysis without contextual information.

1.4 Publications Resulting From This Work

Journal Paper

1. **Yu Wang**, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. "Food image segmentation and classification using deep networks" *IEEE Journal of Biomedical and Health Informatics*, to submit, June 2017.

2. **Yu Wang**, Ye He, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. ”Context based image analysis with applications in dietary assessment and evaluation” *Multimedia Tools and Applications*, under review, Nov 2016.

3. Carol J. Boushey, Edward J. Delp, Ziad Ahmad, **Yu Wang**, Sparkle M. Roberts, and Lynn M. Grattan. “Dietary assessment of domoic acid exposure: What can be learned from traditional methods and new applications for a technology assisted device.” *Harmful Algae*, vol.57 pp.51-55, 2016.

Conference Paper

1. **Yu Wang**, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. “Weakly supervised food image segmentation using class activation maps”, *Proceedings of the IEEE International Conference on Image Processing*, to appear, September, 2017

2. **Yu Wang**, Shaobo Fang, Chang Liu, Fengqing Zhu, Deborah A Kerr, Carol J Boushey, and Edward J Delp. “Food image analysis: the big data problem you can eat!” *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, November 2016.

3. **Yu Wang**, Chang Liu, F. Zhu, C. J. Boushey, and E. J. Delp, “Efficient superpixel based segmentation for food image analysis, *Proceedings of the IEEE International Conference on Image Processing*, September 2016.

4. **Yu Wang**, Ye He, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. “The use of temporal information in food image analysis,” *New Trends in Image Analysis and Processing - ICIAP 2015 Workshops, Lecture Notes in Computer Science*, Vol. 9281, Springer International, pp. 317-325, September 2015.

5. **Yu Wang**, Chang Xu, Carol J. Boushey, Fengqing Zhu, and Edward J. Delp. “Mobile image based color correction using deblurring,” *Proceedings of the*

IS&T/SPIE Conference on Computational Imaging, pp. 940107-940107, February 2014.

2. OVERVIEW OF THE TADA SYSTEM

2.1 System Architecture

Over the past nine years, we have been investigating the use of images and the associated meta-data to assess dietary intake. We have developed the Technology Assisted Dietary Assessment System (TADA) to acquire and process food images [4, 9, 10] that a user takes of their meal before and after eating occasions.

As illustrated in Figure 2.1, the TADA system consists of three main components: the TADA mobile applications, the backend server which is charge of analyzing the uploaded images and hosting the frontend web interface, and the dedicated database that manages images and other types of data.

The TADA system allows users to log their eating occasions using mobile phones. Image processing and computer vision analysis methods are then used to determine the food type, volume, the energy (kilocalories) and nutrients of the food [10,11,45,46]. The TADA system has been used for more than 14 scientifically implemented user studies, including environments in the wild, by more than 800 users who have taken more than 60,000 food images. For example, the Connecting Health and Technology study [47, 48] was a six-month randomised controlled trial (RCT) in 247 young adults (18-30 years). The study aimed to evaluate the effectiveness of tailored dietary feedback and weekly text messaging to improve dietary intake of fruit, vegetables and junk food over six months among a population-based sample of men and women (aged 18 to 30 years) [47, 48].

The system process starts with the user acquiring eating occasion images with our mobile applications. The images and related meta-data including date, time, geolocation and device model are sent to the server (step 1). The automatic analysis for a pair of before and after images is mainly composed of image segmentation, food

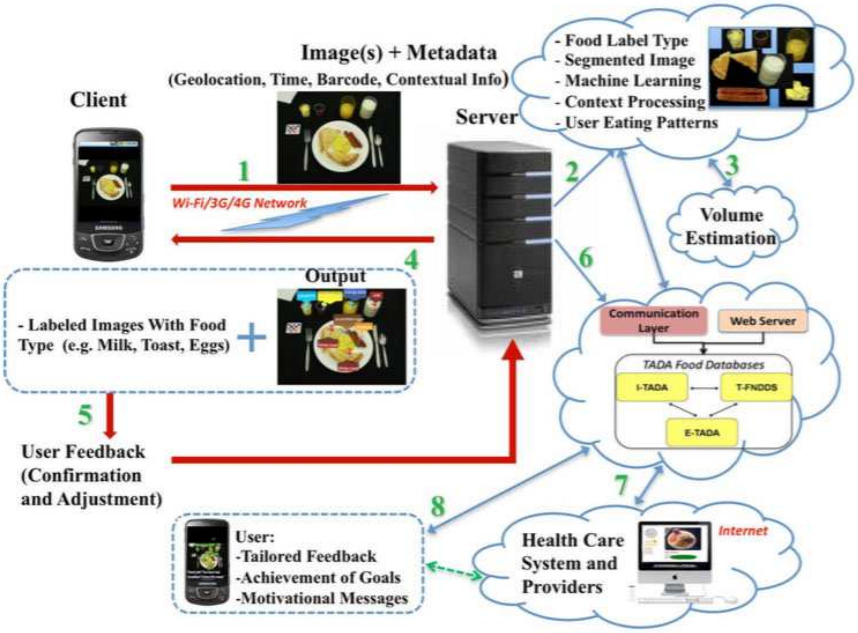


Fig. 2.1.: The architecture of the TADA system.

identification, contextual refinement and weight estimation (step 2 and step 3). After the analysis is done on the server, the structured result including food labels and positions are sent back to the user for review. The user can confirm and/or modify the results (step 4 and step 5). The feedback from the user is stored in the database as a potential groundtruth and is used to refine the previous image analysis results. As a part of our database system, the USDA Food and Nutrient Database for Dietary Studies (FNDDS) is used to estimate the energy and nutrient information given a food label and weight (step 3 and 6). The database contains the most common foods in the US, their weights, nutrient values, and food densities. Finally, these results are displayed on the TADA web interface for the user and the healthcare community (step 7 and 8). The user can keep track of his/her dietary history and the nutrient professionals can utilize the system for dietary recommendations and planning.

2.2 User, Food and Image Oriented Database

To manage and store a large amount of images and meta-data uploaded to the server, we have designed a database system as shown in Figure 2.1. The database system is powered by PostgreSQL and was structured around three key elements namely images, foods and users.

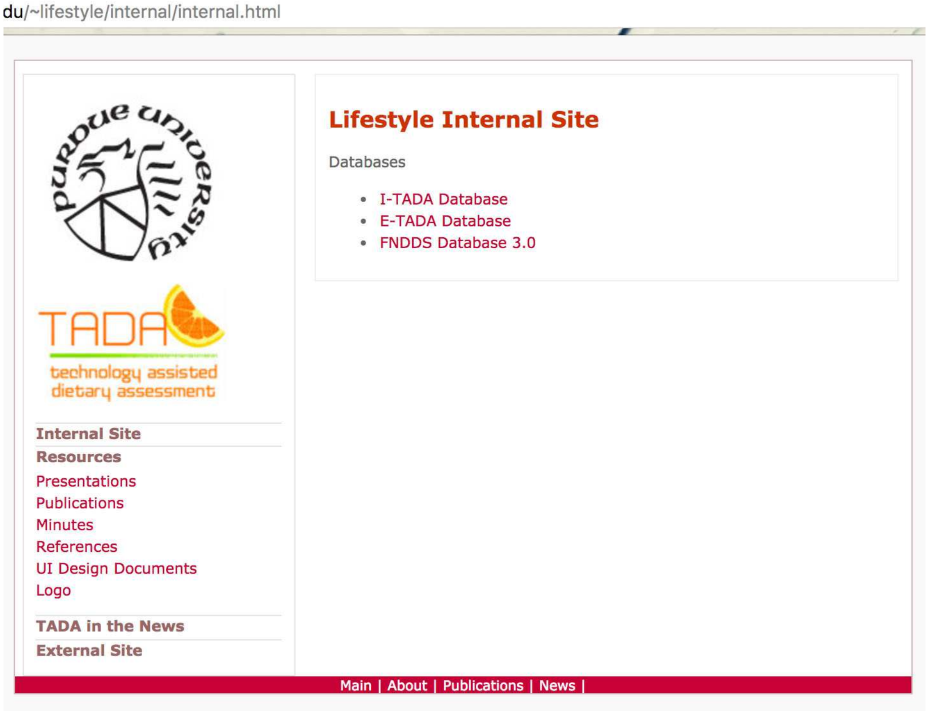
I-TADA is an image database that contains information related to eating occasion images. It is composed of the paths of the original images, the relationship of before and after meal image pairs, the users who acquired the images, the studies that the images belong to and the specifications of the images. E-TADA is a user information dataset that stores the personal information and data of the user studies. We associate each participant with a specifc ID, thus E-TADA contains the IDs, some personal information (e.g. date of birth and weight/height) and the studies which the participants enroll. T-FNDDS is an extension of the FNDDS with visual descriptions generated by the image analysis method and barcode information associated with the packaged food items. These three databases are interconnected to provide a platform for researchers to discover dietary patterns of the users.

2.3 Web Interface

A TADA web interface is the front-end that gives researchers access to the images and various metadata in the database and provides the ability to upload images and retrieve data of interest collectively.

Web.py framework was used to render the website and connect the web content to our database. However, the web.py structure as we previously implemented was insecure, redundant and hard to maintain.

We replace the Python-based web service with PHP, which not only improves the security of the website but also is more efficient to maintain as we implement a modular design. Figure 2.2 to Figure 2.4 showcase some examples of the PHP-powered web interface.



du/~lifestyle/internal/internal.html

Lifestyle Internal Site

Databases

- I-TADA Database
- E-TADA Database
- FNDDS Database 3.0

Fig. 2.2.: Internal website for E-TADA and I-TADA.

2.4 Mobile Applications

High-speed multimedia processors and data network capability make mobile devices ideal as a data collection tool for dietary assessment. This has led to an increasing demand in developing applications on mobile devices to help track dietary intake. Using the iPhone as an example, according to MobiHealthNews' report Consumer Health Apps for Apples iPhone, there were more than 9,000 health applications in July 2011 with 1,263 of them being diet-related. None of the current commercial mobile applications indicate that the standard food composition databases are used. While these applications represent a step forward towards improving dietary habits, it is not obvious how they can be used by healthcare professionals for dietary assessment. In this section, we introduce the design and usability of the TADA mobile application.

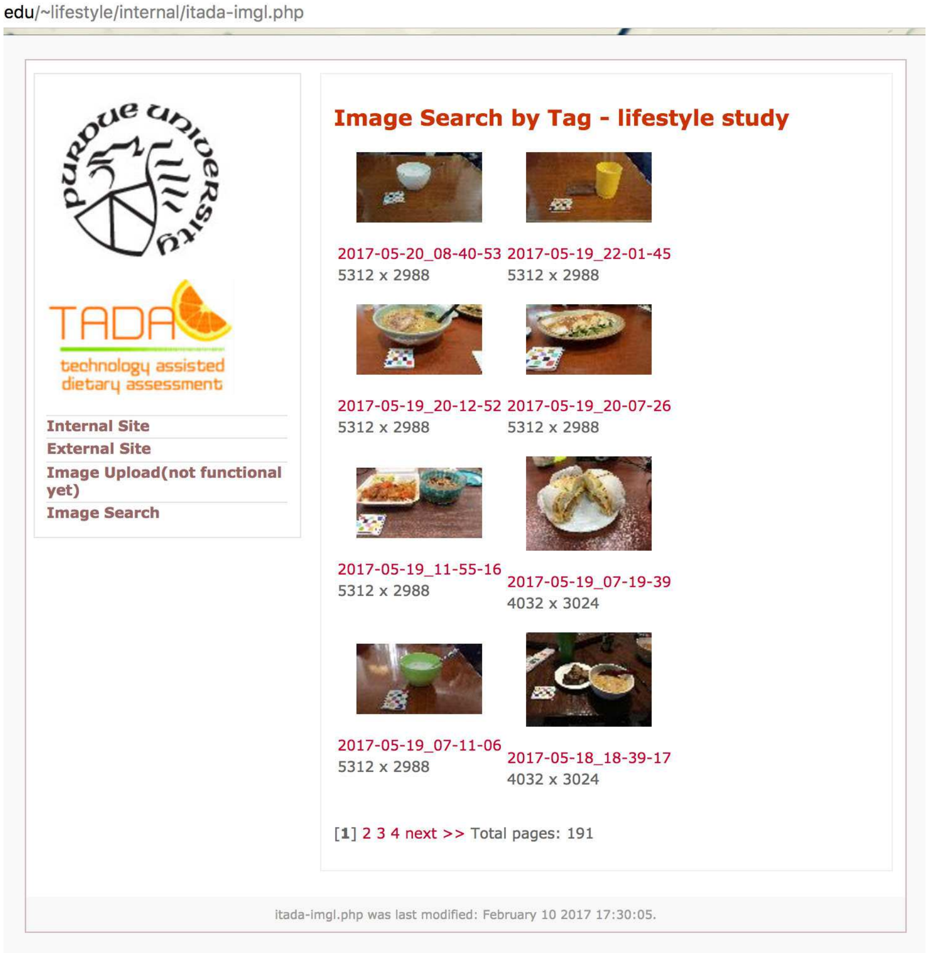

edu/~lifestyle/internal/itada-imgl.php
TADA
technology assisted dietary assessment
Internal Site
External Site
Image Upload(not functional yet)
Image Search

Image Search by Tag - lifestyle study

2017-05-20_08-40-53 5312 x 2988
2017-05-19_22-01-45 5312 x 2988
2017-05-19_20-12-52 5312 x 2988
2017-05-19_20-07-26 5312 x 2988
2017-05-19_11-55-16 5312 x 2988
2017-05-19_07-19-39 4032 x 3024
2017-05-19_07-11-06 5312 x 2988
2017-05-18_18-39-17 4032 x 3024

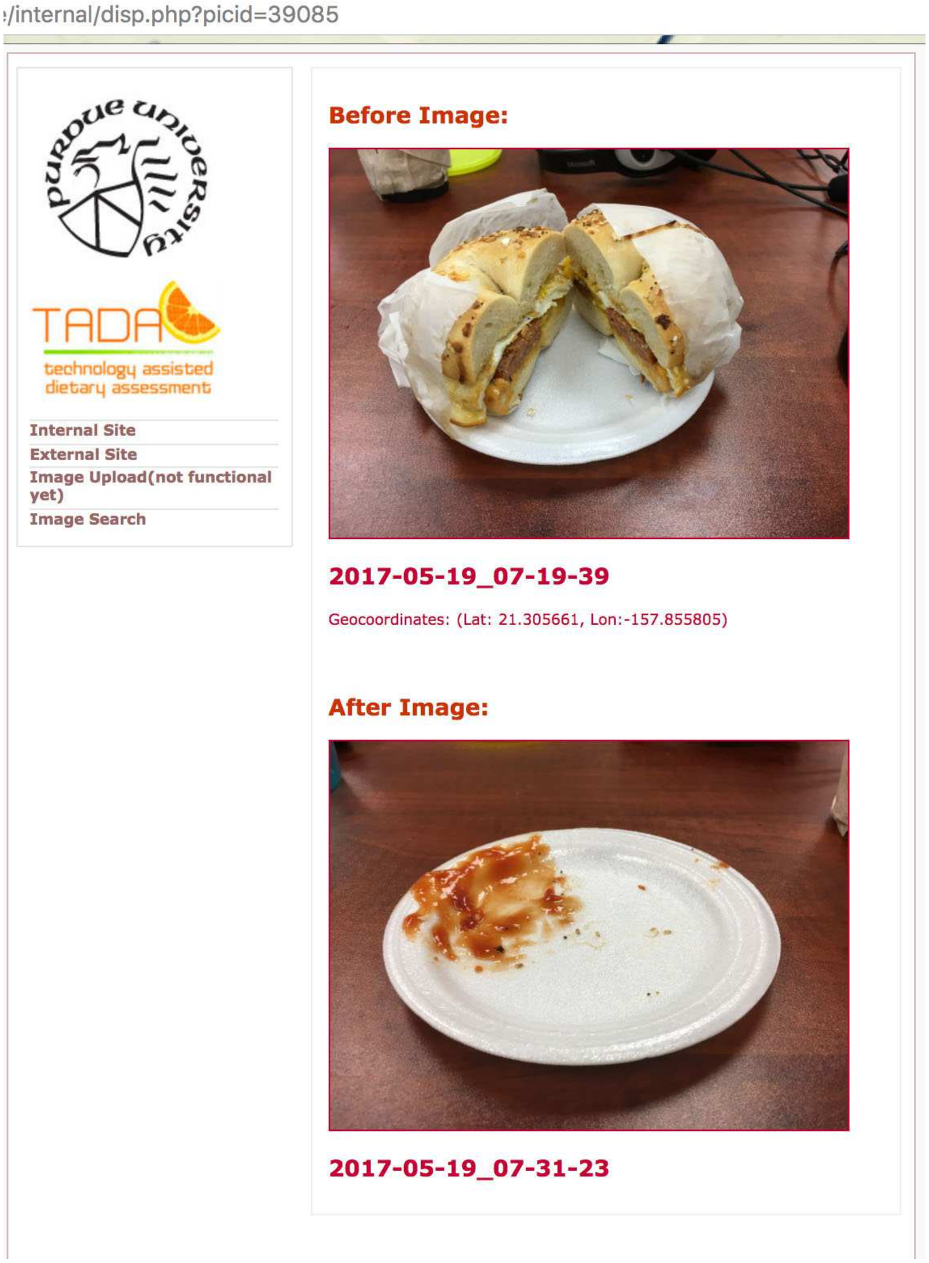
[1] 2 3 4 next >> Total pages: 191

itada-imgl.php was last modified: February 10 2017 17:30:05.

Fig. 2.3.: List of images from a study.

2.4.1 iOS

As one of most popular mobile system in US, the iOS version of the TADA mobile application was first developed in 2008. The application has been validated with many user studies [5, 47–50] and it uses scientifically verified food dataset [12]. Figure 2.5 shows the iOS user interface of the TADA application.



:/internal/disp.php?picid=39085

Before Image:

2017-05-19_07-19-39

Geocoordinates: (Lat: 21.305661, Lon:-157.855805)

After Image:

2017-05-19_07-31-23

PURDUE UNIVERSITY
TADA
technology assisted dietary assessment
Internal Site
External Site
Image Upload(not functional yet)
Image Search

Fig. 2.4.: Before and after images and associated information.

2.4.2 Android

Since 2014, lots of efforts have been put into the development of the Android counterpart, as the Android system has some advantages over iOS and we simply cannot ignore the Android users when we are expanding our user studies. One of the

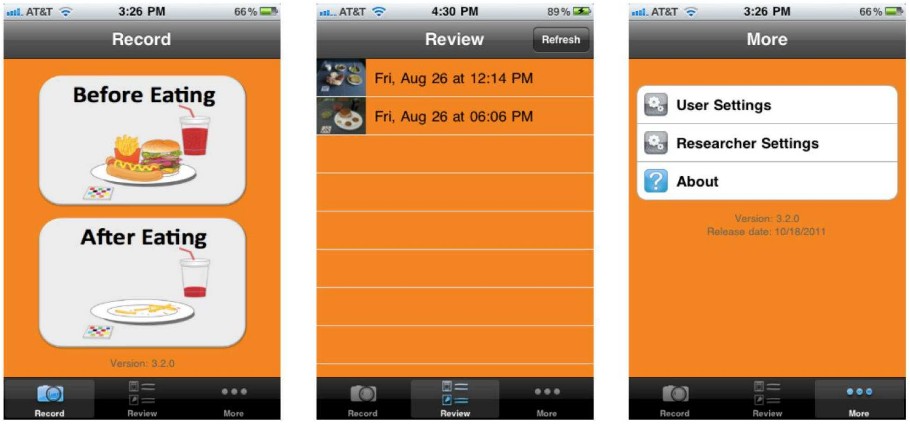


Fig. 2.5.: Main views of the TADA iOS application.

advantages of the Android system is that it is more open for developers to distribute their applications for a longer term. At the beginning, the design of the TADA Android application essentially follows the iOS prototype with some modifications due to the Android API. Most recently, we have developed a new version which focuses more on the material design and higher APIs.

Figure 2.6 illustrates the three main views of the Android user interface: *Record*, *Review* and *More*. In the *Record* view, there are two large buttons with vivid background figures indicating the before and after eating functionality.

Whenever a user wants to record his/her eating occasion, he/she can complete the image acquisition process by taking a pair of images before and after he/she finishes the meal. As the image recording is most frequently used and is the key functionality of the TADA application, we design the *Record* view with two buttons occupying the majority of the screen, so that the user can quickly identify and tap on the correct buttons. The *Record* view is also the default view after the application first launches. The user can easily navigate to other views by tapping on the tab bar at the bottom of the screen.

To assist the user in taking better images, we implemented a customized camera view instead of using the camera application in the Android system. Once the user

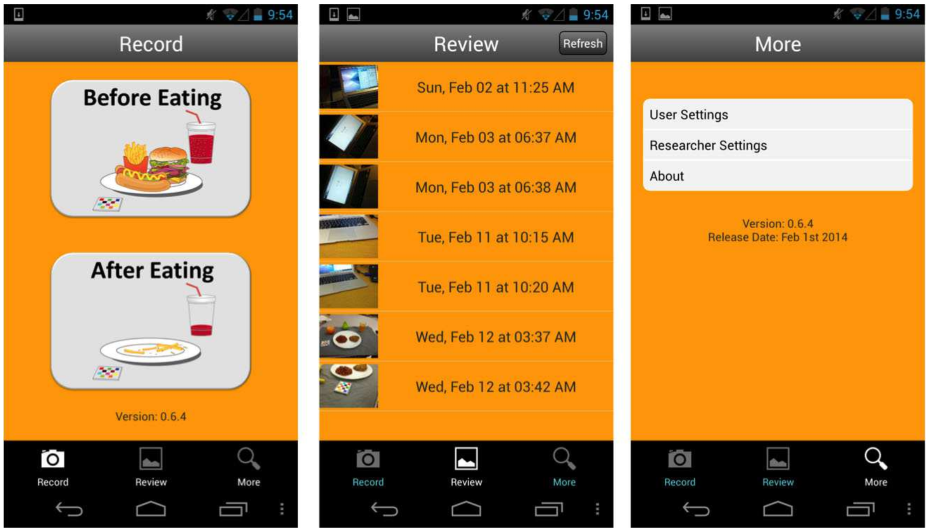
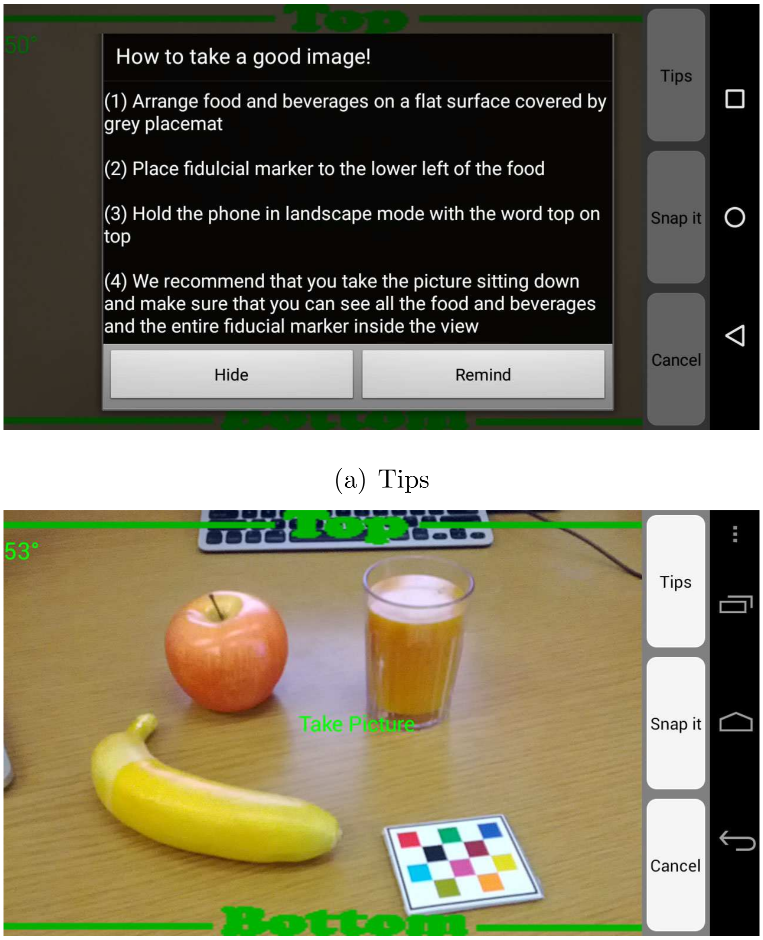


Fig. 2.6.: Main views of the TADA Android application.

clicks on either "before eating" or "after eating" button, the customized camera will be launched (see Figure 2.7). Two most obvious features of the designed camera view are the angle displayed at the top left corner and the image capture guideline. The orientation sensors are used for obtaining the tilting angle of the device, which is useful for estimating the camera positioning and food volume. The guideline turns green when the angle is within the preferred range, i.e. between 45 degree and 60 degree; otherwise, it is red. Whenever ready, the user can click the "Snap It" button to take an image.

An alert notification stating the tips of taking a good image is set to pop up before acquiring an image (Figure 2.7(a)). These tips can be turned off later in the user settings resided in the *More* view.

The user need to confirm saving the image he/she has taken by pressing the "Use" button in Figure 2.8. As Figure 2.8 shows, a set of onboard image quality check including the fiducial marker detection and the blur detection is running. After the quality check is done, the user has an option to save the image anyway if it does not pass the check. If it does, the image is automatically saved on the device and the user interface will return to the *Record* view.



How to take a good image!

(1) Arrange food and beverages on a flat surface covered by grey placemat

(2) Place fidulcial marker to the lower left of the food

(3) Hold the phone in landscape mode with the word top on top

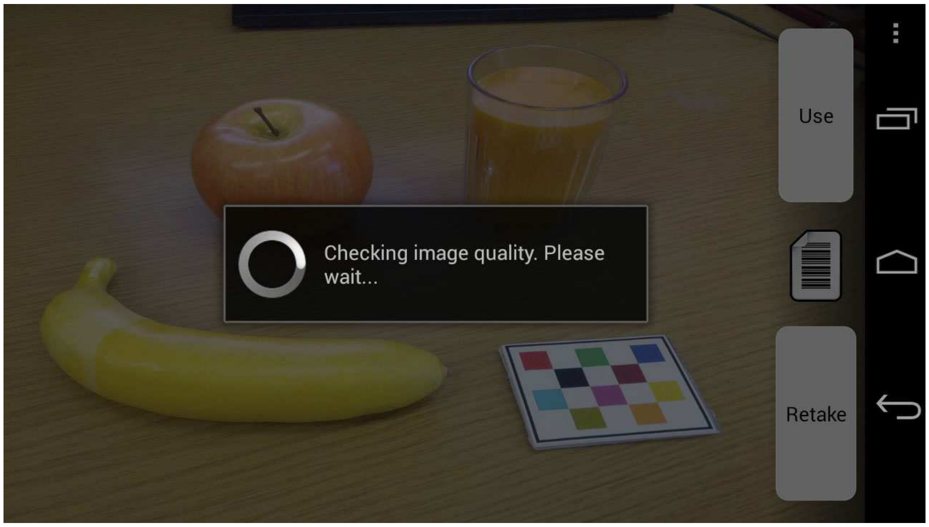
(4) We recommend that you take the picture sitting down and make sure that you can see all the food and beverages and the entire fiducial marker inside the view

(a) Tips

(b) Pose Guidelines

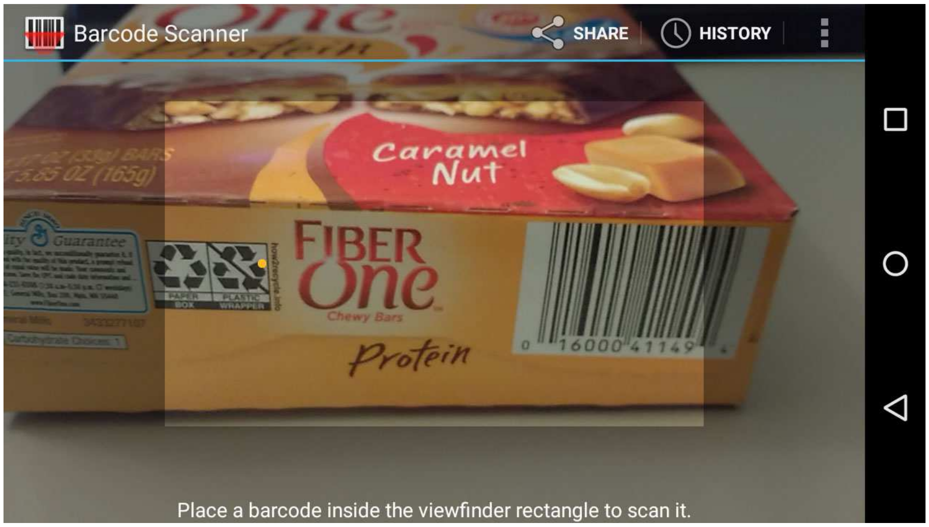
Fig. 2.7.: Image acquistion guide in the TADA Android application.

For each eating occasion, we not only save the image files but also write the metadata of each eating occasion into a specific file with a .rec extension, or the REC file. The REC file stores the device ID, the user ID, and the timestamp of the eating occasion. If the user permits the application to use the location services, the GPS information of each image is written into a file with .gps extension, or the GPS file. If the user wants to use the embedded barcode scanner, the barcode information will be saved to the BAR file. In the current implementation, we used the Zxing barcode scanner and its user interface is shown in Figure 2.9.



Checking image quality. Please wait...

Fig. 2.8.: Image quality checking in the *Preview* view.



Place a barcode inside the viewfinder rectangle to scan it.

Fig. 2.9.: Barcode scanner in the TADA Android application.

Once the after eating image is captured and saved, the pair of images along with the REC file, GPS file and BAR file are uploaded to the server. The background uploading process is handled by the Android system. If the Internet connection is not established or somehow broken, the uploading process will be paused. Once the connection is re-established, the data are sent asynchronously. To comply with the RESTful framework, all the data are sent in a certain format under HTTPS protocol.

The *Review* view shown in the middle of Figure 2.6 is mainly a list of eating occasions that have been analyzed on the backend server. The “Refresh” button on the top right of the screen is for retrieving any available results from the server. Each entry in the list of eating occasions is composed of a thumbnail of the before eating image and its timestamp in an easily readable format. To ensure that the list scrolls smoothly back and forth, the application asynchronously loads thumbnail images and maps each entry to the corresponding before eating image. The user can start reviewing an eating occasion by clicking on any entry in the list. The corresponding before eating image is then displayed in landscape with the food labels/pins reflecting the analysis results received from the server (see Figure 2.10(1)). The user can add, remove, change and confirm any label/pin. If some labels are overlapping, the user can zoom in up to 3 times to have a better view. We use different colors to display the labels and pins, but we reserve the green color for the confirmed labels. Figure 2.10(1) shows all the confirmed pins and labels in this case. To change, remove or confirm a label, the user can tap on the label and then confirm, delete a label by using the tool bar in the bottom of the search food view as shown in Figure 2.10(3). The *Suggested Food* section lists top 4 food suggestions. If the correct label is not in the suggestions, the user can search in the Complete Food List or use the search bar at the top of the view (see Figure 2.10(4)). By clicking on the search bar, an Android search dialog will show up. In the search dialog, each result occupies two lines of text. The first line is the food name while the second line of text is a more detailed description of the food. For instance, as shown in Figure 2.10(4), two types of diet coke are listed in the search result. Whenever there is a missing label, the user can add a label by drawing a contour and associate a food with it (see Figure 2.10(2)(6)). After the user confirms all the modifications, the user feedback is sent to the server and saved in the database. Such feedback can be potentially treated as the groundtruth information, which is extremely important for building a personalized learning model.

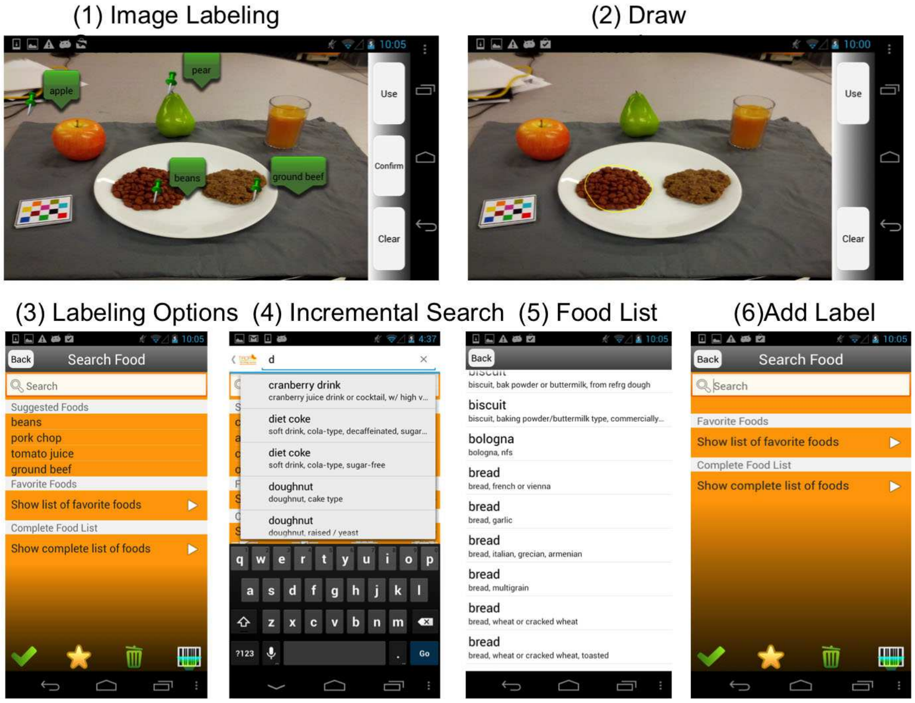
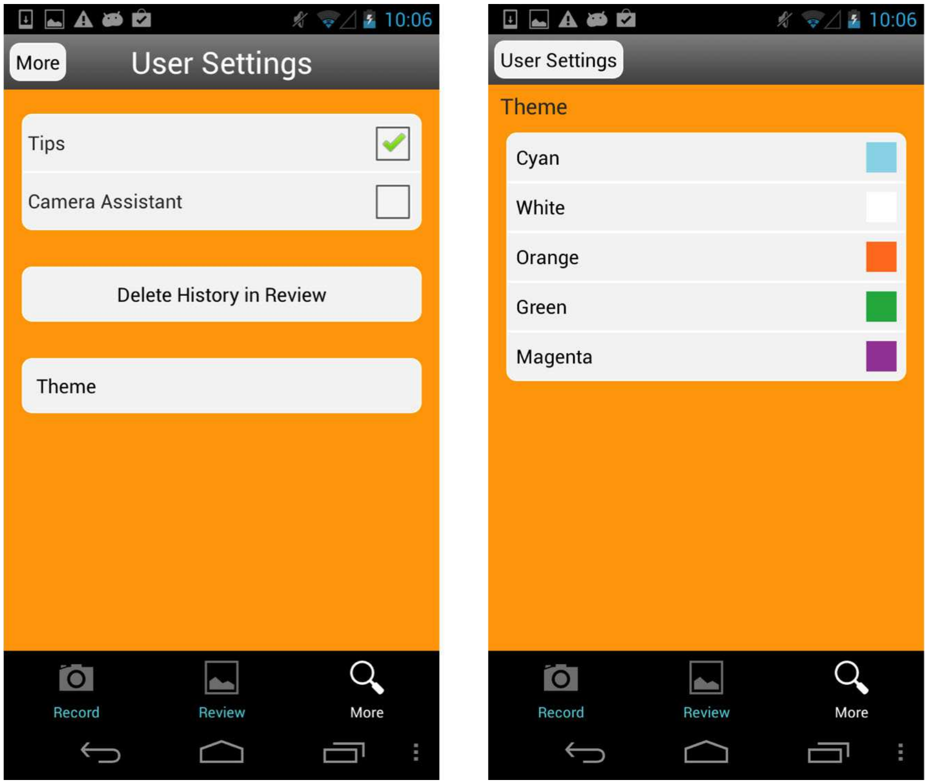


Fig. 2.10.: Image labeling in the review process.

The *More* view is the home for both user settings and researcher settings as shown in Figure 2.6 on the right. The user's preferences of using the camera guideline, the tips and the background color are saved in the system level configuration (Figure 2.11). The "Researcher settings" is a password-protected space, in which the user ID, the server IP and three meal reminders can be set by authorized personnel (Figure 2.12).

As the Android system gradually updates, the TADA application is also evolving to bring out a better user experience. Figure 2.13 shows some features developed for newer versions of the Android system. For example, Figure 2.13(a) demonstrates a material design for the *Record* view and the user can navigate between different views by swiping. Figure 2.13(b) showcases the new camera view developed with a newer camera API. A much cleaner *Preview* view is demonstrated in Figure 2.13(c).



Tips ☑

Camera Assistant ☐

Cyan

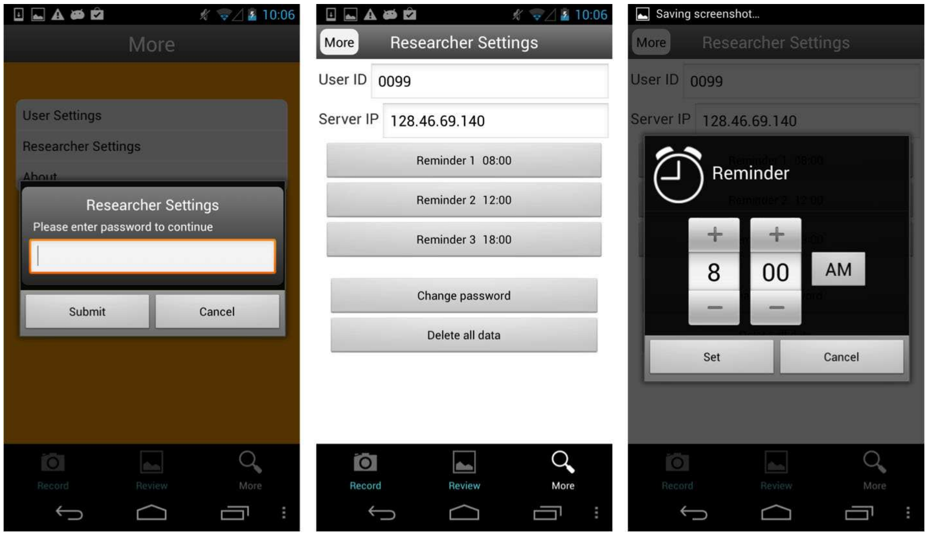
White

Orange

Green

Magenta

Fig. 2.11.: The interface for user settings.



Researcher Settings

Please enter password to continue

User ID 0099

Server IP 128.46.69.140

Reminder 1 08:00

Reminder 2 12:00

Reminder 3 18:00

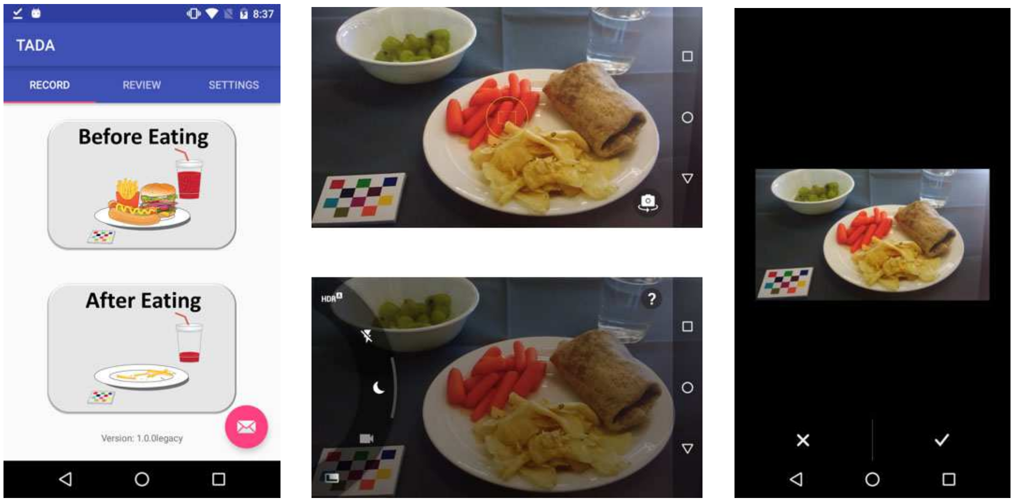
Reminder

8 00 AM

Fig. 2.12.: The interface for researcher settings.

2.5 User Studies

The TADA system has been tested, validated and used globally by researchers, dietitians and nutritionists for various purposes. We have more than 14 user studies involving more than 800 users who acquired more than 60,000 food images under controlled and community-dwelling conditions [47–49, 51, 52]. Most of our studies showed that not only did we reduce the burden of collecting daily dietary food records on users but we also allowed health care professionals to have real-time access to the records.



(a) (b) (c)

Fig. 2.13.: (a) Material design of the *Record* view. (b) Newer camera API. (c) Cleaner *Preview* view with faster image processing.

In [51] we studied the ability of adolescents aged 11-18 years to identify foods in images of their meals 10-14h postprandial and estimate portion size. We showed that the automated processes performed better at estimating portion size than the adolescents. The TADA system examined the consumption of razor clam [52] with respect to domoic acid consumption which can be toxic for humans. A study described in [49] recruited 135 volunteers (78 adolescents, 57 adults) to use the mFR for one or two meals under controlled conditions in order to evaluate the set of skills among adolescents and adults to use mFR and to compare their preference regarding to mFR. The results show that most of the users are able to easily use the mFR, while the adults were more likely than adolescents to remember to capture images and include all foods and beverages in their images, but they were less efficient than adolescents to capture a satisfactory image. In the Connecting Health and Technology (CHAT) study [47,48], 247 young adults aged 18-30 years participated in a 6-month study in a randomized controlled trial (RCT). The study aimed to evaluate the effectiveness of tailored feedback through the mFR to make dietary habit changes [47,48]. Result shows that tailored dietary feedback have an important effect on reducing sugar-sweetened beverages and energy dense nutrient poor foods such as fast food, as well as reducing body weight in those who were overweight.

2.6 Crowdsourcing System

In 2008, J. Howe [53] defined "crowdsourcing" as the act of making a task normally performed by a designated agent and outsourcing it to a large group of people. It was also referred to as the wisdom of the crowd [54] and J. Howe further discussed that the wisdom resides in how the crowd is used.

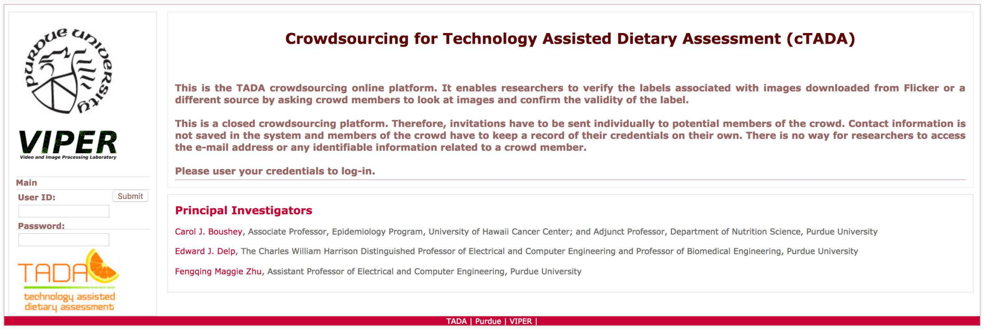
The Internet provides an ideal platform to distribute tasks and collect "wisdom" from the crowd. For example, the web-based crowdsourcing platforms, such as Amazons Mechanical Turk (MTurk), Freelancer, have shown their potentials in many applications [55].

One of the most intriguing examples by the successful crowdsourcing is an online game called "Foldit" [56] where players can design proteins on their computers in search of the lowest energy structure. The game has helped researchers to create an 18-fold-more-active enzyme [57].

In the field of dietary assessment, some researchers have tried to use the crowdsourcing techniques to directly estimate the energy intake from images [58,59]. PlateMate [59] is an end-to-end crowdsourcing system which uses MTurk to provide estimates of food intake and composition. The authors claimed the PlateMate system achieved similar accuracy as a trained dietitian based on their evaluations. In [58], a crowd was asked to provide a "healthiness" score based on food pictures acquired by a mobile application. The study also demonstrates that the untrained raters can be as effective as trained experts. However, it usually takes several hours for those crowdsourcing based systems to produce results, which might not be feasible for real-life applications.

As deep learning becomes popular, the demand for large-scale image datasets has risen. Many popular datasets, such as the ImageNet [60], the Common Objects in Context dataset(COCO) [61], were created with the help of crowdsourcing. Here, we describe the crowdsourcing system we have developed to contribute to building a large-scale food image dataset and providing more and richer groundtruth (structured labels, segmentation masks).

Figure 2.14 shows the web interface for users to access our system. The user IDs and temporary passwords are generated for the first-time user.



Crowdsourcing for Technology Assisted Dietary Assessment (cTADA)

This is the TADA crowdsourcing online platform. It enables researchers to verify the labels associated with images downloaded from Flicker or a different source by asking crowd members to look at images and confirm the validity of the label.

This is a closed crowdsourcing platform. Therefore, invitations have to be sent individually to potential members of the crowd. Contact information is not saved in the system and members of the crowd have to keep a record of their credentials on their own. There is no way for researchers to access the e-mail address or any identifiable information related to a crowd member.

Please user your credentials to log-in.

Principal Investigators

Carol J. Boushey, Associate Professor, Epidemiology Program, University of Hawaii Cancer Center; and Adjunct Professor, Department of Nutrition Science, Purdue University

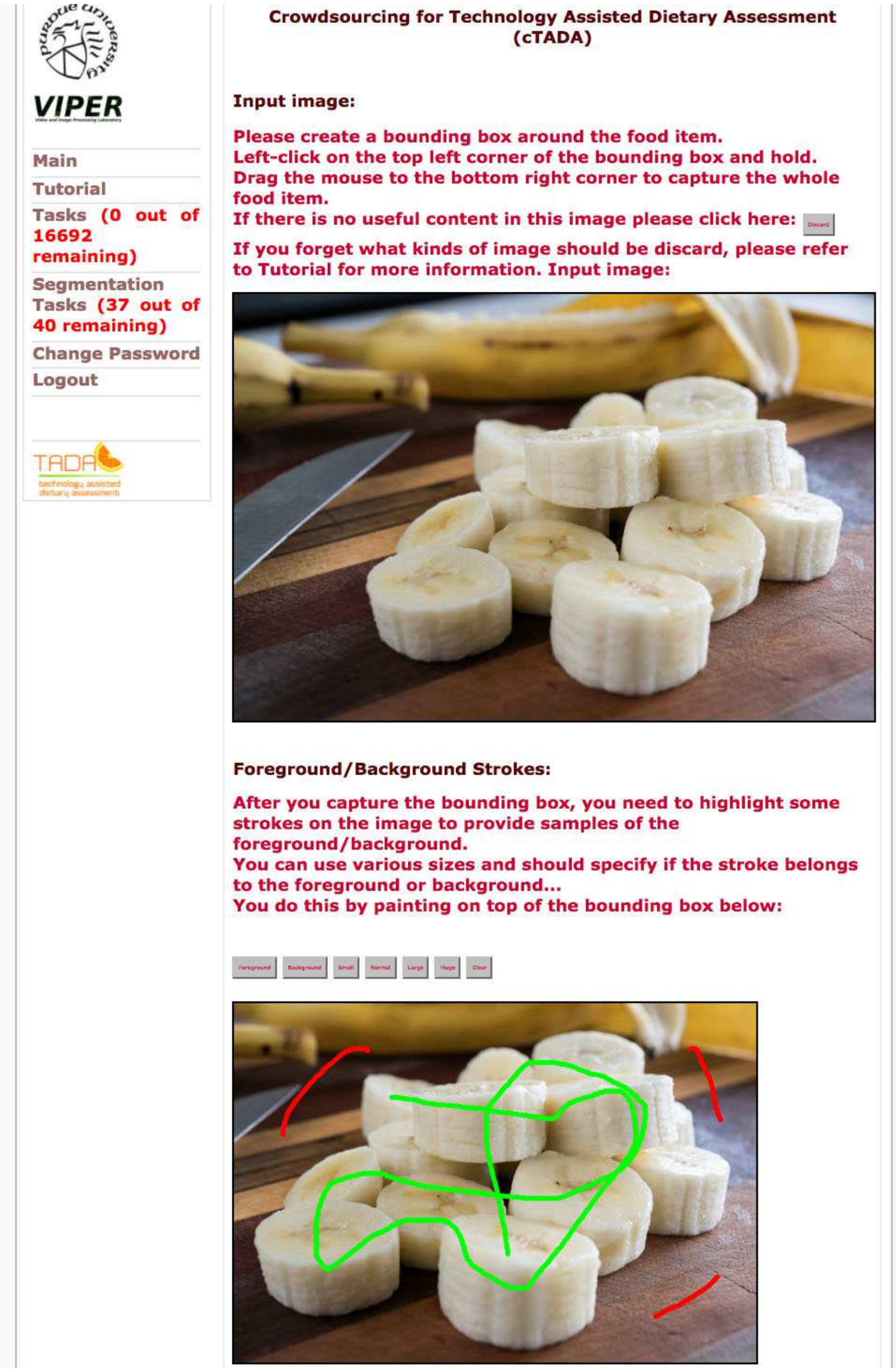
Edward J. Delp, The Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering, Purdue University

Fengqing Maggie Zhu, Assistant Professor of Electrical and Computer Engineering, Purdue University

Fig. 2.14.: CTADA crowdsourcing web interface.

Our crowdsourcing system has two main tasks. First, in the classification task, a user is presented with an image of a certain food category and the user needs to provide a binary answer (true or false), which indicates whether the displayed image and the label are associated.

Second, in the segmentation task, a user is asked to draw foreground/background strokes on an image and categorize the region of interest. As shown in Figure 2.15, the user should drag a bounding box on the original image and draw the strokes with the supplied tools. The bounding box and strokes are used for getting an initial segmentation mask. In Figure 2.16, two lists of food are available. The first list contains some coarser food groups like beverages and fruits. The second one serves as a finer categorization of the first option.


VIPER
Main
Tutorial
Tasks (0 out of 16692 remaining)
Segmentation Tasks (37 out of 40 remaining)
Change Password
Logout
TADA
technology assisted dietary assessment
Discard
Foreground
Background
Small
Normal
Large
Huge
Clear

Crowdsourcing for Technology Assisted Dietary Assessment (cTADA)

Input image:

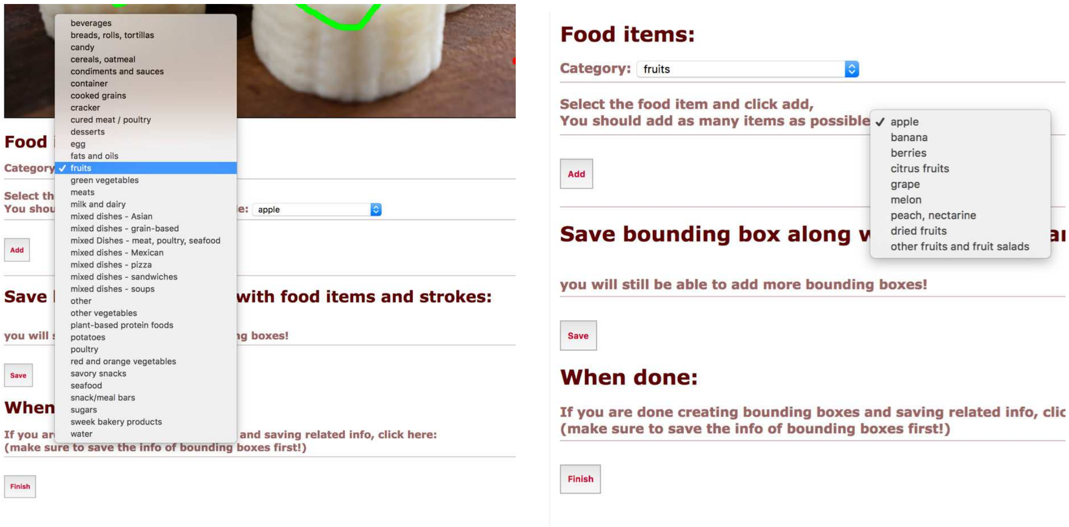
Please create a bounding box around the food item.
Left-click on the top left corner of the bounding box and hold.
Drag the mouse to the bottom right corner to capture the whole food item.
If there is no useful content in this image please click here:

If you forget what kinds of image should be discard, please refer to Tutorial for more information. Input image:

Foreground/Background Strokes:

After you capture the bounding box, you need to highlight some strokes on the image to provide samples of the foreground/background.
You can use various sizes and should specify if the stroke belongs to the foreground or background...
You do this by painting on top of the bounding box below:

Fig. 2.15.: CTADA segmentation task.



Food

Category

Select th
You shou

e: apple

Save | with food items and strokes:

you will s ng boxes!

When

If you ar and saving related info, click here:
(make sure to save the info of bounding boxes first!)

(a)

Food items:

Category: fruits

Select the food item and click add,
You should add as many items as possible

Save bounding box along w ai

you will still be able to add more bounding boxes!

When done:

If you are done creating bounding boxes and saving related info, clic
(make sure to save the info of bounding boxes first!)

(b)

Fig. 2.16.: (a) Main food categories. (b) Sub food categories.

3. MOBILE IMAGE BASED COLOR CORRECTION USING DEBLURRING

3.1 Color Correction

3.1.1 Related Work on Color Calibration

Dietary intake, the process of determining what someone eats during the course of a day, provides valuable insights for mounting intervention programs for prevention of many chronic diseases such as obesity and cancer. Accurate methods and tools to assess food and nutrient intake are essential for epidemiological and clinical research on the association between diet and health.

Color information is of great importance in our dietary assessment system as shown in Figure 2.1 and it serves as a key feature to identify foods [10, 11]. Thus, a consistent color descriptor of an object is critical. The colors of an object recorded by a camera depend mainly on three factors: illumination conditions in the scene (which are unknown in most cases), object intrinsic surface properties and various photometric parameters (e.g., exposure time, white balancing, gamma correction) [62]. A real world example is that the rendered colors of the same scene can be quite different even with the same camera from slightly different angles. Some approaches seek to overcome these problems by estimating illumination invariance color descriptors from training images, including the RGB histogram, color moments, and C-SIFT [63, 64]. In [63], a combined set of color descriptors with invariance properties surpass the performance of intensity based descriptors by 8% on category recognition.

An alternative approach to characterize the imaging properties is based on the spectral response/sensitivity of the camera. If the camera spectral sensitivity is known, then it is possible to estimate a relationship between the spectral sensitivity of the camera and the CIE color matching functions [65–68]. This approach, however, is not practical for common application because the spectral sensitivity of the camera should be measured by using specialized devices, such as monochromators, or radiance meters.

Our goal is to achieve color constancy under all kinds of lighting conditions so that the color of food can be used as a proper classification feature. When we look at an image acquired by a mobile phone camera, each pixel can be represented as a function f_i, where i is the color index (e.g. R, G, B). f_i is mainly dependent on three factors: the illuminant spectral power distribution $I(\lambda)$, the surface spectral reflectance $S(\lambda)$ and the sensor spectral sensitivities $V_i(\lambda)$.

$$f_i(x, y, S) = \int I(\lambda)S(\lambda)V_i(\lambda)d\lambda, i = R, G, B \tag{3.1}$$

The color sensor response form a vector $\boldsymbol{F(S)} = (f_R(S), f_G(S), f_B(S))$, which is also referred as the RGB tristimulus (R, G, B). Suppose that two images have been acquired from the same scene under different lighting conditions and cameras. For any pixel in the two images,

$$\begin{aligned} \mathbf{RGB}_1 &= \mathbf{F}_1(S) \\ \mathbf{RGB}_2 &= \mathbf{F}_2(S) \end{aligned} \tag{3.2}$$

Furthermore,

$$\mathbf{RGB}_1 = \mathbf{F}_1(S) = \mathbf{F}_1(\mathbf{F}_2^{-1}(\mathbf{RGB}_2)) \tag{3.3}$$

We would like to be able to express explicitly such transformation, $T = \mathbf{F}_1 * \mathbf{F}_2^{-1}(\cdot)$, between an unknown illuminant and a reference. Many chromatic adaptation techniques have been proposed to address this problem [69–71]. Since this is an ill-posed inverse problem most of the proposed solutions lacks uniqueness and stability. It has been shown that the universal best and the universal worst technique do not exist [72]: the method that performs best for a specific image depends on the image content.

There are generally two ways of achieving color correction. The first approach changes the overall colors in an image and is often used for colors other than neutrals to appear correct or pleasing. Methods for this type of correction are generally known as gray balance, neutral balance or white balance [73, 74]. Gray world is one of the most well-known gray balance methods [75, 76]. It is based on the assumption that given an image with sufficient amount of color variations, the average value of the R, G, and B components of the image should average to a common gray value. Another opponent technique is known as the white patch, which assumes that the maximum response in an image is caused by a perfect reflectance [77]. To combine both gray world and white patch approaches, Alessandro and Carlo Gatta proposed Automatic Color Equalization (ACE) in [78]. Their method extends the Retinex model of color equalization, merging Retinex with the Gray world and the White Patch equalization methods. Recently, the use of visual information automatically extracted from the images gas been investigated. Moreno et al. [79] obtained memory colors for three different objects (grass, snow and sky) using psychophysical experiments. They then used a supervised image segmentation method to detect memory color objects to color correct the image using a weighted Von Kries method. S. Bianco and R. Schettini [80] investigated color statistics extracted from faces in a scene to estimate illuminants. However, their results are largely based on the performance of the face detector and the knowledge of the corresponding skin color.

The second approach is usually referred to as color calibration uses the image of a reference chart for each set of acquisition conditions. Wang et al. [81] used a Munsell ColorChecker as the reference target. They then picked 13 color patches to train the parameters for the correction model. Adrian Ilie and Greg Welchin proposed a two-phase calibration technique in [82], where a 24-sample GretagMacbeth [83]ColorChecker was set up in each image acquired by different cameras. The two-phase method consists of an iterative closed-loop hardware calibration and software refinement, which is argued to ensure color constancy across multiple imaging devices.

From extensive studies the current TADA system has adopted the concept of using a reference target [7, 84]. Feedback from the participants in our studies indicated that it would be easy to use a credit card-sized fiducial marker due to the convenient

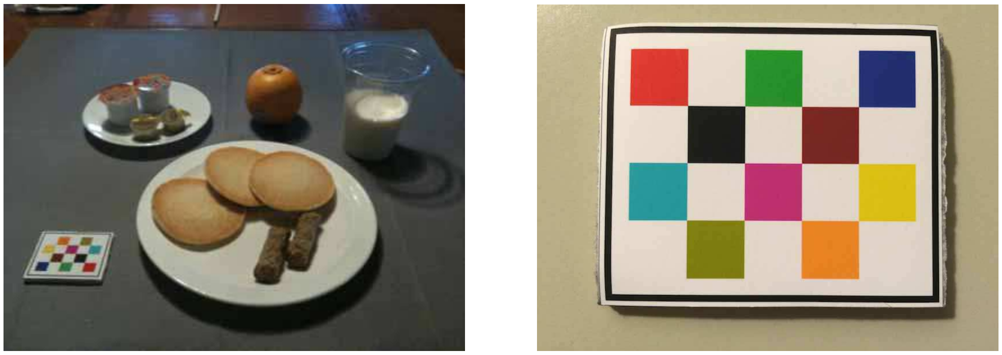


Fig. 3.1.: An example of the color fiducial marker used in the TADA System.

incorporation into their current lifestyles [5, 51]. Thus, we decided to use a compact checkerboard pattern to for color calibration (see Figure 3.1). The color checkerboard was designed to have the dimension of $7 \times 6\,cm^2$. The color patches were chosen to cover the full color spectrum.

3.1.2 Proposed Method

In this section, we describe the selection of the color space we use for color correction and then compare our method to our previous work [7]. Color correction is done on the backend server, which mainly consists of color extraction from the checkerboard and color mapping to the D65 reference and matching the acquired image to the reference lighting condition [7]. In this chapter, we investigate polynomial models using three different color spaces and compare them to our previous work.

3.1.3 Color Space Models

A color space is a mathematical model used to describe how colors can be interpreted as tuples, typically of three or four elements. sRGB color space is commonly used in mobile cameras and displays [62]. The transformation between sRGB color space and linear RGB space is defined as follows:

$$C_{sRGB} = \begin{cases} 12.92C_{linear}, C_{linear} \leq 0.0031308 \\ 1.055{C_{linear}}^{1/\gamma} - 0.055, C_{linear} > 0.0031308 \end{cases}$$

where C represents R,G or B channel and γ is the gamma correction value. In our previous work [7], the checkerboard image captured using a mobile telephone camera under the D65 illumination was used as the reference. We implemented a different approach for measuring the color patch. The checkerboard shown in Figure 3.1 is placed inside a "SpectraLight II" illumination booth and we use a spectral radiometer to measure each color patch on the checkerboard. The output of spectral radiometer is in XYZ color space. The conversion from linear RGB color space to CIEXYZ color space is defined as [62],

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix}$$

The inverse conversion from CIEXYZ to linear RGB color space is,

$$\begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} = M^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Here, we propose to use LMS color space for color correction. LMS color space is derived from the human visual system. Humans have three distinct types of color receptors, which are referred to as *long*,*medium* and *short* cones [85]. Though the LMS color space is not commonly used in color specification, it is often used for chromatic adaptation. It has simple and positive color matching function for each channel. It is computationally simpler compared to other nonlinear color spaces, such as the

CIELAB color space. It is also proportional to the illuminant energy. The coordinates in the XYZ system are related to LMS through the following transformation [86],

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.4002 & 0.7076 & -0.0808 \\ -0.2263 & 1.1653 & 0.0457 \\ 0 & 0 & 0.9182 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.4}$$

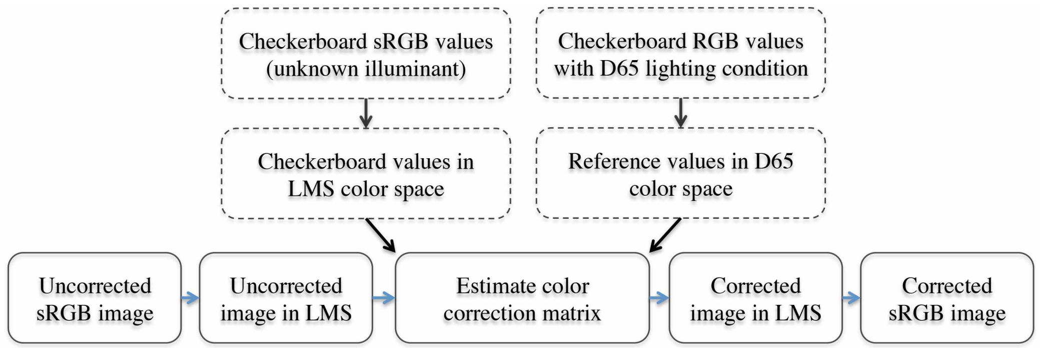

Checkerboard sRGB values (unknown illuminant)
Checkerboard RGB values with D65 lighting condition
Checkerboard values in LMS color space
Reference values in D65 color space
Uncorrected sRGB image
Uncorrected image in LMS
Estimate color correction matrix
Corrected image in LMS
Corrected sRGB image

Fig. 3.2.: Diagram of our proposed color correction method.

Figure 3.2 shows a diagram of our proposed color correction method. The uncorrected image is first converted into LMS color space. Then, we implement and optimize the polynomial transforms to find the correction matrix. For each color patch, the tristimulus values in LMS color space can be represented as a vector $V : (L_i, M_i, S_i)^T (i = 1, 2, \ldots, 11)$, we have 11 colors on the checkerboard including black and white. Similarly, the reference checkerboard has 11 corresponding color values, denoted as $R : (RL_i, RM_i, RS_i)^T (i = 1, 2, \ldots, 11)$. We use the following vector $X : [L, M, S, LM, LS, MS, 1]^T$ to estimate the color correction matrix. The transformation model can be represented as,

$$\begin{cases} CL_i = a_{11}L_i + a_{12}M_i + a_{13}S_i + a_{14}LM_i + a_{15}LS_i + a_{16}MS_i + a_{17} \\ CM_i = a_{21}L_i + a_{22}M_i + a_{23}S_i + a_{24}LM_i + a_{25}LS_i + a_{26}MS_i + a_{27} \\ CS_i = a_{31}L_i + a_{32}M_i + a_{33}S_i + a_{34}LM_i + a_{35}LS_i + a_{36}MS_i + a_{37} \end{cases}$$

where CL_i,CM_i and CS_i are the corrected tristimulus. The equation can also be rewritten in matrix form as,

$$\begin{bmatrix} CL \\ CM \\ CS \end{bmatrix} = A^T X = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \\ a_{14} & a_{24} & a_{34} \\ a_{15} & a_{25} & a_{35} \\ a_{16} & a_{26} & a_{36} \\ a_{17} & a_{27} & a_{37} \end{bmatrix}^T \begin{bmatrix} L \\ M \\ S \\ LM \\ LS \\ MS \\ 1 \end{bmatrix} \tag{3.5}$$

where A is the color correction matrix and X is the polynomial combination. Thus, we want to find a matrix A, which minimizes the overall error between the corrected image and the reference across all 11 color patches. We further formulate this problem as finding constrained least square solution. By using the notation above, we have

$$A = \arg\min_X \sum_{i=1}^{11} \|AX - R\|^2 \tag{3.6}$$

Equation 3.6 can be solved using Levenberg-Marquardt methods [87]. Finally, we correct the image in the LMS color space using A and convert it back to the sRGB space for display.

3.1.4 Experimental Results

To evaluate the performance of the proposed color correction methods, we use GretagMacbeth Colorchecker [88] as the testing target. Both the TADA fiducial marker and GretagMacbeth Colorchecker were placed inside a SpectraLight II illumination booth. We acquired several images of the two checkerboards using four different illuminants, i.e. simulated daylight (CIE D65, 6500 K), horizon daylight (simulated early morning sunrise or afternoon sunset, 2300 K), CIE A (incandescent home lighting, 2856 K), and commercial fluorescent (cool white, 4000 K). All the images were acquired using iPhone 5 camera.

The captured images with non-D65 illuminations were corrected using the method described in Section 3.1.3. Here, we compared our proposed method with the similar approach using the CIELAB and sRGB color spaces. Let $(R_i, G_i, B_i)(i = 1, 2, \ldots, 24)$ denote the reference color of each patch in the GretagMacbeth Colorchecker and $(R_i^c, G_i^c, B_i^c)(i = 1, 2, \ldots, 24)$ be the corrected values of corresponding patch under various lighting conditions. The average Euclidean distance for all 24 pairs is defined as:

$$\Delta = \frac{1}{24}\sum_{i=1}^{24}\left\|(R_i^c, G_i^c, B_i^c)^T - (R_i, G_i, B_i)^T\right\| \tag{3.7}$$

Table 3.1 shows the mean error between the reference and corrected images for different methods. The entry $Total$ in the table is simply the summation of R, G and B channel errors. The column of LMS demonstrates the error of the method we proposed and the results from the similar technique using sRGB and CIELAB color space verified our choice of choose LMS as the color correction space. Even though our method does not produce consistently the smallest error for each channel, the overall RGB error is approximately a 10% improvement compared with the correction method using CIELAB color space. This implies that even though the CIELAB color space is uniform with respect to the Human Visual System (HSV), it is not necessarily the best choice when it comes to the linear color correction model, since the model expects each channel to be correlated when combining the polynomial terms. In our experiment, we used the fixed gamma value of 2.2 based on te iPhone 5's camera specification. By examining some of the most popular smart phones on the market, we concluded that such gamma is plausible (see Table 3.2).

3.2 Single Image Deblurring

3.2.1 Blind Deconvolution Based Deblurring

In our previous work [7], an image quality measurement method as well as a non-linear color correction model using the CIELAB color space was proposed. However, an underlying problem persists that a user might still send a blurry image to our

Table 3.1.: Errors (Δ) between the reference image and the corrected images

Lighting	Error	LAB	sRGB	LMS
Incandescent	Red	7.98	7.44	7.76
	Green	9.54	8.86	7.59
	Blue	10.56	9.58	7.56
	Total	28.08	25.88	22.91
Horizon Light	Red	7.53	6.34	3.30
	Green	3.84	3.98	3.85
	Blue	11.85	11.55	9.37
	Total	23.22	21.87	16.52
Coolwhite	Red	3.54	3.44	3.22
	Green	4.18	4.15	4.31
	Blue	3.88	3.65	2.89
	Total	11.60	11.24	10.52

Table 3.2.: Gamma correction values of popular smart phones

Mobile Phones	iPhone 6	iPhone 5	iPhone 4s	Galaxy S5	Galaxy S4
Gamma	2.23	2.22	2.1	2.25	2.16

image analysis system even our image quality “checker” suggests retaking the image. There are cases where the user may be reluctant to retake the image of his/her meal or the image on the small mobile telephone screen appears to be good enough. Often, the checkerboard in a blurred image cannot be correctly detected and consequently color correction will be skipped by our system. This imposes a critical problem for the image analysis steps, i.e. food segmentation and identification. This then becomes a Blind Deconvolution (BD) problem with the unknown blur represented as a Point Spread Function (PSF).

Blind deconvolution is the process of recovering a sharp version of a blurry image. It is also well known to be ill-posed, small perturbations of the data produce large deviations in the resulting solution [89]. Mathematically, the general model for a linear degradation caused by blurring and additive noise is given by

$$y = h \otimes x + n \tag{3.8}$$

where x is a visually sharp image or original image, n is known noise and h is a nonnegative blur kernel, whose support is small compared to the image size. When the noise is ignorable, the objective of blind restoration is to estimate x and h. Often, the model above is also represented in terms of a matrix formulation, that is,

$$\vec{y} = \mathbf{H}\vec{x} \tag{3.9}$$

where the vectors $\vec{x}$ and $\vec{y}$ represent the original image and the observed image respectively by stacking the image matrix into a vector. **H** is a Block Toeplitz with Toeplitz Blocks (BTTB) matrix.

Single-image motion deblurring have been extensively studied in the past few years and achieved a few milestones [90–93]. The effective techniques that extended naive maximum a posterior (MAP) inference were broadly adopted in many applications [94, 95].

Recent methods have characterized x using natural image statistics [96–99]. These techniques exhibit some common principles. A. Levin et al. [100] argued the failure of the MAP approach and suggested that the key component making blind deconvolution possible is not the choice of the prior, but the estimator. In [101], D. Krishnan et al. pointed out that there is a major drawback in many common forms of image priors because the minimum of the resulting cost function does not correspond to the true sharp solution. They proposed a new image regulation method using the ratio of the l_1 norm to the l_2 norm on the high frequencies of an image. MAP-based methods can be categorized in two groups: methods with explicit edge prediction [102] and the ones with implicit regularization to noises [101, 103]. For example, Shan et al. [103]

proposed to use a large regularization weight to suppress insignificant structures and adopt a sparse image prior, which results in a crisp-edge image. This method is useful to remove detrimental image structure, guiding kernel estimation in a good direction. Krishnan et al. [101] used an L_1/L_2 regularization in the optimization step.

Based on Krishnan's approach [101], we introduce an image deblurring scheme using a saliency map. The idea behind using visual saliency is that we want to reduce the processing time by analyzing a sub-image. The sub-image should contain enough features to estimate the blur kernel. In our application, it is plausible to assume that the blur in a image is uniform and only comes from slight camera movement, such as camera shift or in-plane rotation. There are mainly two reasons for such assumption. First, we have implemented an image quality check on the mobile telephone, which should prevent users from taking blurry images or images without the fiducial marker present. Second, in our image dataset, most of the food images are acquired in a stationary environment with reasonably adequate lighting condition and the fiducial marker is always detected as a salient region, if present. Since the checkerboard region contains plenty of corner or edge features as well as a wide range of colors, the estimated blur kernel is consistent to what is analyzed from the entire image. The results show that our saliency based image deblurring is robust and fast.

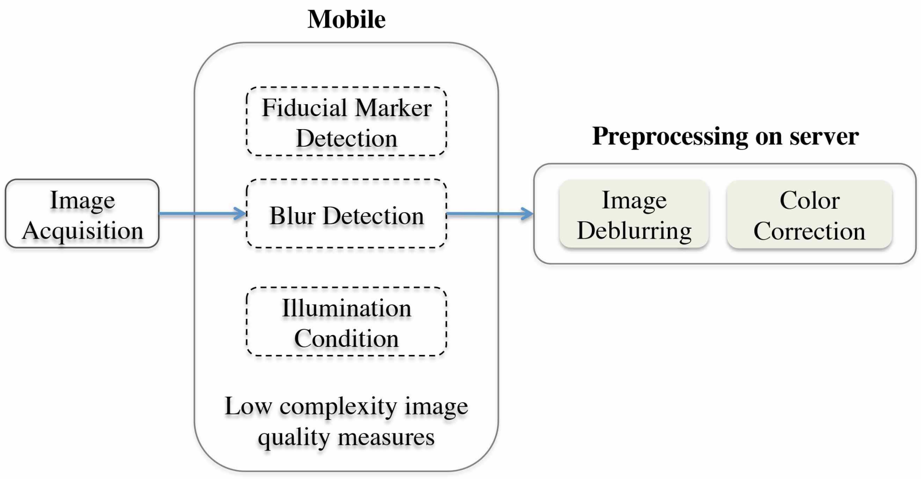


Fig. 3.3.: TADA mobile quality measure system.

3.2.2 Proposed Method

In this section, we explain our proposed deblurring technique in detail. Figure 3.3 illustrates the workflow from mobile quality measure to image preprocessing on the server. The user first takes a food image that contains the color fiducial marker under an unknown illumination. Then, several image quality checks are initiated before the user can send the food image to our backend server. The examination includes checkerboard detection [6], blur detection and a coarse illumination condition check. Due to the limited computational resources and the need for quick feedback on the mobile device, simple processing approaches are used. If the image does not pass the blur detection on the mobile phone, image deblurring will be triggered. Both image deblurring and color correction are implemented on the backend server to complete the preprocessing. We want to restore the blurry images to the maximum extent so that they can be color corrected for further analysis. As shown in Equation 3.8, we observe the resulting blurry image y and the goal is to recover the unknown sharp image x as well as the blur kernel h. Based on the deblurring technique proposed in [101], we introduce a faster and robust deblurring method using visual saliency. Our method dramatically reduces the computational time without sacrificing restoration quality. As shown in Figure 3.4, the proposed deblurring method consists of four steps. Given an input blurry image, the saliency map of the image is first computed. A saliency map is a multi-scale feature map which contains local spatial discontinuities in the modalities of color, intensity and orientation. We adopted the idea of image signature for faster saliency detection, which was first introduced by X. Hou etc. in [104]. If we denote the grayscale blurry image as x, which is the mixture of foreground and background, its image signature is defined as

$$ImageSignature = sign(DCT(x)) \tag{3.10}$$

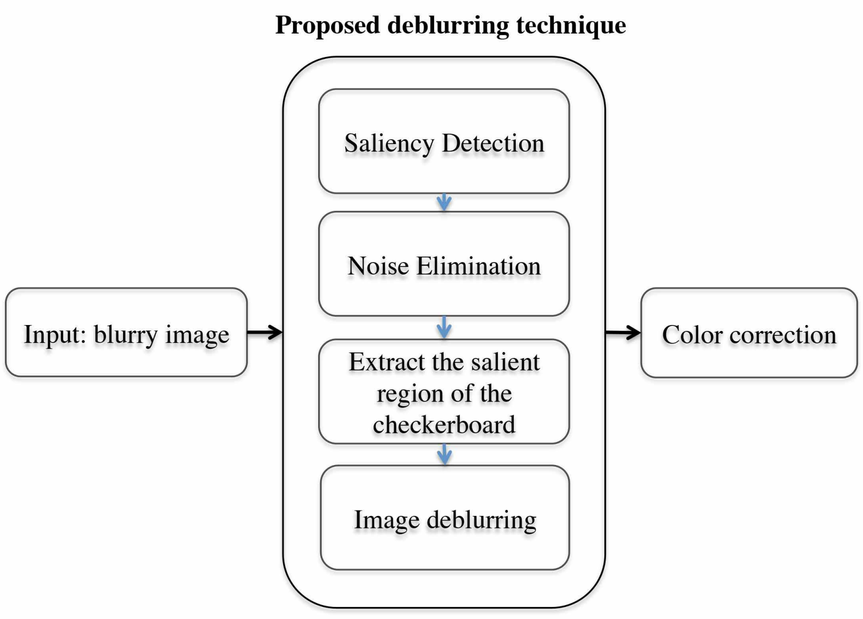


Fig. 3.4.: Proposed image deblurring technique.

where DCT represents Discrete Cosine Transformation. Consider the reconstructed image $\tilde{x} = IDCT[ImageSignature]$, the saliency map s is computed by smoothing the squared reconstructed image $\tilde{x}$,

$$s = g * (\tilde{x} \circ \tilde{x})$$

where g is a Gaussian kernel, $*$ is convolution symbol and $\circ$ represents entry-wise product operator. Then we use a Flood Fill technique [105] to eliminate the noisy saliency regions, especially small blobs. If we consider a sub-image containing the checkerboard, it possesses many of visually recognizable features, such as corners, edges and contrast color patches, so it almost always is detected as a salient region. However, it is likely that more than one region will be detected. Figure 3.5 shows two examples of the initial saliency map. Our goal is to extract the area containing the checkerboard and use that to estimate the blur kernel. This can be achieved by analyzing the histogram of each salient region. For one salient region, the histogram of each channel is split into 8 bins, which can be represented by an 8 dimensional vector. The element with the largest value is discarded and 7 other elements are

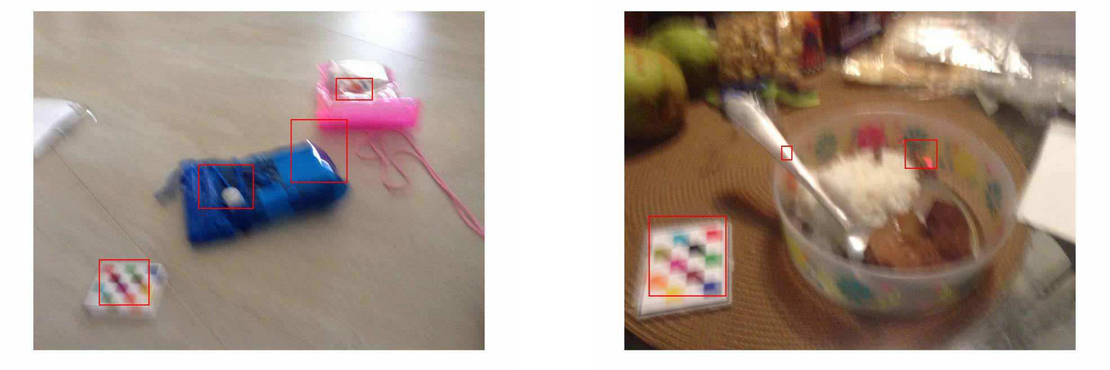


Fig. 3.5.: Examples of saliency regions.

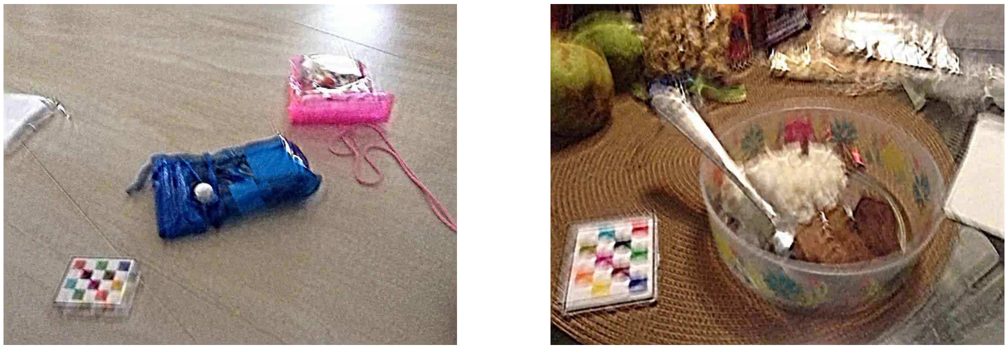
then normalized. By combining 3 channels, we get a 21 dimensional feature vector. The cross-correlation between such feature vector and the reference checkerboard histogram is computed to find the optimal match.

After we extract the checkerboard region, we use Krishnan's approach [101] to estimate the blur kernel. Krishnan's method is described as follows:

1. Use derivative high-pass filters on the blurry image y, creating a high-frequency image g
2. Blind multi-scale estimation of blur matrix h from g using a coarse-to-fine pyramid of image resolutions. At each scale, update sharp high-frequency image g and h using l_1/l_2 regularization. Use bilinear interpolation to up-sample the current kernel to finer level as initialization.
3. Image recovery using non-blind algorithm [106].

3.2.3 Experimental Results

To evaluate our visual saliency based image deblurring technique, we manually choose 25 blurry images from the TADA free-living study. The free-living study contains a number of 315 meal images, which were acquired under natural eating conditions by 11 participants. Two examples are shown in Figure 3.5. All food



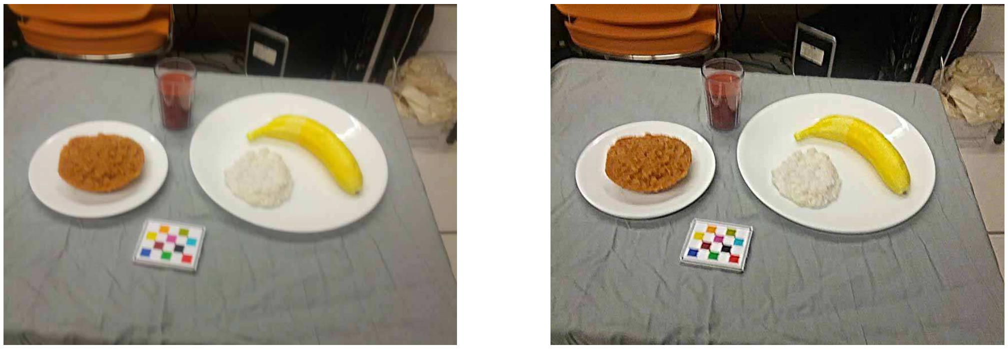
(a) Checkerboard can be detected.

(b) Checkerboard cannot be detected.

Fig. 3.6.: Examples of deblurred images.

images acquired in this study were taken under natural eating conditions by our participants. We also acquired another 25 images of plastic food using Samsung Galaxy Nexus. When acquiring those images, we tried to simulate the real life situation by deliberately moving the camera slightly to create blur effect. Now, we have a total of 50 images as testing data. The TADA checkerboard was included in all the images, but none could be detected due to blurriness. After applying our method to the testing images, 31 out of 50 were correctly detected to have the TADA checkerboard. Therefore, color correction can be applied to them.

Figure 3.6 illustrates two deblurred images corresponding to the original ones in Figure 3.5. The one on the left was detected to have the TADA checkerboard after image deblurring, even though it seems visually unpleasant in Figure 3.6. The image on the right failed to be detected with a checkerboard pattern due to the heavy rotation and shift of the camera. A plastic food image example is shown in Figure 3.7. The fiducial marker was correctly located in the deblurred image in this case. In addition to the robust image restoration, our method achieved approximately $\frac{1}{6}$ of computational time compared to Krishnan's algorithm without using saliency map. As for the two cases in Figure 3.6(from left to right), Krishnan's algorithm took 51.45s and 74.42s respectively and the proposed method consumed 8.13s and 11.51s including saliency detection as well as deblurring. All the experiments were conducted on OSX Yosemite with 2.6G quad core i7 CPU and 16G RAM. The images were scaled to 800×600 to speed up the process.



(a) The original image.

(b) The deblurred image.

Fig. 3.7.: Example of blurry and deblurred plastic food images.

4. EFFICIENT SUPERPIXEL BASED IMAGE SEGMENTATION

4.1 Overview of Image Segmentation Methods

The accurate estimate of energy and nutrients consumed using food image analysis is essentially dependent on the correctly labeled food item and a sufficiently well-segmented region. Food labeling primarily relies on the correctness of interest region detection, which makes food segmentation extremely crucial. Although the human vision system can group pixels of an image into meaningful objects without knowing what objects are present, effective object segmentation from an image is in general a highly unconstrained problem [10].

Image segmentation is a process of partitioning an image into several disjoint and coherent regions in terms of some desired features. Segmentation methods can be classified into three major categories, i.e. region grouping methods [107,108], contour to region methods [109–111], and graph-based methods [112,113].

Alpert et. al [108] proposed a bottom-up probabilistic framework using multiple cue integration. It showed promising results when compared to other approaches. An example of a contour to region method is the hierarchical segmentation introduced in [110]. This has lead to several methods based the global probability of boundary (gPb), Ultrametric Contour Map (UCM) [110], and then methods that transform a contour into a hierarchy of regions while preserving contour quality [114]. Donoser et. al [111] proposed to locally predict oriented gradient signals by analyzing mid-level patches to reduce computational time of UCM. B. Catanzaro et. al [115] tried to address the UCM computational issue using parallel computing with GPUs.

Graph-based approaches can be regarded as image perceptual grouping and organization methods based on the integration of multiple features along with the spatial information. The common theme for graph-based approaches is the construction of a weighted graph where each vertex corresponds to a pixel or a region of the image. The weight of each edge connecting two pixels or two regions represents the likelihood that they belong to the same segment. A graph is partitioned into multiple components that minimize some cost functions. There are mainly two types of graph-based approaches: *merging* and *splitting*. The efficient graph based method, also known as the local variation (LV) proposed by Pedro et al [112] is a *merging* method. It is an efficient algorithm concerning computation complexity. Among *splitting* methods, normalized cuts (Ncut) [113] is extensively used.

Due to the complexity of food images (e.g. occlusion and cluttered background), food image segmentation is a difficult task. In [23], researchers experimented with 10 kinds of food with containers. Thus the formable part model (DPM) and a circle detector were used to constrain the food region of interest. Bettadapura et. al [29] used hierarchical segmentation in their implementation. A semantic segmentation method based on a deep neural network was recently proposed in [35]. Such method requires a huge amount of manually segmented food images.

Food image segmentation has been extensively addressed in our previous work resulting in a joint segmentation/classification multiple-hypothesis technique [10]. We also investigated the use of local variation and integrated it with food classifiers so that we can iteratively use the classification results to refine the segmented regions [116]. For each segment, a set of color, texture and local region features are used for our learning network, including the use of k-Nearest Neighbor (KNN), vocabulary tree and Support Vector Machine (SVM) [117]. However, it is difficult to prove that the refined segment using classification results will produce higher confidence score.

In the following sections, a simple yet effective segmentation method that integrates normalized cut and superpixels using multiple features is proposed. This method is different than our previous work in that we do not use a feedback approach and we construct a new graph model. The main contributions of this chapter are summarized as follows: (1) we introduce an efficient way of constructing weighted graph based on superpixels in multiple feature spaces, (2) we proposed a new image segmentation evaluation method which is specifically suitable for multi-food images , (3) we evaluate the proposed method on both the publicly available Berkeley Segmentation Database and our own food dataset. We are able to achieve competitive results, especially for food segmentation.

4.2 Normalized Cut on Superpixels

Graph-based image segmentation techniques generally represent the problem in terms of a graph $G = (V, E)$ where each node $vi \in V$ corresponds to a pixel in the image, and the edges in E connect certain pairs of neighboring pixels. A weight is associated with each edge based on some property of the pixels that it connects, such as their image intensities. Depending on the method, there may or may not be an edge connecting each pair of vertices. A graph partitioning method attempts to organize nodes into groups such that the intra-group similarity is high and the inter-group similarity is low. A *Cut* which partitions the graph or subgraph into two disjoint sets A and $B = V - A$ is sometimes defined as a total weight of the removed edges:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

The problem of finding the minimum cut has been extensively studied. However, minimum cut introduces a bias towards small sets of isolated nodes. To address the bias, Shi and Malik [113] proposed the normalized cut and developed approximation methods for computing the NP-hard problem. Instead of looking at the value of total edge weight connecting the two partitions, Ncut computes the cut cost as a fraction of the total edge connections to all the nodes in the graph,

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in A to all nodes in the graph and similarly for $assoc(B, V)$. Some properties of using Ncut in the TADA system are studied in our previous work [10,117–119]. We compare Ncut with Local Variation method and active contour methods and choose Local Variation over the others for it gives us competitive results and has fast running time.

General graph-based segmentation methods use low level features to measure similarity between two sets of pixels. For example, Ncut uses pixel intensity and difference of oriented Gaussian filter [113]. For an image of a complicated scene, using low level features often result in noisy segments. Another drawback with Ncut is the increase in computation associated with increased image size. "Superpixels" often are referred as local groups of pixels which have similar characteristics [120]. Superpixel methods have several useful properties such as they usually align well with object contours if the objects are not too blurry or the background is not too cluttered. They also enforce local smoothness because pixels belong to a superpixel are often from the same object.

To address the above disadvantages using Ncut, intuitively we might want to combine Ncut with superpixels. We can obtain higher level features from superpixels and reduce the size of the affinity matrix in Ncut. We shall call this approach 'SNcut.' We will discuss below the challenges of doing this including superpixel denoising, graph formation and food segmentation evaluation.

We use the simple linear iterative clustering (SLIC) method [121] to get an initial superpixel segmentation in SNcut. SLIC is a fast and memory efficient method that address local clusterings of pixels in 5D space consisting of L, a, b from the CIELAB color space and x, y pixel coordinates. It has shown good performance for several popular segmentation datasets [121].

Table 4.1.: SLIC algorithm

Efficient superpixel segmentation

```
1. Initialize cluster centers by sampling pixels at regular grid steps S
2. Perturb cluster centers in a $n \times n$ neighborhood to
   the lowest gradient position
3. repeat
4.   for each cluster center do
5.     Assign the best matching pixels from a $2S \times 2S$ square neighborhood
       around the cluster center according to the distance measure $D_s$
6.   end for
7.   Compute new cluster centers and residual error E
8. until $E \leq threshold$
9. Enforce connectivity
```

As described in [121], regular grid interval S is defined as $\sqrt{N/K}$, where N is the number of pixels and K is the desired number of approximately equally-sized superpixels. The distance measure D_s is defined as follows:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{4.1}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{4.2}$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \tag{4.3}$$

where i, k index pixels and m is the customizable parameter.

When SLIC tries to preserve the shape or boundary of objects, there is a tradeoff between the similarity measure in the LAB color space and spatial distance between pixels, also regarded as proximity. This sometimes results in small segmented regions scattering around the “true segment”. We propose to use a Gaussian filter with a variance, σ, on the original image and use a median filter [122] with a fixed radius in pixel, r, following SLIC to merge the noisy segments. We then construct a weighted graph by using the proximity and similarity in color and texture features within the superpixel neighborhood, which will be discussed in Section 4.3.

4.3 Color and Texture Cues

When constructing the weighted graph of superpixels we only allow adjacent superpixels to be connected which ensures a local Markov relationship [123]. We use average RGB pixel values and a customized local binary pattern (LBP) [124], discussed below, as the color and texture cues for the superpixels. The average RGB pixel value is obtained for all the pixels in the corresponding superpixel. LBP is a particular case of the Texture Spectrum model proposed in [125]. The customized LBP feature vector is created in the following manner: (1) Examine each pixel that has a 3×3 neighborhood in the superpixel with its 8 neighbors. If the center pixel value in L channel is greater than its neighbor, the output is 1, otherwise 0. Following a clockwise order, concatenate the binary results in an 8-dimensional vector, (3) By using the resulting vector as a binary number, we can convert it to a decimal number and then construct a histogram for each superpixel, (4) Finally, the histogram is sampled into 9 bins and normalized.

To translate the color and texture features into edge weights in our graph model, we first use χ^2 test [126] in texture space and L_2 distance in color space to obtain similarity measures. The χ^2 distance between two LBP vectors, g and h, associated with two superpixels, S_g, S_h is defined as:

$$D_{texture} = \chi^2(g,h) = \frac{1}{2}\sum_i \frac{(g(i)-h(i))^2}{g(i)+h(i)} \tag{4.4}$$

L_2 measure in the RGB space is denoted as D_{color}. We then map the similarity measurement to a probability estimate:

$$P_{texture} = e^{-\frac{D_{texture}}{\sigma_{texture}}},\ P_{color} = e^{-\frac{D^2_{color}}{\sigma_{color}}} \tag{4.5}$$

Based on the experiments, we set σ_{color} to be 255 and $\sigma_{texture} = \beta D_{texture}$, where $\beta \in [8, 12]$. Finally, the edge weight is obtained,

$$\begin{aligned} w_{g,h} =& \mathbf{I}(P_{color}, P_{texture}) \max\{P_{color}, P_{texture}\}+ \\ & (1-\mathbf{I}(P_{color}, P_{texture})) \min\{P_{color}, P_{texture}\} \end{aligned} \tag{4.6}$$

where $\mathbf{I}(\cdot)$ represents the indicator function:

$$\mathbf{I}(P_{color}, P_{texture}) = \begin{cases} 1, & P_{color} > \epsilon_1, P_{texture} > \epsilon_2 \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

4.4 Object Based Segmentation Evaluation

Existing segmentation evaluation metrics include region differencing assessments [127–129] which count the degree of overlapping between segmented regions, boundary matching [130] and [131], Variation of Information (VoI) [132] and non-parametric tests such as Cohen Kappa [133] and normalized Probabilistic Rand Index (PRI) [134].

Precision and recall [130, 131] are widely used in boundary-based segmentation evaluations. However, some types of error, such as an under-segmented region that overlaps with two ground-truth segments with only a few missing boundary pixels in between, cannot be detected by boundary-based evaluation. We believe for food image segmentation a precise segment of each food is preferred over partially accurate boundaries. We adopt the region-based precision and recall proposed in [127] and adjust several criteria to focus on the objects of interest, or foods in our case.

Let $S = \{S_1, S_2, \ldots, S_M\}$ be a set of segments in an image generated by a segmentation method and $G = \{G_1, G_2, \ldots, G_N\}$ a set of ground-truth segments of all foreground objects in the same image, where S_i $(i = 1, \ldots, M)$ and G_j $(j = 1, \ldots N)$ represent the individual segment respectively. M is the total number of segments generated by the segmentation method and N is the total numbers of segments of the ground-truth.

The precision P for segment S_i and ground-truth G_j can be defined as $P(S_i, G_j) = \frac{|S_i \cap G_j|}{|S_i|}$, where $|A|$ is the total number of pixels in A. Similarly, the recall R for S_i and G_j can be defined as $R(S_i, G_j) = \frac{|S_i \cap G_j|}{|G_j|}$. The F-measure, F [135] can then be estimated using precision and recall:

$$F(S_i, G_j) = \frac{1}{\alpha \frac{1}{P(S_i,G_j)} + (1-\alpha) \frac{1}{R(S_i,G_j)}}. \tag{4.8}$$

where α is the weight. We do not have a preference between precision and recall. Hence we set $\alpha = 0.5$.

Furthermore, we want to know which objects in the image are correctly segmented and how accurate are those segments. It is likely that $M \neq N$, hence we want to match each foreground segment S_i to a G_j where S_i and G_j represent the same object in the image. In order to fairly match the segments, we introduce the overlap score (O) [128], $O(S_i, G_j) = \frac{|S_i \cap G_j|}{|S_i \cup G_j|}$. Unlike precision and recall, the overlap score is low for both over-segmentation and under-segmentation, but only high when the segmentation is precise and accurate. The evaluation is then done as follows:

1. The background ground-truth segment G_B can be found as $G_B = 1 - \bigcup_{j=1}^{N} G_j$.

2. For each segment S_i in S, $P(S_i, G_B)$ and $P(S_i, G_j)$ with all the G_j in G are estimated. If $P(S_i, G_B)$ is higher than all the $P(S_i, G_j)$ for all G_j, we consider (S_i) as a background segment.

3. For each remaining S_i, $O(S_i, G_j)$ is estimated with all the G_j in G, and (S_i, G_j) is considered as a matched pair if it has the highest overlap score.

4. Let $I_j = \{S_i$: (S_i, G_j) is a matched pair, $i \in [1, \ldots, M]\}$ and $l(I_j)$ be the number of elements in I_j that matches G_j, the precision for this image can be estimated as:

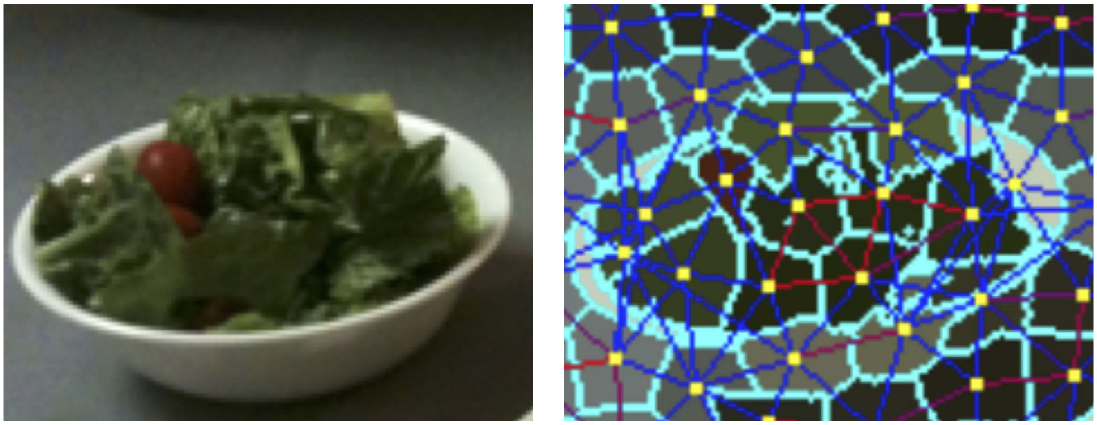
$$P = \frac{1}{M} \sum_{\substack{j=1 \\ l(I_j) \neq 0}}^{M} \frac{1}{l(I_j)} \sum_{S_i \in I_j} P(S_i, G_j) \tag{4.9}$$

The Recall and Overlap score can be estimated in a similar way.

The above conveys our ideas of "good food image segmentation": (1) Compared to the natural image segmentation criteria, we do not care much about the background extraction because some simple filters are able to eliminate over-segmented background [116]. Classifiers are often more effective in differentiating background from foods. (2) We favor a precise food segment and punish over-segmentation by averaging all suitable matches because classifying many small segments are time-consuming and often generates unreliable results.

4.5 Experimental Results

In this section, we evaluate the proposed image segmentation method on the TADA food segmentation dataset (TSDS) and the Berkeley Segmentation Dataset (BSDS) [136]. The TSDS contains 200 hand-segmented food images, which are collected from a larger dataset including a total of 1453 images of 56 commonly eaten food items acquired by 45 users in natural eating conditions. Each image contains 5 different food items and 3 utensil items on average. The BSDS consists of 500 natural images of diverse scene categories. Each image is manually segmented by 5 human subjects on average.



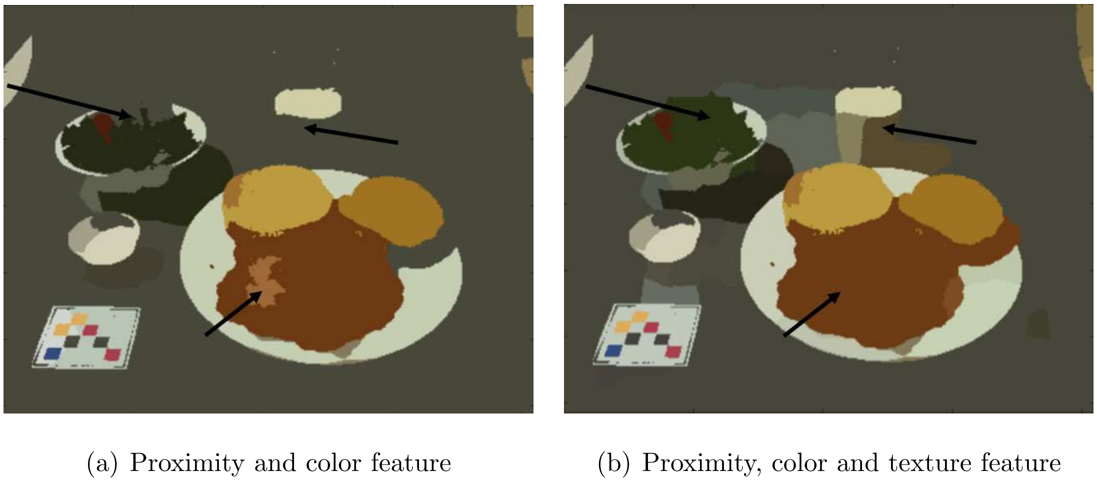
(a) Original Image (b) SLIC superpixels with graph connections

Fig. 4.1.: SLIC superpixel with labeled graph connections. The red connections indicate two similar superpixels, and the blue connections indicate two dissimilar superpixels.

4.5.1 TADA Food Dataset

In the TSDS, we mainly compare the proposed segmentation method with local variation [112] and hierarchical segmentation [110] for two reasons. First, based on our previous work [10,116], local variation outperforms other methods including Ncut, Mean Shift, active contours for the goal of food segmentation. Second, hierarchical segmentation is a popular method for general image segmentation, and has recently been adopted for food identification tasks [29]. For the proposed method discussed in Section 4.3, we use $\sigma = 0.9$, $r = 12$, $\beta = 10$. $\epsilon_1, \epsilon_2 \in [0.6, 0.8]$ and the number of superpixels ranging from 120 to 200 for the following experiment.

Figure 4.1 demonstrates the formation of the weighted graph using SLIC superpixels. In Figure 4.1, all the adjacent superpixels are connected based on their similarity in proximity, color and texture spaces. The red connections indicate that two superpixels are more similar while the blue connections show dissimilarity. Figure 4.2 shows the efficacy of the customized LBP feature. Using a graph model based only on color feature and proximity results in the image on the left. The image on the



(a) Proximity and color feature

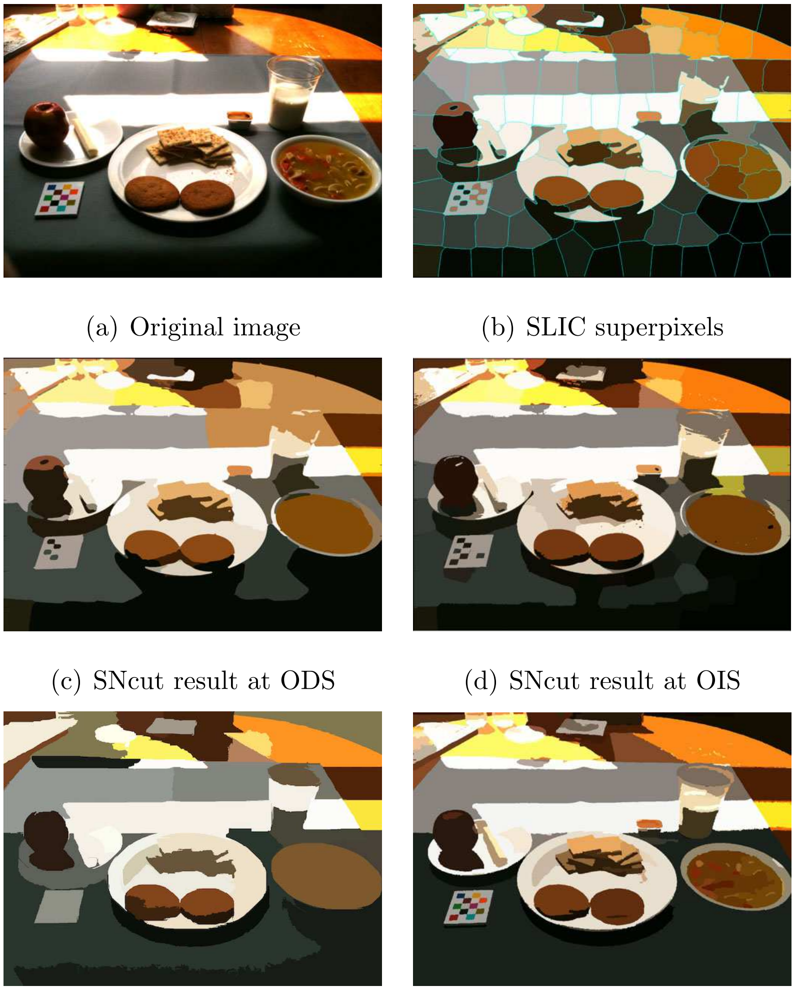
(b) Proximity, color and texture feature

Fig. 4.2.: Proposed segmentation method with different features. Arrows points to areas of segmentation difference.

right takes texture feature into account as well. The black arrows in Figure 4.2 point at some of the segmentation variations. As shown in Figure 4.2(b), texture feature helps to include food regions with higher color variance.

Figure 4.3 compares the results of SNcut to LV and hierarchical segmentation. We see that both SNcut and the hierarchical image segmentation are better at preserving the actual food contour than LV. However, hierarchical segmentation is often sensitive to edges insides food items, resulting over-segmented regions, for example, the noodle soup in Figure 4.3.

Figure 4.4 shows the PR curve and F-measure scores of SNcut, LV and hierarchical segmentation in different configurations. We evaluate all three methods using the object based evaluation metric discussed in Section 4.4. As illustrated, SNcut outperforms both LV and hierarchical segmentation. Because our evaluation method averages precision and recall over the number of segments associated with a certain food ground-truth, both precision and recall scores are low when the image is severely under-segmented or over-segmented.



(a) Original image

(b) SLIC superpixels

(c) SNcut result at ODS

(d) SNcut result at OIS

(e) LV result

(f) Hierarchical result

Fig. 4.3.: **Proposed segmentation method. From (a) to (f):** Original image, initial SLIC superpixels, SNcut output by thresholding at Optimal Dataset Scale [114], SNcut output by thresholding at Optimal Image Scale [114], output of LV and the hierachical segmentation (More examples are included in the supplemental material).

4.5.2 Berkeley Dataset

We report 4 different segmentation metrics for the BSDS based on the region based evaluation method proposed in [114]: the Optimal Dataset Scale (ODS) or best F-measure on the dataset for a fixed scale, the Optimal Image Scale (OIS) or aggregate

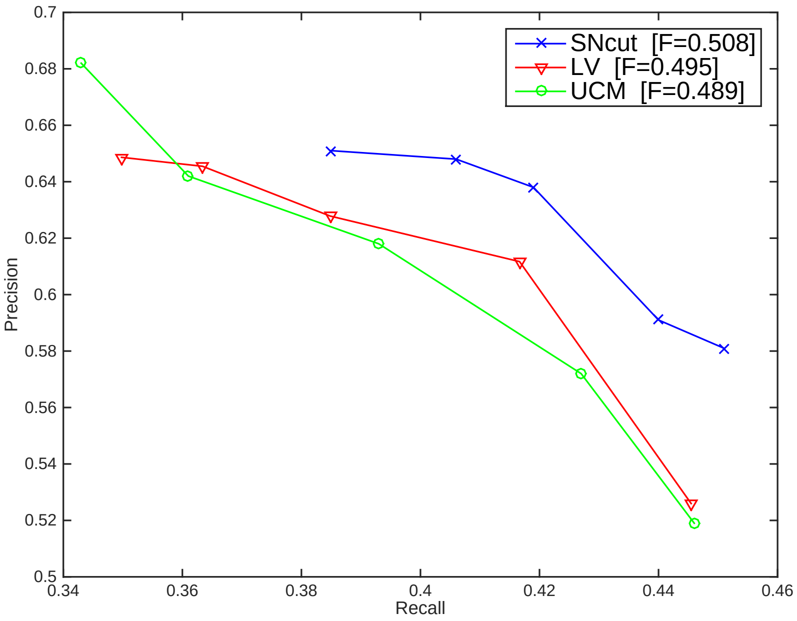


Fig. 4.4.: Precision and recall on the TSDS using the evaluation method discussed in Section 4.4.

F-measure on the dataset for the best scale in each image, Variation of Information (VoI) and Probabilistic Rand Index (PRI). VoI computes the amount of information in ground-truth not contained in the segmentation result. PRI measures the likelihood of a pair of pixels being grouped in two segmentations. Better segmentation usually has higher PRI and lower VoI.

Some examples are shown in Figure 4.5. The scores are summarized in Table 4.2. The results for methods other than SNcut are collected from [114]. We see that SNcut improves the score from Ncut by a relatively large margin. However, SNcut does not obviously separate itself from other methods when it compares to the hierarchical segmentation or gPb-owt-UCM. gPb-owt-UCM trains on natural images to learn significant object contours, so it is more resistant to over-segmentation inside the object of interest when we set a proper threshold for the boundaries. Moreover, SNcut depends on the initial superpixel map to extract the objects of interest, which results

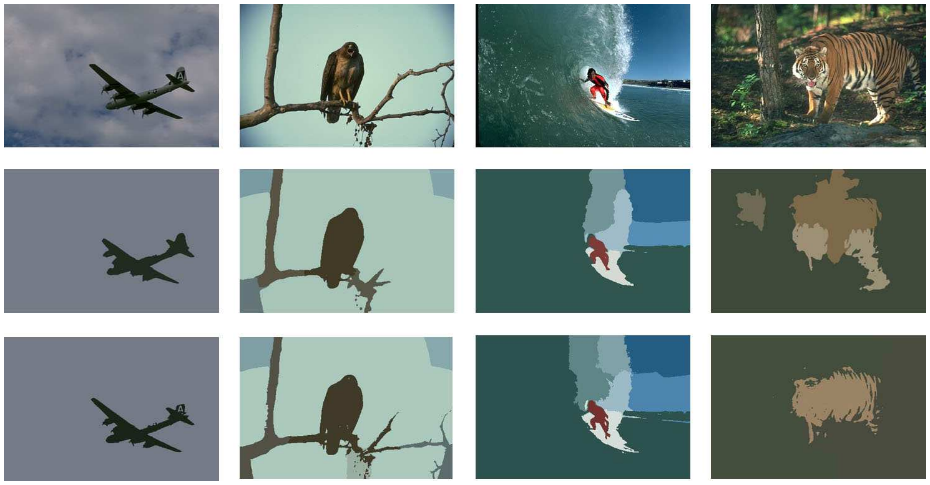


Fig. 4.5.: **Proposed segmentation method. From top to bottom:** Original image, segmentations output by thresholding at ODS and OIS. We have included a supplementary file which contains more examples of our segmentation method.

Table 4.2.: Region based segmentation evaluation on the BSDS.

Methods	PRI	VoI	ODS	OIS
Ncut	0.75	2.18	0.44	0.53
LV	0.77	2.15	0.51	0.58
Mean Shift	0.78	1.83	0.54	0.58
SNcut	0.78	2.11	0.55	0.59
gPb-owt-UCM	0.81	1.68	0.58	0.64

in loss of global information. Compared with the objects in the natural images, foods usually have more homogeneous textures and more uniform colors. Thus, the way superpixels enforce local smoothness works better for food images. More exmaples are shown in Figure 4.6.

On average, SNcut takes less than 3 seconds to segment one 481 × 321 image in BSDS. It is comparable to LV and 20 times faster than Ncut. For the full resolution 2048×1536 images in the TADA food dataset, SNcut takes 45 seconds to compute on

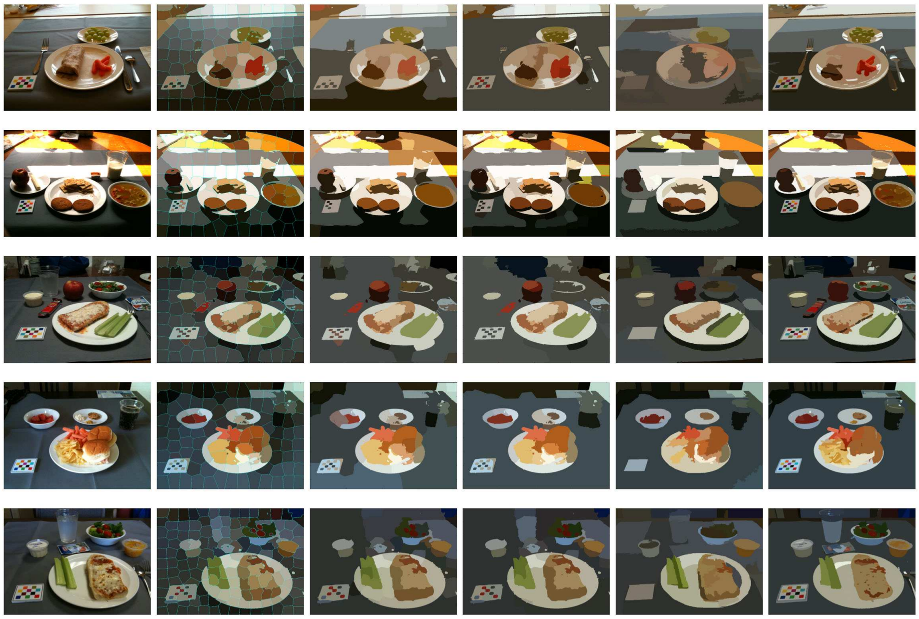


Fig. 4.6.: **From left to right:** original image, SLIC superpixel, SNcut results at ODS, SNcut results at OIS, local variation result, hierarchical segmentation.

average, which is 1.5 times faster than gPb-owt-ucm. All experiments were conducted on a desktop with quad-core 3.7GHz CPU and 16GB RAM.

In this chapter, we present a segmentation method that combines Ncut and superpixel techniques. We also introduce an object based segmentation evaluation method, especially for multi-food images. Experimental results suggest that the proposed method using multiple simple features is effective for food segmentation. According to our evaluation metric, SNcut outperforms some widely used segmentation methods and it also produces competitive results for natural images based on other segmentation benchmarks. In the future, we would like to investigate supervised learning on the weighted graph formation and explore a GPU implementation to further speed up the segmentation process.

5. WEAKLY SUPERVISED IMAGE SEGMENTATION

5.1 Overview of Weakly Supervised Image Segmentation Methods

In recent years the concept of deep learning [32] has been gaining widespread attention. As convolutional neural network (CNN) [137] gradually becomes dominant in many computer vision related areas, various recognition and classification tasks have been improved from the previous state-of-art methods [137–139]. Existing CNN models take advantage of labeled data which are used to learn which features are effective in a task as opposed to manually designed features. However, for more structured prediction, such as semantic segmentation, obtaining the pixel-level training data or even labeled bounding boxes is extremely time-consuming and expensive. For example, fully the convolutional network [138] requires careful annotation of the segmentation mask. Fast/Faster RCNN [139, 140] uses labeled data in the form of bounding boxes. Such dependency on fully supervised training poses a major limitation on scalability concerning the number of classes or tasks [141].

In the field of food image analysis, there is no publicly available segmentation ground-truth image dataset. The bounding box information provided in the UEC-FOOD256 dataset [39] is far from sufficient. Im2Calorie [35] uses several CNN models to analyze food intake, but the authors have not yet released their Food-201 dataset. Therefore, we would like to explore weakly supervised learning where only image-level labels indicating the presence or absence of objects are required.

Semantic image segmentation, i.e. assigning a semantic class label to each pixel of an image, is an important topic in computer vision. Many works require full supervision or pixel-level annotation to achieve this goal. In [142], the authors proposed a discriminative segmentation method. Multiple features including shape, texture, color and edge were incorporated in a conditional random field (CRF) model to deal with the ambiguities rooted from local prediction. The groundtruth labeling was required by the CRF model at training time. Compared to [142], C. Farabet et al. [143] used dense features extracted from a multiscale CNN to represent a pixel in the input image. Each pixel was then labeled based on the maximum likelihood given its feature vector. B. Hariharan et al. [144] described a simultaneous detection and segmentation (SDS) method which was essentially based on the Fast RCNN framework. The segmentation masks were obtained by refining the region proposals of interest.

Even though fully supervised segmentation approaches have demonstrated promising performance, collecting fully annotated training data poses a significant bottleneck to scale up the segmentation models. Thus weakly supervised segmentation methods were proposed to reduce the annotation effort. Previous work [145, 146] on weakly supervised learning showed that the output from a classification network can not only predict labels but also estimate object locations. In [147], a new loss function was proposed that uses location, classes and boundary priors to improve a segmentation system. N. Pourian et al. [148] used a spectral clustering approach that groups coarsely segmented image parts into communities. A community-driven graph is then constructed that captures spatial and feature relationships between communities while a label graph captures correlations between image labels. Finally, mapping the image level labels to appropriate communities is formulated as a convex optimization problem. In [146], the Class Activation Map (CAM) coupled with global average pooling (GAP) layers was introduced. This work showed that CNNs trained for the classification purpose also learned to localize visual objects without additional bounding box annotations.

In this chapter, we describe a graph based segmentation method for food images that uses a weakly supervised saliency model as prior knowledge. The contribution of this work is two-fold. First, we improve the CAM as a top-down saliency model by introducing a new pooling technique. Second, we incorporate the CAM trained on food datasets in the Biased Normalized Cut (Biased Ncut) segmentation method [149].

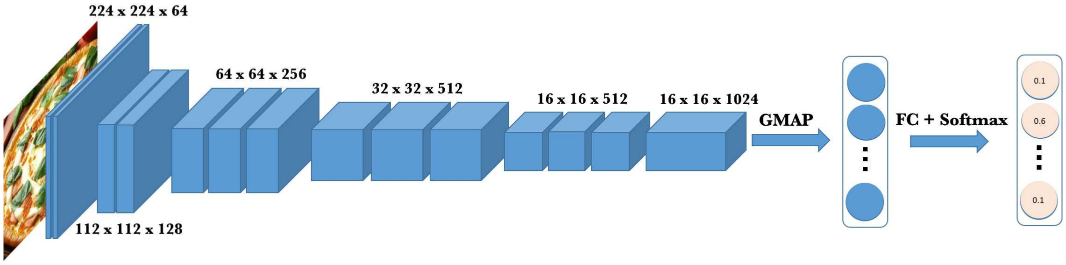


Fig. 5.1.: Network architecture for weakly supervised learning.

The proposed method shows promising results using various testing datasets and we believe it can also be used as an initial step before manual ground-truthing.

5.2 Network Architecture for Weakly Supervised Learning

Our model uses the fully supervised network of [150], known as VGG-16, that consists of 13 convolutional layers and 3 fully connected layers. To adapt the VGG-16 architecture to weakly supervised learning, we introduce several modifications. First, we add a 1024-channel convolutional layer and remove the first fully connected layer in the VGG-16 network. Second, we replace the max pooling layer before the fully connected layers with our proposed Global Max-Average Pooling (GMAP) layer. Figure 5.1 illustrates the proposed network architecture. We design the GMAP layer as a cascade combination of a Global Max Pooling (GMP) layer and a Global Average Pooling (GAP) layer. Furthermore, we extend the capability of GMAP by allowing adaptive pooling kernels. Similar to the ROI pooling layer [140], the size of pooling kernel varies based on the desired output, so that the output can be connected with a fully connected layer regardless of the size of the input images to the network.

Global Max-Average Pooling. As discussed in [146], the GAP layer outputs the spatial average of the feature map at the last convolutional layer. For example, if there are 1024 feature maps at the last convolutional layer, the GAP will generate a 1024 dimensional vector. We adopt the Class Activation Map [146], which is essentially a weighted sum of the feature maps of the last convolutional layer.

GMP and GAP have been successfully used in previous studies [147]. However, they both have their disadvantages. GMP tends to underestimate the regions of objects as the max pooling technique encourages the response from the single location of the highest activation. And GAP is more prone to overestimate object sizes, because it takes all the activations into account. To overcome these disadvantages in the context of semantic segmentation, we propose a new pooling technique, namely GMAP. The cascade structure of max and average pooling can be viewed as a generalized pooling layer of GAP and GMP,

$$F = \sum_{j=0}^{\lfloor (W-\alpha)/\beta \rfloor} \sum_{i=0}^{\lfloor (W-\alpha)/\beta \rfloor} \max(f_\alpha(\left\lceil \frac{\alpha}{2} \right\rceil + j\beta, \left\lceil \frac{\alpha}{2} \right\rceil + i\beta))/N \tag{5.1}$$

where $W \times W$ is the dimension of a feature map, f_α is the window function of size α, β represents the stride of the max pooling kernel and $N = \lfloor (W-\alpha)/\beta \rfloor^2$. From Equation 5.1, we can see that F becomes GAP if we let $\alpha = \beta = 1$ and it becomes GMP if we let $\alpha = W$. At this point, the proposed network as shown in Figure 5.1 takes 224×224 RGB images as input and generates a $1 \times 1 \times 1024$ vector after the GMAP layer and finally outputs a $1 \times 1 \times N$ vector of confidence scores. N is the total number of classes.

Adaptive Kernel and multi-label classification. The Region of Interest (ROI) pooling was first introduced in [140], which is essentially a simplified version of Spatial Pyramid Pooling (SPP) layer [151]. The goal of the ROI pooling layer or SPP layer is to adapt the various size of ROIs in the region proposal based networks. To complete the design of GMAP layer, we adopt the idea of the adaptive kernel. In other words, α in Equation 5.1 can be a function of W. Besides, the proposed network can also be extended to multi-label classification using multi-scale sliding window training as introduced in [145]. As shown in Section 5.4, we assume that one image only contains a single category of object.

5.3 Graph Based Segmentation

With the class activation map (CAM), the challenge is to use the prior knowledge for segmentation. It seems intuitive to incorporate salient stimuli [152] or fine-grained region proposals [153] into a graph model for the segmentation task. In [152] both bottom-up salient stimuli and object-level shape prior were integrated into min cut/max flow optimization. Such energy minimization is initialized with saliency map which is computed through context analysis based on multi-scale superpixels. Object-level shape prior is then extracted combining saliency with object boundary information. In [153], Cheng *et al.* implemented an iterative GrabCut [154] method which replaces user inputs with thresholded saliency maps.

Here, we incorporate the sampled CAM as a top-down constraint in Biased Normalized Cut (Biased Ncut) [149]. Compared to a saliency map [152] as shown in Figure 5.2, the weakly trained CAM is better at localizing the object of interest. Given a region of interest in the image, i.e. the CAM in our case, we would like to segment the image so that the segment is biased towards the specified region. The image is modeled as a weighted undirected graph $G = (V, E)$. The weight, w, on any edge, E, is a similarity measure between the end nodes of the edge. A region is modeled as a subset $T \in V$, of the vertices of the image. We are interested in the cut $(S, \bar{S})$, which not only minimizes the normalized cut value, $Ncut(S)$, but achieves sufficient correlation with the region specified by T, where

$$Ncut(S) \overset{def}{=} \frac{cut(S,\bar{S})}{vol(S)} + \frac{cut(S,\bar{S})}{vol(\bar{S})} \tag{5.2}$$

$$\bar{S} \overset{def}{=} V \backslash S \tag{5.3}$$

$$cut(S,\bar{S}) \overset{def}{=} \sum_{i \in S, j \in \bar{S}} w(i,j) \tag{5.4}$$

$$vol(S) \overset{def}{=} \sum_{i \in S, j \in V} w(i,j) \tag{5.5}$$

Belief Propagation. From Figure 5.2(b), we can see that the CAM peaks at where the network believes to show the most prominent feature of a specific class in the

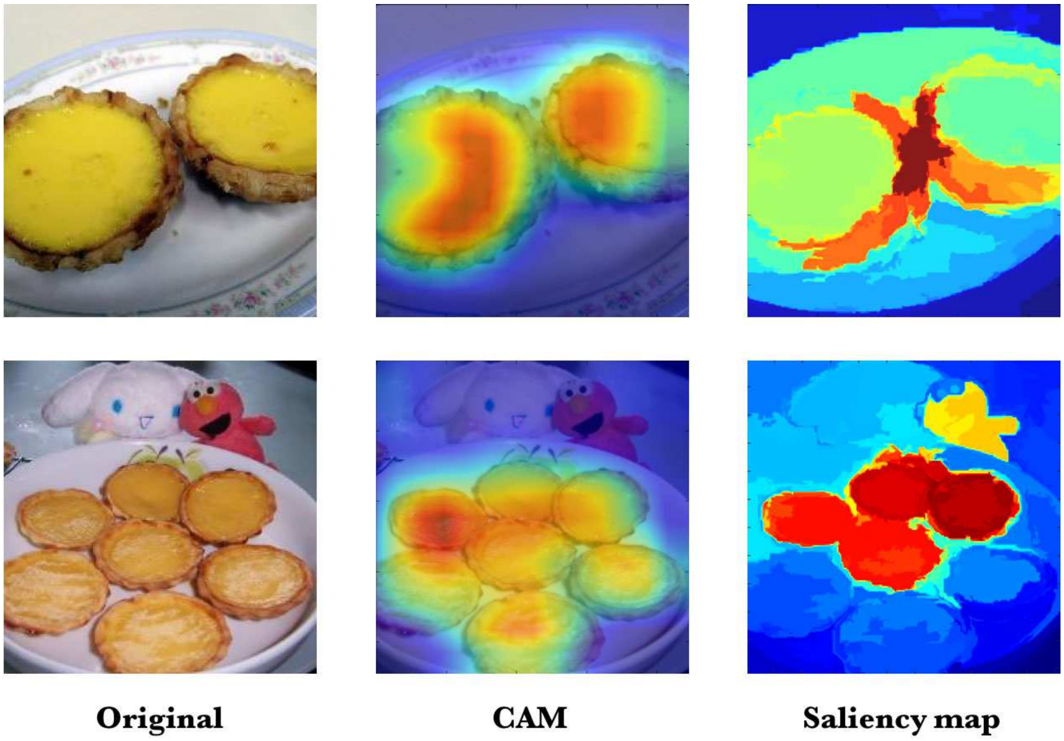


Fig. 5.2.: From left to right: the original image, its class activation map and saliency map [152].

image. However, it may not identify a part of the object as prominent even though the part of the object shares similar color and texture as its surroundings. To deal with this issue, we propose to use a multi-scale superpixel method to distribute the confidence that the network puts on certain regions in the image to their surroundings with similar color and texture.

Given an image, let $[S_1, ..., S_p, ...S_P]$ be the superpixel mask at different scales, where P indicates the number of scales we use and let B be the initial CAM. For any pixel $(\hat{x}, \hat{y})$ of a certain superpixel in S_p, we define its belief as follows,

$$B_p(\hat{x}, \hat{y}) = \frac{\sum_{(x,y)\in S_p}(B)}{||S_p||} \tag{5.6}$$

where $||S_p||$ represents the total number of pixels in the superpixel. So, if the superpixel is larger or the resolution of the superpixel mask is coarser, the belief is diffused more. We compensate the diffusal by introducing finer superpixel masks. Local variation [112] is used as the primary superpixel method, because it is fast and relatively good at preserving edges. Finally, the new CAM is obtained by normalizing the original CAM and the propagated belief across all the superpixel scales,

$$B'(x, y) = \frac{B + \sum_p(B_p(x, y))}{Z} \tag{5.7}$$

where Z is a normalization term that makes sure $B'(x, y) \in [0, 1]$.

Gassian Mixture Model and Sampling. We use a Gaussian Mixture Model in the new CAM to generate a trimap [155]. A trimap normally partitions an image into three regions: a definite foreground, a definite background and an unknown region. Then the foreground is uniformly sampled with a fixed step, P, and these sampled points are used as seeds, s_T, in the Biased Normalized Cut [149]. Given the graph $G = (V, E)$, the Laplacian of G, L_G and the normalized Laplacian, $\mathcal{L}_G$ are defined as follows,

$$L_G = D_G - A_G \tag{5.8}$$

$$\mathcal{L}_G = D_G^{-\frac{1}{2}} L_G D_G^{-\frac{1}{2}} \tag{5.9}$$

where D_G and A_G are the adjacency matrix and diagonal degree matrix of G. Finally, the optimal cut, $x*$, is obtained by combining the eigenvectors of $\mathcal{L}_G$ in the following way,

$$x* \propto \sum_{i=2}^{K} \frac{u_i^T D_G s_T u_i}{\lambda_i - \gamma} \tag{5.10}$$

where λ_i represents the i^{th} smallest eigenvalue, u_i is the corresponding eigenvector and γ is a correlation parameter [149].

5.4 Experimental Results

In this section, we describe our classification and segmentation experiments where we use several datasets to validate the proposed method, and we assume that one image only contains a single category of object.

Classification. To validate the proposed pooling method, we trained various models using Caltech-256 [156], UECFOOD-256 [39] and Food-101 [27]. Caltech-256 [156] contains 30607 images of 256 object categories. UECFOOD-256 [39] consists of more than 31,000 images from 256 food categories, most of which are popular foods in Japan and other Asian countries. Food-101 [27] contains 101 food categories, each of which has 1000 images. Each dataset is randomly split in the 70/10/20 fashion for train/validation/test sets. We used a pretrained VGG-16 network to initialize the first 13 layers in our model and all the experiments were done in the Tensorflow [157].

Table 5.1 compares the Top 1 classification accuracy of different pooling methods in the proposed network. Training the model with GAP was performed with stochastic gradient descent with learning rate of 0.01 and momentum of 0.9 while learning rate of 0.002 and momentum of 0.9 were used for the other pooling methods. $GMAP-\alpha-\beta$ represents GMAP with a $\alpha \times \alpha$ max pooling kernel and stride of β. As shown in the table, the network with $GMAP - 4 - 2$ shows slightly better results across the three datasets. Figure 5.3 illustrates the visual differences of the CAMs when different pooling methods are used. Recently Yanai *et al.* reported 67.57% on the

Table 5.1.: Comparison of different pooling methods.

Accuracy (%)	Caltech-256	UECFOOD-256	Food-101
GMP	81.05	63.97	72.75
GAP	81.09	64.01	72.78
GMAP-2-2	81.20	64.80	73.81
GMAP-3-3	81.05	64.01	73.55
GMAP-4-2	81.53	64.89	74.02

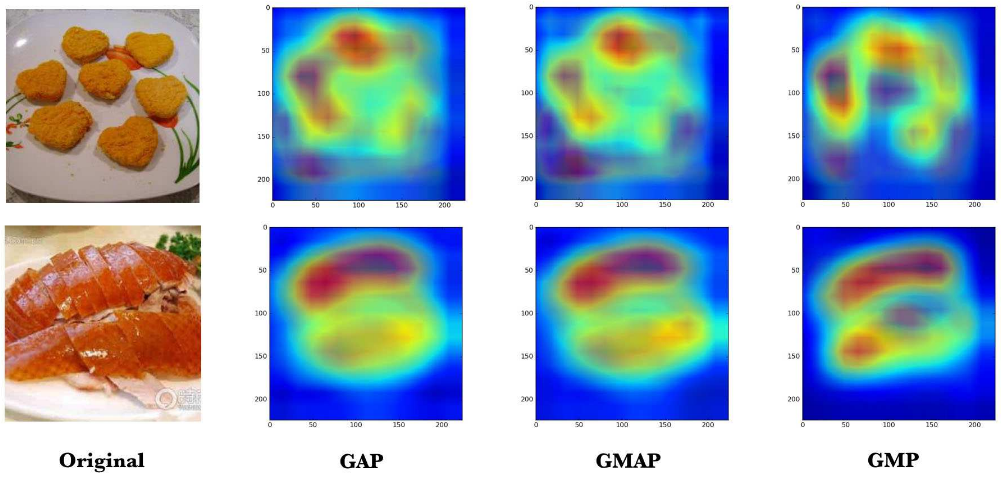


Fig. 5.3.: Class activation maps using different pooling methods.

UECFOOD-256 using a modified AlexNet [34] and the best result, 78.11%, on the Food-101 is achieved using GoogleNet by Ao *et al.* [33]. Compared to their work, our model demonstrates comparable accuracy despite using a much simpler network architecture.

Furthermore, we picked the images of 31 food categories from Food-101 [27] that are common in UECFOOD-256 and we named it the Food-31 dataset. We wanted to test the proposed model with $GMAP - 4 - 2$ trained on UECFOOD-256 [39] using the Food-31 dataset, since the images from these two datasets were initially collected from different sources and thus they should occupy slightly different domains in the feature space. As shown in Figure 5.4, the images of the same category look quite different in the different datasets. We achieved 85.8% accuracy over the 31,000 images in the Food-31 dataset without any fine-tuning.

Segmentation. To evaluate the segmentation accuracy on the food images, we use a free-living study [158] from the TADA system. It consists of 1453 images of 56 commonly eaten food taken by 45 participants within a week and we have manually ground-truthed over 900 food segments with labels. To our knowledge, there is no publicly available segmentation ground-truth for dataset food images and we would

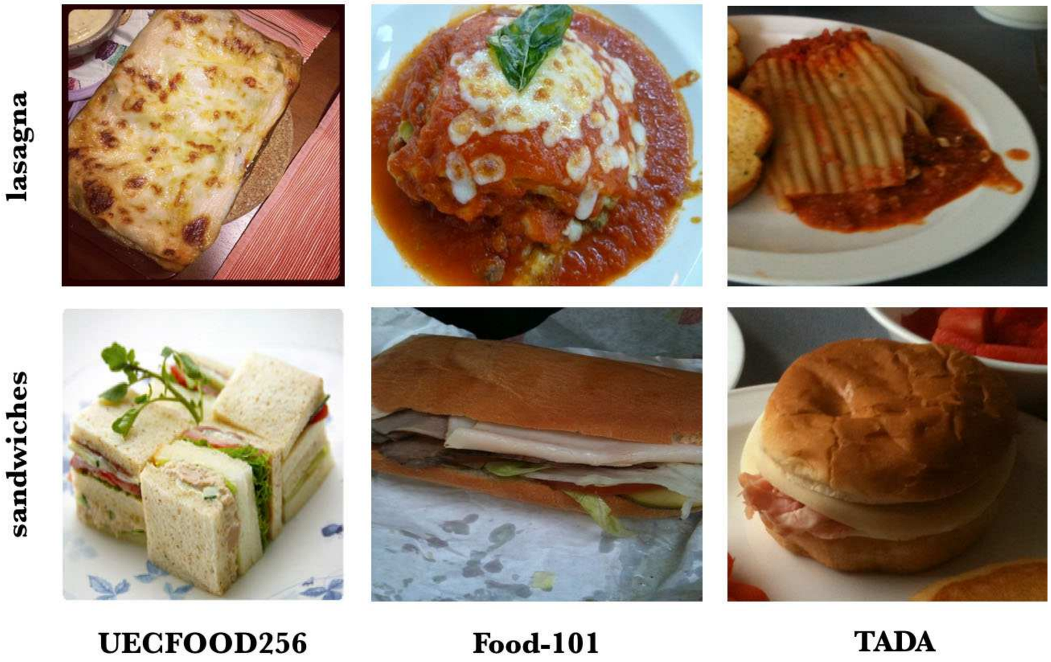


Fig. 5.4.: Examples from different datasets.

like to release our data for the academic use soon. Nine out of the 56 food categories in the free-living study have the same counterparts in the Food-101 [27] (see Figure 5.5) and there are 317 ground-truth in total.

Based on our experiment, we choose $P = 40$, $K = 16$ and $\gamma = 1e{-}4$ as discussed in Section 5.3. Figure 5.6 shows an example image from the free-living dataset. Seeds in Figure 5.6(c) are sampled from a trimap generated from Figure 5.6(b). Figure 5.6(d) represents the combination of the reshaped eigenvectors as discussed in Section 5.3.

The final segmentation masks are obtained by binarizing the biased normalized cut. We use a region based metric [128] to evaluate the segmentation masks. Figure 5.7 shows the precision and recall [159] when various thresholds are used. Compared to our previous work, i.e. SNcut [160], the biased normalized cut based on the belief-propagated CAM demonstrates superior performance. More examples are shown in Figure 5.8.

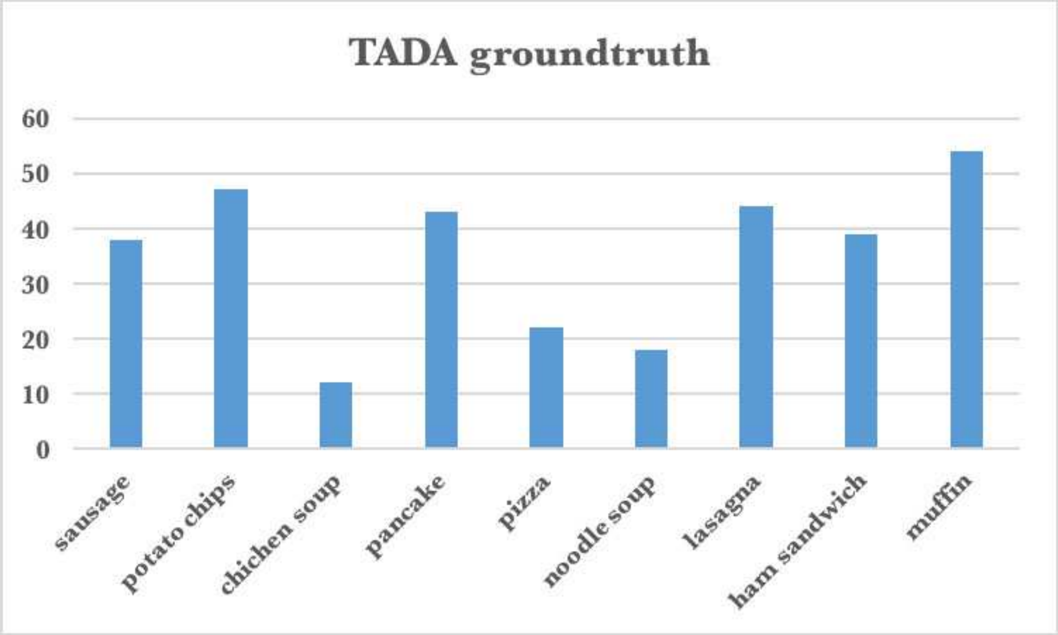
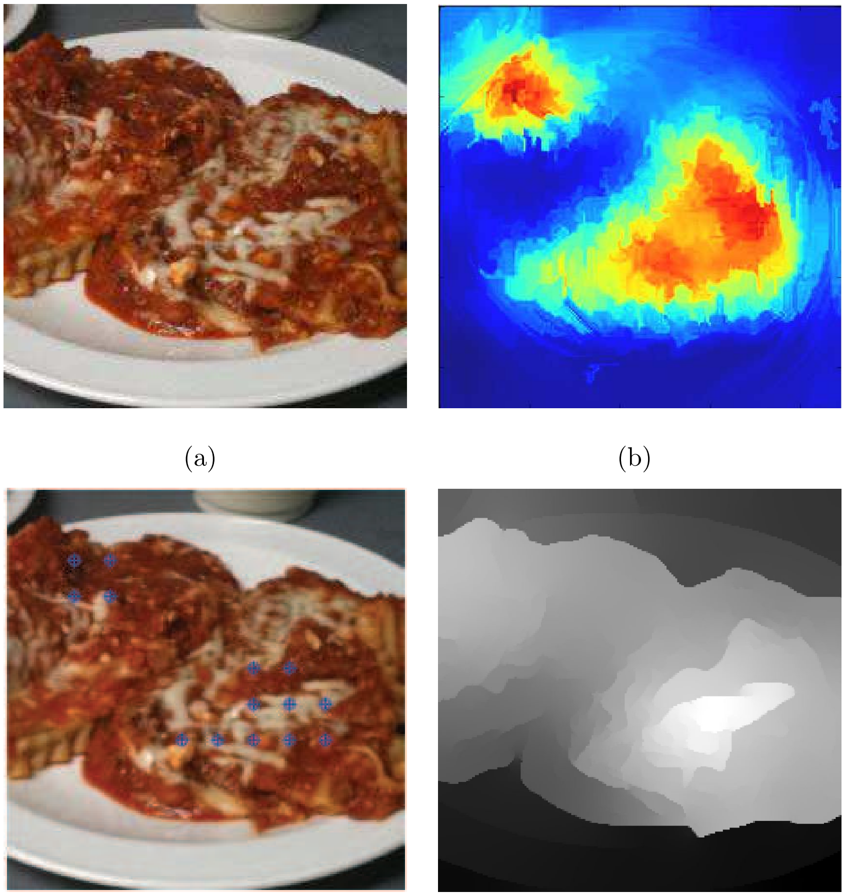


Fig. 5.5.: TADA groundtruth statistics of 9 selected food categories which are common in the Food-101 dataset.

In this chapter, we described a weakly supervised CNN model with a new pooling technique and incorporate a class activation map for graph based segmentation. Our experiments shows promising results for both classification and segmentation tasks. In the future, we would like to test our model using a larger dataset and investigate multi-food segmentation.



(a)

(b)

(c)

(d)

Fig. 5.6.: (a) Original image. (b) The belief-propagated class activation map. (c) Seeds, s_T as discussed in Section 5.3. (d) The biased normalized cut.

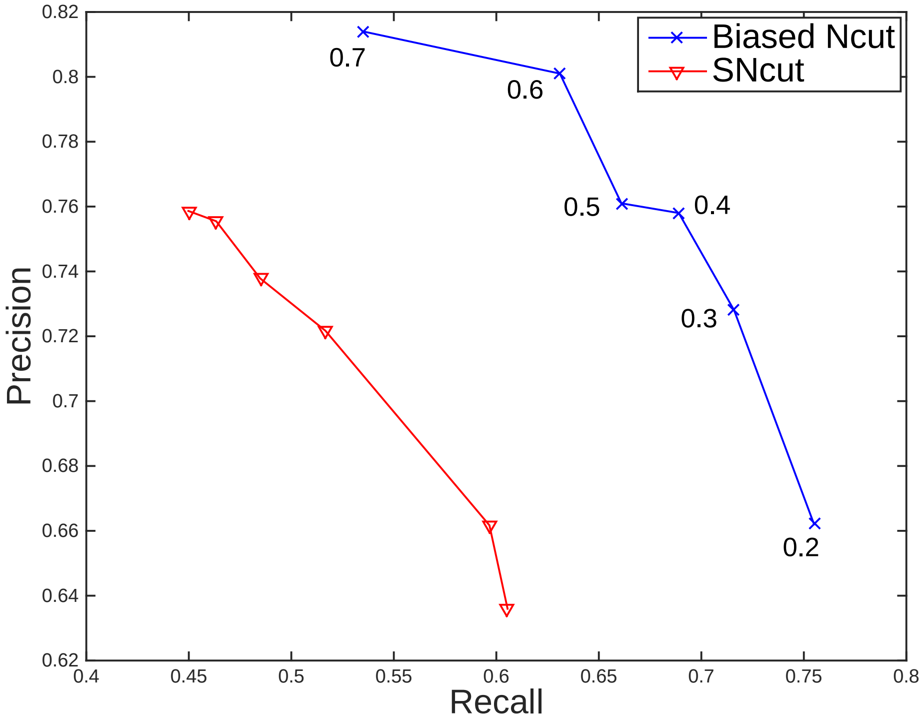


Fig. 5.7.: Precision and recall of the segmentation results. **Blue**: Biased Ncut with the CAM prior. **Red**: SNcut

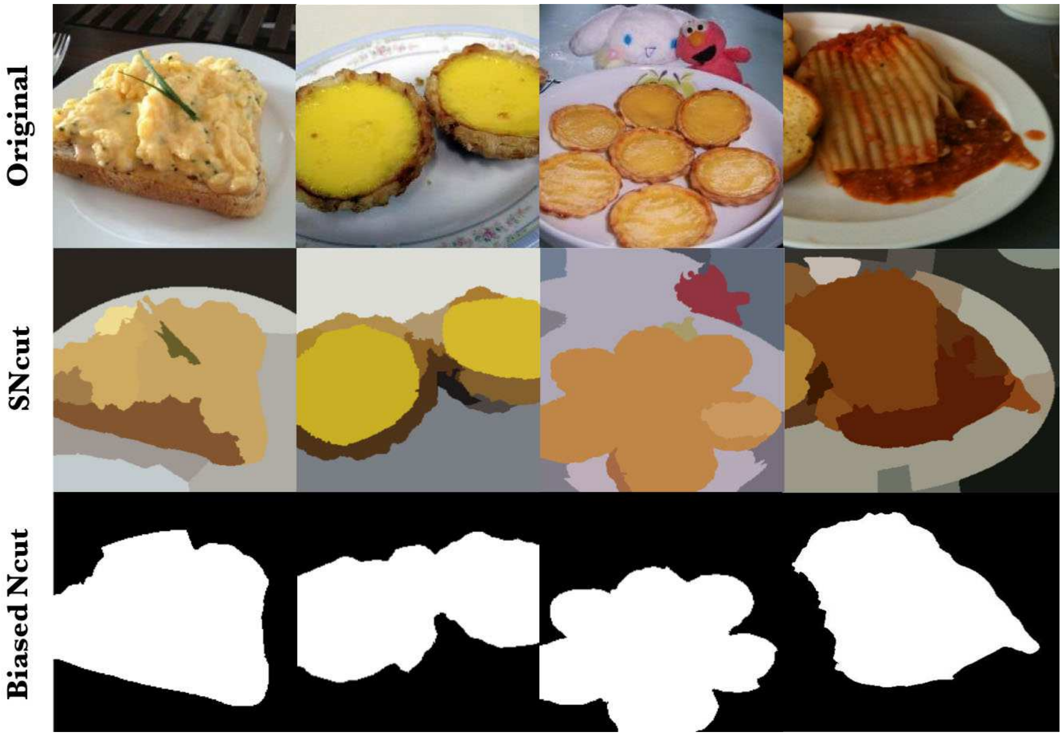


Fig. 5.8.: Example segmentation masks.

6. LARGE SCALE FOOD IMAGE ANALYSIS USING DEEP NETWORKS

In the past two decades, we have witnessed the power of deep learning trickling down to many aspects of the modern society: from social media to cyber security, from music/movie recommendation service to AI assistants. However in the 90s, most researchers counted deep learning technique out due to the lack of efficient optimization methods and computational power. Hand-engineered features were the unstoppable force in the field of computer vision. This chapter is organized as follows: first, we review popular deep networks for both classification and detection; we then propose a three-stage food recognition pipeline which consists of a food/non-food localizer, a food classifier and a food segmenter; next, we investigate adversarial examples in the domain of food images; finally, experimental results are presented and discussed.

6.1 Overview of Classification Networks

AlexNet In 2012, AlexNet [137] rekindled the interest of deep neural networks by winning the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [161]. Then it became one of the most reputed network structure. AlexNet evolved from LeNet [162] to a larger neural network with 5 convolutional layers and 3 fully connected layers. It was designed to learn complex features and object structures. A. Krizhevsky et. al. [137] proposed to use Rectified Linear Units (ReLU) as the nonlinear activition function,

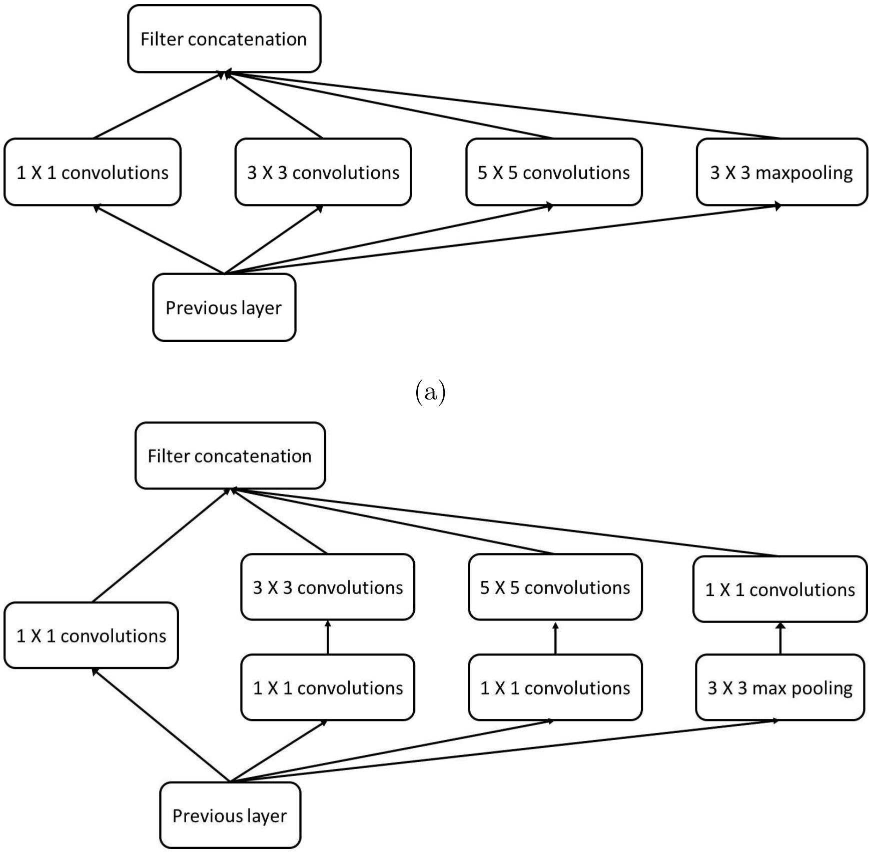
$$f(x) = \max(0, x) \tag{6.1}$$

It has been demonstrated that using activation functions with non-saurating nonlinearities results in much faster training time. AlexNet also ultilizes the dropout technique to reduce overfitting. Tested on the subset of ImageNet with more than 1000 images in each of 1000 categories [161], it achieved top 5 error rate of 15.3% with 6 days of training. Figure 6.1 illustrates a collection of randomly selected images from ImageNet.

Fig. 6.1.: Random images from the ImageNet [161].

VGGNet The network structures [150] developed by Visual Geometry Group (VGG) at Oxford University were the first to use the regulated 3 × 3 convolutional kernels in each convolutional layers. Compared to the first convolutional layer of AlexNet

(11 × 11 with stride 4), K. Simonyan and A. Zisserman [150] stacked up to 19 convolutional layers of very small receptive fields and got the second place in the classification task of ILSVRC 2014. The authors argued that a stack of smaller kernels has similar effect as a larger kernel.



(a)

(b)

Fig. 6.2.: (a) Early implementations of the Inception module. (b) Inception module with 1 × 1 bottleneck.

Inception Module and GoogLeNet As we stated above VGGNet [150] ranked second in the classification task of ILSVRC 2014, GoogLeNet [36] was the winner with a top 5 error rate of 6.7%. GoogLeNet [36] is a 22-layer network structure that breaks the general approach of simply stacking convolutional and pooling layers.

Table 6.1.: The structure of the VGG-16 and VGG-19 networks

Network Structure	
VGG-16	VGG-19
input (224 × 224 RGB)	
conv3-64 conv3-64	conv3-64 conv3-64
maxpool	
conv3-128 conv3-128	conv3-128 conv3-128
maxpool	
conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool	
conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool	
conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool	
FC-4096	
FC-4096	
FC-1000 + softmax	

The authors introduced an optimal local sparse structure called Inception Module (Figure 6.2) in GoogLeNet. The goal was to reduce the number of weights in the readily available dense structure while achieving comparable performance.

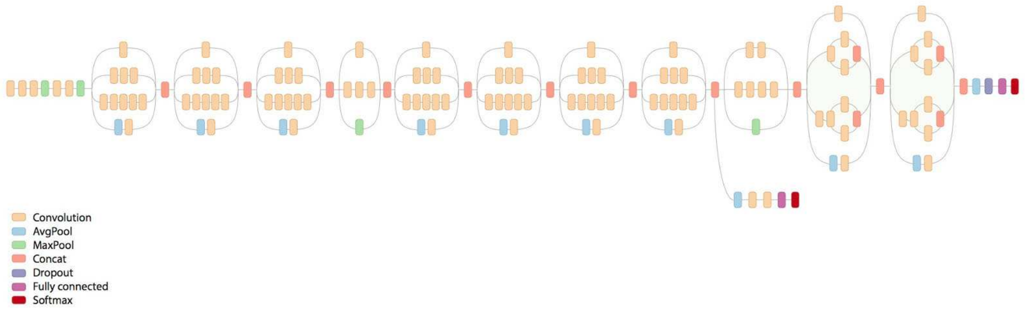


Fig. 6.3.: The structure of Inception-3 [163].

The Inception module, inspired by the idea of Network in Network [164], effectively increases the width of the network by considering features of a layer at different scales. It also considers cross-channel correlations by using 1×1 convolution while reducing feature dimensions as shown in Figure 6.2(b).Thus, despite being deeper, GoogLeNet is 9 times smaller than AlexNet and 3 times smaller than VGGNet in terms of the number of parameters.

As a follow-up to the first generation of GoogLeNet [36], C. Szegedy et al. [165] discussed the discretionary design principles of more advanced Inception modules, for example, avoiding representational bottlenecks and balancing the width and depth of the network.

As a result, the Inception-v3 network as illustrated in Figure 6.3 was proposed. It consist of 42 layers and uses spatial factorization and auxiliary classifiers to achieve a promsing margin over the previously reported result [165].

Residual Networks From LeNet [162] to GoogLeNet [36], we have witnessed the growth of network models regarding their size and capacity. Although it seems that the most straightforward way of improving the performance of deep neural networks is to increase the width or the depth, the best model only contains less than 25 layers [36]. Researchers strive to resolve issues inherited from larger networks, such as overfitting, vanishing gradient and computational limitation.

In late 2015, K. He et al. [166] proposed a new CNN architecture called Residual Network (ResNet) with jawdroppingly 152 layers. Aside from the new record of the number of layers, ResNet achieved an incredible error rate of 3.6% in ILSVRC 2015.

To overcome the obstacle of vanishing/exploding gradients, the authors introduced a residual learning framework. The idea of residual learning is based on a hypothesis that if multiple nonlinear layers can asymptotically approximate a function $H(x)$, they can also asymptotically approximate the residual function, $H(x) - x$.

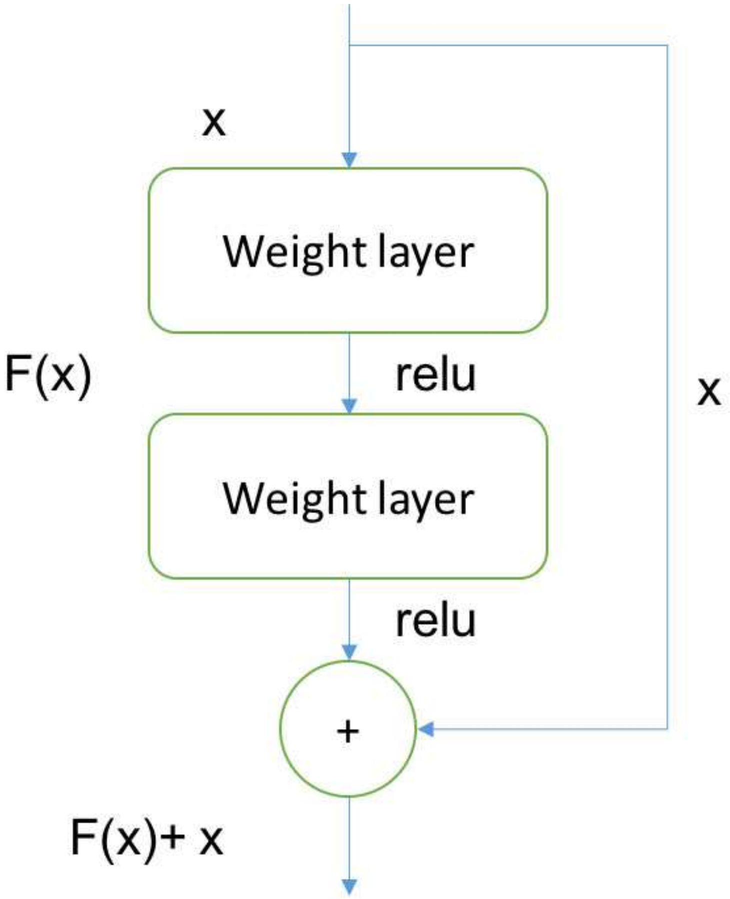


Fig. 6.4.: Residual learning block.

As shown in Figure 6.4, $F(x)+x=H(x)$ and x is the shortcut connection, which allows the gradient to pass backwards directly. By stacking these layers, the gradient could theoretically skip over all the intermediate layers and reach the bottom without being diminished.

DenseNet DenseNet [167] is one of the worth-noting works on making CNNs deeper. Compared to the shortcut in ResNet, DenseNet takes the insights of the skip connection to the extreme. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs to all subsequent layers. Hence, the l^{th} layer has l inputs, consisting of the feature-maps of all preceding convolutional blocks. Its own feature-maps are passed on to all $L-l$ subsequent layers. This generates $\frac{L(L+1)}{2}$ connections in an L-layer network. Figure 6.5 illustrates a dense block with 5 convolutional layers.

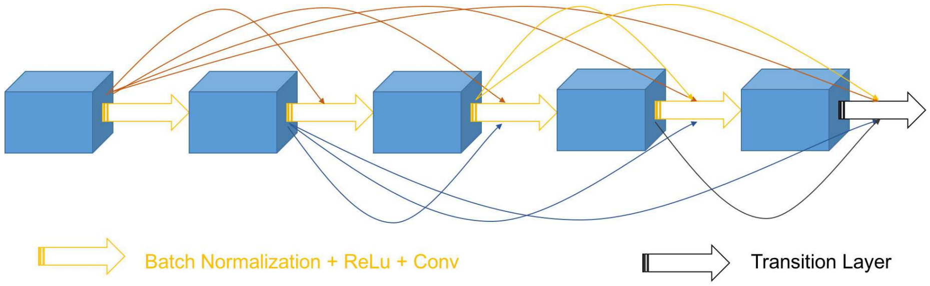


Fig. 6.5.: A 5-layer dense block with a growth rate of 4.

6.2 Overview of Object Localization Networks

From RCNN To Faster RCNN Compared to image classification, object detection usually requires more complex and structured methods to solve. RCNN [168] was arguably the first work that used deep features extracted from a CNN and showed superior detection performance on the PASCAL VOC dataset [169] as compared to methods using HOG-like or SIFT-like features. In RCNN [168], selective search [170] is used to generate roughly 1000-2000 region proposals per given image. Then the proposals are resized to a fixed resolution 227 × 227. Depending on the application, the resizing can be done by down-sampling, cropping, padding or interpolation. Next, the proposals are fed into a CNN. At the bottom of the CNN, usually after the last fully connected layer, the feature of a proposal is extracted. Based on the intersection of the proposal and the groundtruth, some proposals and corresponding features are filtered out. Finally, the remaining features and corresponding class labels are used to train an SVM. It is almost obvious that the computation in RCNN is not optimized as every proposal has to go through the whole CNN. Besides, the global context in the input image is ignored, because the SVM classifier is only trained on the proposed image patches. Furthermore, it is relatively hard to train three separate parts in the network. As a successor of R-CNN [168], Fast R-CNN [140] was proposed to deal with the cons of R-CNN. At the training time, a Fast R-CNN network takes an entire image and a set of object proposals as input. The network first processes the whole image with several convolutional and max pooling layers to produce feature maps. Then, for each object proposal, a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map. Each feature vector is used to optimize a multi-task loss function. The multi-task loss function combining the losses from two sibling output layers, i.e. the softmax layer to predict labels and the bounding box regression layer to refine the proposals, is defined as follows,

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \tag{6.2}$$

where p is the output vector from the softmax layer, u is the groundtruth label, t^u is the vector indicating the bounding box regression offsets and v is the groundtruth bounding box. In [140], the log loss was used for L_{cls} and the smooth L1 loss for L_{loc}. Both R-CNN [168] and Fast R-CNN [140] rely on the generic region proposal method, such as selective search [170]. In general, generic region proposal methods are only implemented on CPU, which becomes a bottleneck if we want to achieve faster object detection in real time. S. Ren et al. [139] proposed a Region Proposal Network (RPN) to replace the generic region proposal methods and they combined RPN with Fast R-CNN structure to form the Faster RCNN network. RPN is a fully convolutional network for effectively proposing RoIs. At training time, Faster RCNN uses a similar multi-task loss function as Fast RCNN and RPN generates translation-invariant anchors at multiple scales to assist bounding box regression. Since 2015, Faster RCNN has become especially impactful and has led to numerous follow-up works [171–174].

SSD Single Shot MultiBox Detector [171] (SSD) differs from Faster RCNN in that it is a single feed-forward network without a secondary RPN. It maps the bounding boxes in the output space to a set of boxes at other feature levels and generates default boxes over different aspect ratios. The authors proposed a matching strategy to correspond default boxes to a groudtruth at different levels. The multi-task loss function is slightly different from the ones used in Fast/Faster RCNN as the class loss function is replaced by a softmax over multi-class confidence vector. As SSD does not resample features for each bounding box, it results in faster detection speed. SSD also achieved competitive accuracy on the PASCAL, COCO and ILSVRC datasets. Figure 6.6 shows an example of the SSD network.

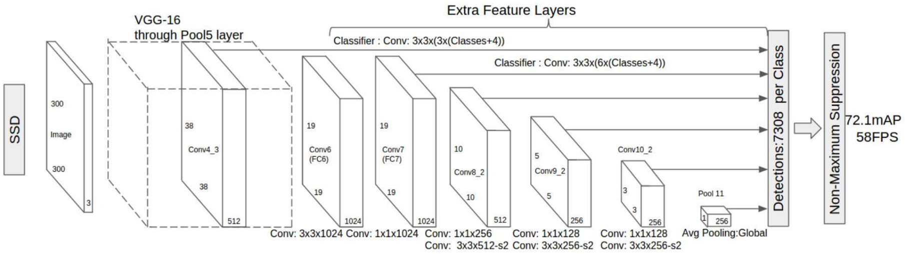


Fig. 6.6.: Single shot detector [171].

YOLO Similar to SSD, You Only Look Once (YOLO) [173] is another network with a single branch. As shown in Figure 6.7, the input image is divided into regions of the same size. Then a set of bounding boxes and objectness scores are predicted for each region. To produce the final result, these bounding boxes are weighted by the predicted probabilities. The authors envisioned YOLO to be a real-time detector with reasonable accuracy and good scalability, as a simplified version can achieve up to 155 FPS with GPUs. More recently, J. Redmon et al. proposed an updated version or YOLO-v2 [175] that can detect over 9000 different object categories. The authors focused on resolving the shortcomings of the original YOLO, such as frequent localization errors and low recall value. Some of the improvements include adopting the batch normalization technique and using a higher resolution classifier during training.

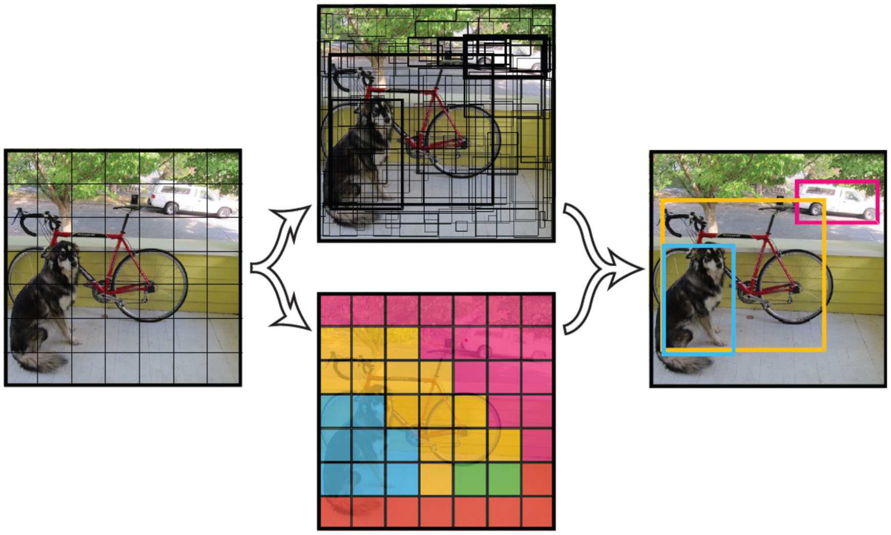


Fig. 6.7.: YOLO divides an input image into a grid of cells and predicts bounding boxes with corresponding confidence scores [173].

6.3 Techniques To Improve Training Efficiency

Batch Normalization Batch Normalization (BN) [176] is one of the most important works in the last five years of deep learning research. It has helped the current network structures to achieve faster learning and higher overall accuracy.

Stochastic gradient descent (SGD) has proved to be effective to train deep networks and its goal is to minimize a loss function while adjusting a hyperparameter θ,

$$\theta = argmin_{\theta} \frac{1}{m} \sum_{i=1}^{m} l(x_i, t_i, \theta) \tag{6.3}$$

where x_i and t_i are the input data and the corresponding target value. m is the size of the mini batch.

However, the change of the distributions from layers to layers poses a problem as the weights in layers are consistently adapted to the new distribution. Such change in the distributions is often referred to as *Internal Covariate Shift*. During back propagation, the internal covariate shift can cause the learned weights to compensate the outliers instead of producing required outputs. This results in a longer time of convergence. The basic idea of BN is to regularize input features of a certain layer to have the mean of zero and the variance of one. However, simply whitening the data might change the representation of the layer. To address this, the authors introduced a parameter transformation algorithm or *Batch Normalizing Transform*,

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \tag{6.4}$$

$$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{6.5}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{sqrt(\sigma_B^2 + \epsilon)} \tag{6.6}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN \tag{6.7}$$

Modern deep networks, such as Inception-v3, ResNet, have all incorporated the BN layers.

Deformable Convolutional Layer A new form of convolution and pooling, i.e. deformable convolution and deformable RoI pooling were recently introduced in [177]. The idea of the deformable convolutional networks (DCN) is rooted from one of the weaknesses of general deep networks. Without specific supervision, the deep networks usually do not generate well with the rotated and scaled data. DCN offers an ability to adapt to various geometric transformation.

Figure 6.8 shows the structure of the deformable convolution layer (DCL). For a traditional convolution kernel K, the output at each position is computed as follows,

$$y(\hat{p}) = \sum_{p_i \in O} K(p_i) x(\hat{p} + p_i) \tag{6.8}$$

where $\hat{p}$ represents the center position of the convolution kernel, p_i is any position surrounding the center with an offset of 1 and O is the offset space.

In DCL, the authors added an learnable offset Δp_i, such that,

$$y(\hat{p}) = \sum_{p_i \in O} K(p_i) x(\hat{p} + p_i + \Delta p_i) \tag{6.9}$$

By adding Δp_i, the convolution kernel is able to learn image context from irregular positions.

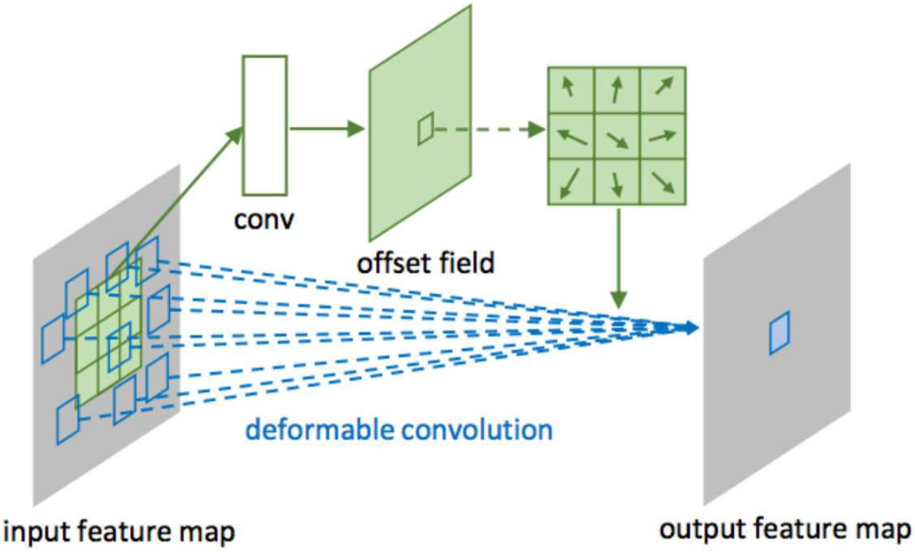


Fig. 6.8.: 3 × 3 deformable convolution [177].

6.4 Proposed Method

Related Work The deep learning based food image analysis has gained a huge popularity over the past few years. In 2014, Bossard et al. [27] introduced the arguably first large-scale food image dataset, or “Food-101” and reported the top 1 classification accuracy of 56.40% using AlexNet. Another widely used food image dataset, known as “UEC-FOOD100/256”, was created by K. Yanai et al. [39]. In 2015, the same group of researchers used a slightly modified AlexNet to mine deep features and used a one-vs-rest linear SVM to achieve the top 1 classification accuracy of 78.77% and 67.57% respectively on the UEC-FOOD100/256 dataset [34]. The authors compared the features extracted from the original and finetuned network to the handcrafted features and claimed the finetuned features outperformed the others. Later in 2015, S. Ao et al. [33] applied similar idea of using finetuned features to the Food-101 dataset and compared the deep features extracted from AlexNet and GoogLeNet. They reported the top 1 and 5 classification acurracy of 78.11% and 93.51% using GoogLeNet.

M. Bolanos et al. [178] proposed a two-step food image analysis method which is composed of a food localization step using food activation maps and a classification step using GoogLeNet. They achieved 95.64% food-vs-nonfood localization accuracy with a 0.5 Intersection over Union (IoU) thresholding and 79.20% top 1 accuracy on the Food-101 dataset.

Even though a number of deep networks were studied in the domain of food images, an end-to-end food analysis pipeline is still lacking. Based on our previous study, image segmentation plays a significant role in dietary analysis [10,116]. In this section, we describe a three-stage food analysis pipeline as shown in Figure 6.9. The proposed method involves three stages: 1) a food-specialized localizer, 2)a fine-grained food classifier pretrained on the ImageNet categories and 3) a food segmentation method, for example, the weakly supervised technique discussed in chapter .

6.4.1 Food Localization

Different from our previous work [10,116], where the automatic analysis starts with a generic segmentation method, followed by extracting features from each segment,

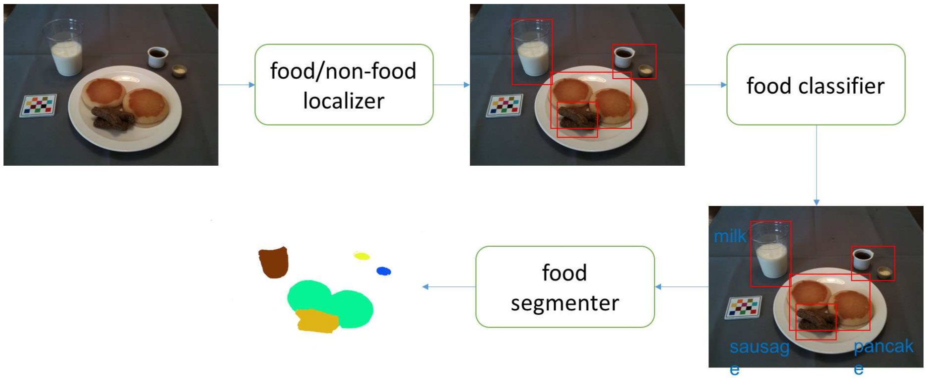


Fig. 6.9.: Three-stage food image analysis diagram.

and finally food labels are predicted based on a majority vote, the proposed method starts with a dedicated food localizer.

In [178], the authors adopted the class activation map [146] and retrained it using the food and non-food images. Here, we propose to use the Faster RCNN as a baseline localizer and we change the mid-level convolutional layers to the deformable convolutional layers [177]. As mentioned in Section 6.2, the Faster RCNN has two branches: the Feature Extraction Network (FEN) and the Region Proposal Network (RPN). The softmax layer of the FEN is modified so that it can perform binary classification. We evaluate the VGGNet, ResNet-50 and ResNet-101 as our candidate FENs. All the FENs are pretrained on the Pattern Analysis Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes Challenge (VOC) 2012 dataset and finetuned using the UEC-FOOD256 dataset.

6.4.2 Food Classification

After obtaining the fine-grained proposals generated by the food localizer, the next step is to classify each proposal to one of the 101 food categories. We evaluate GoogLeNet, ResNet-101 and Inception-v3 as our primary classifiers. All the networks are pretrained on ImageNet and then finetuned using the Food-101 dataset, as transfer learning has been proven to be particularly effective for the general object recognition tasks. We modify the softmax layer to have 102 outputs, i.e. 101 food classes plus an additional "background" category.

6.4.3 Adversarial Examples

Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake. They have become notorious as they are imperceptibly different from the original images but are readily confusing to the classifier. Here, we want to investigate the adversarial examples in the food image domain using the following techniques.

Fast Gradient Sign Method In 2015, I. Goodfellow et al. [179] proposed the Fast Gradient Sign Method (FGSM) for generating adversarial examples using the derivative of the loss function of the CNN with respect to the input feature vector. Given an input image, FGSM perturbs each feature in the direction of the gradient by ϵ, where ϵ is a parameter that determines the magnitude of the perturbation. For a network with loss $J(\Theta, x, y)$, where Θ represents the CNN parameters, x is the CNN input, and y is the label of x, the adversarial example is created as

$$x^* = x + \epsilon sign(\nabla_x J(\Theta, x, y))$$

Jacobian-based Saliency Map Attack Jacobian-based Saliency Map Attack (JSMA) [180] is an iterative method that creates adversarial examples by targeting on another specific class of object. For an input x and a neural network N, the output for class j is denoted $N_j(x)$. To achieve an output of target class t, $N_t(X)$ must be increased while the probabilities $N_j(X)$ of all other classes $j \neq t$ decrease, until $t = argmax_j \; N_j(X)$. This is accomplished by exploiting the adversarial saliency map, which is defined as

$$S(X,t)[i] = \begin{cases} 0, \text{if } \frac{\partial F_t(X)}{\partial X_i} < 0 \text{ or } \sum_{j \neq t} \frac{\partial F_j(X)}{\partial X_i} > 0 \\ (\frac{\partial F_t(X)}{\partial X_i}) |\sum_{j \neq t} \frac{\partial F_j(X)}{\partial X_i}|, \text{otherwise} \end{cases}$$

for an input feature i. Starting with a normal sample x, we locate the pair of features $\{i, j\}$ that maximize $S(X, t)[i] + S(X, t)[j]$, and perturb each feature by a constant offset ϵ. This process is repeated iteratively until the target misclassification is accomplished.

6.5 Experimental Results

6.5.1 Food Datasets For Localization and Classification

Food-101 As we mentioned before, Food-101 [27] is arguably the first dedicated food image dataset of the large volume. It contains 101,000 images of 101 foods that are commonly consumed around the world.

UPMC Food-101 UPMC Food-101 [38] can be considered a "twin dataset" of Food-101, as they share the same 101 categories and similar volume. In the UPMC Food-101 dataset, each food class has roughly 800-900 images, which are collected from Google Images searches using the food name plus "recipe". However, many images from this dataset are well sorted and some are not necessarily acquired from an eating occasion.

UEC-FOOD256 UEC-FOOD256 [39] is an extension of UEC-FOOD100. It contains images of 256 different foods. Each food class has 150 samples on average. UEC-FOOD256 also provides bounding box information.

TADA groundtruth The groundtruth dataset is created from a free-living study [158] we have conducted, in which 45 participants acquired 1453 images of 56 commonly eaten food within a week. As for now, the TADA groundtruth dataset contains over 900 food segments with labels.

Table 6.2.: RPN parameters used in finetuning

Parameters	Value
Anchor scales	8,16,32
RPN positive overlap rate	0.7
RPN batch size	256

6.5.2 Food Localization

Two datasets were used for training the food localizer: the PASCAL VOC 2012 and the UEC-FOOD256 dataset. The PASCAL VOC 2012 dataset contains 11530 images of 20 common objects and it has 27450 region annotations in the form of bounding boxes. We split the dataset in an 80%/20% fashion and trained the Faster RCNNs with different FENs (VGG-19, ResNet-50 and ResNet-101) on it. At the point, all the networks have a softmax layer with 20 outputs.

We finetuned the pretrained networks using UEC-FOOD256. Based on the bounding box data provided in UEC-FOOD256, we first converted it to the PASCAL annotation format. All the food items are simply relabeled as "food", because we are only interested in the food-vs-nonfood classification in the food localizer. Common data augmentation techniques, such as random crop, color jittering and horizontal flipping, were applied and we used a random 70%/20%/10% split for the training/validation/testing sets. A stochastic gradient descent (SGD) optimizer with momentum of 0.9 and learning rate of 0.001 was used. Some important parameters related to the RPN are summarized in Table 6.2.

Figure 6.10 demonstrates some examples of the detected food items in UEC-FOOD256. Finally, we modified the mid-level convolutional layers to adapt DCL. We used a thresholded Intersection over Union (IoU) score to evalute the localization accuracy. If a predicted food region has an $IoU > 0.6$ with a matching groundtruth

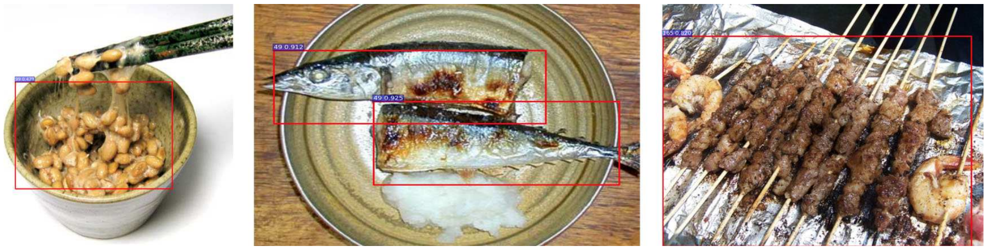


Fig. 6.10.: Examples of the localized food items in the UEC-FOOD256 dataset.

Table 6.3.: Food localization accuracy using different models

Model	Without DCL	With DCL
VGG-19	88.57%	**90.05%**
ResNet-50	93.44%	**94.52%**
ResNet-101	94.61%	**95.49%**

bounding box, we consider the food item is correctly detected. The localization accuracy is defined as $\frac{\text{\# of detected food items}}{\text{Total \# of food items}}$.

Table 6.3 shows the localization accuracy with and without applying DCLs. As we can see, the networks with DCLs consistently produce better rates. We attribute the slight performance boost to the fact that DCL has helped to detect food items with rotation changes and with various scales. The best result of 90.49% is achieved with ResNet-101 as the FEN.

More examples from the TADA groundtruth dataset are shown in Figure 6.11.

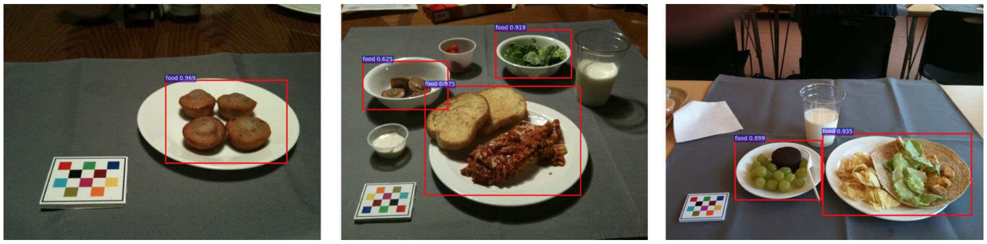


Fig. 6.11.: Examples of the localized food items in the TADA groundtruth dataset.

Table 6.4.: Top 1 and 5 food classification accuracy using different models

Model	Top 1	Top 5
GoogLeNet	76.70%	94.22%
ResNet-101	82.61%	96.11%
Inception-v3	**83.27%**	**96.35%**

6.5.3 Food Classification

As aforementioned, the proposed classification network features a 102-way softmax layer to accommodate the 101 food classes and one "background" class. We used the Food-101 dataset and all the non-food classes in the Caltech256 dataset [156] to finetune the pretrained networks. For each class including the "background", we applied a random 70%/20%/10% split for the training/validation/testing sets. An SGD optimizer with the momentum of 0.93 and a step learning rate starting from 0.01 was used.

We report the testing accuracy in Table 6.4. As ResNet-101 and Inception-v3 both have a larger capacity than GoogLeNet, they demonstrate better generalization on the testing set. Figure 6.12 shows how the validation accuracy of Inception-v3 improves over epochs. In this example, the learning rate was downgraded by 0.1 at the 25th epochs. Based on our implementation, we found that Inception-v3 converged faster and produced a better result than ResNet-101.

When we tested the finetuned Inception-v3 on the UPMC Food-101 dataset, we got 70.12% top 1 accuracy. As the UPMC Food-101 contains many non-food images, we do not consider it as a benchmark dataset.

6.5.4 Adversarial Examples

To evaluate the adversarial examples in the domain of food images, we used the network structure we introduced in Chapter 6.4. We tested the model trained on

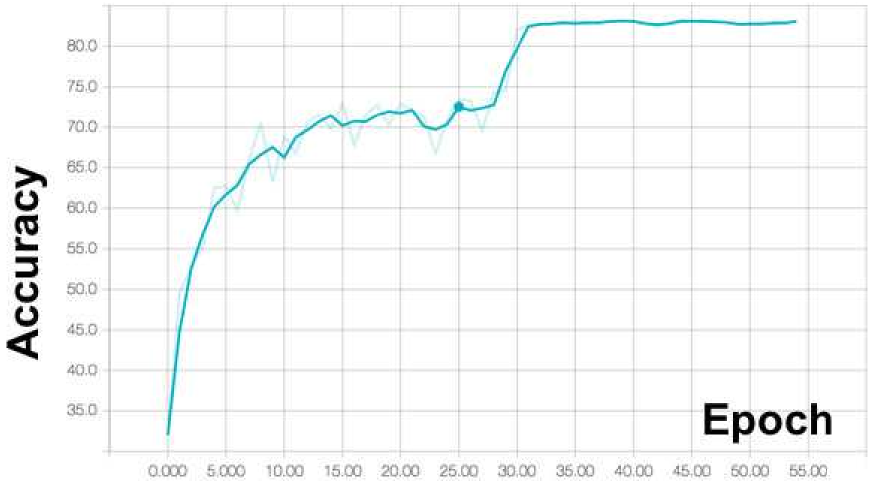


Fig. 6.12.: Validation accuracy of Inception-v3 during training.

the Food-101 dataset using untargeted attacks with FGSM and targeted attacks with JSMA.

Figure 6.13 shows an adversarial example generated by FGSM. The original image was correctly predicted as “ramen” while the tampered image was labeled as “pizza” with very high confidence, even though there is no difference between the two to our bare eyes.

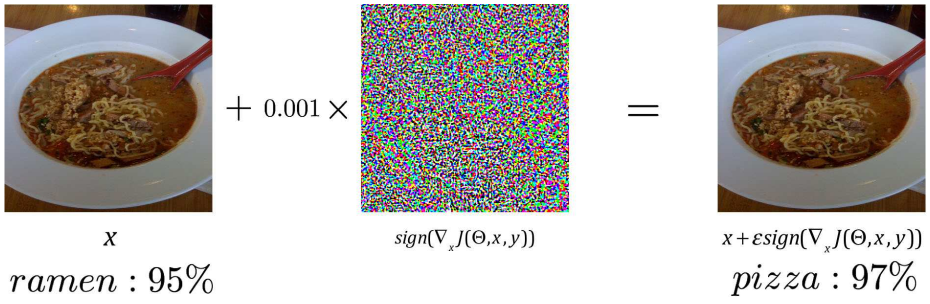

+ 0.001 ×
=
x
$ramen$: 95%
$sign(\nabla_x J(\Theta, x, y))$
$x + \varepsilon sign(\nabla_x J(\Theta, x, y))$
$pizza$: 97%

Fig. 6.13.: An adversarial example generated by FGSM. The original and tampered images are classified with high confidence to different food categories.

In Table 6.5, we report the error rates and the average confidence scores using different FGSM attacks. While the attacks with ϵ ranged from 0.001 to 0.1 all produce

Table 6.5.: Error rates and confidence scores after FGSM attacks with various ϵ

ϵ	Error rate (%)	Average Confidence (%)
0.001	85.5	89.3
0.005	86.8	86.0
0.01	89.2	82.7

Table 6.6.: Error rates and confidence scores of three example food classes after a JSMA attack

Food class	Error rate (%)	Average Confidence (%)
ramen	90.1	88.9
steak	91.8	89.7
sushi	87.4	92.2

high error rates with high confidence scores, we can see that the confidence declines when the attack is stronger.

To evaluate the targeted attack with JSMA, we used the testing set of three food classes in the Food-101 dataset. Table 6.6 summarizes the results. With JSMA, we were able to generate an image that gets misclassified to a specific food class. In this case, we created "ramen" images that were mislabeled as "steak" and "steak" as "sushi" and so on.

In the future, we would like to investigate these adversarial examples using the localization networks and also use them to help the networks to overcome overfitting issues.

7. THE USE OF CONTEXTUAL INFORMATION IN FOOD ANALYSIS

7.1 Related Work on Using Contextual Information

“Context” refers to any prior knowledge that is not derived from the image pixel values [181]. The use of contextual information has gained attention in psychology and computer vision with respect to its effects on visual search, localization and recognition [181–183]. Integrating contextual information with visual information in an object categorization framework is a challenging task [184]. Semantic object context was used in post-processing to reduce visual ambiguity in [183]. Classification techniques, such as boosting [185] and Logistic Regression [186], conditional random field [183] have been developed to use contextual information in order to maximize the classification performance.

We consider the contextual dietary information as the non-pixel data that yields additional information about a user’s diet. Examples of contextual information in dietary assessment include the time, date, and location (GPS coordinates) of a meal occasion, the dietary patterns or combinations.

Our previous work on food classification has shown that there are several issues that need to be addressed [10]. These include the inability to differentiate visually similar food items, e.g. diet coke vs. regular coke, nonfat milk vs. 2% milk, solely based on their appearance in the image. Another issue is the selection of training data for different classes. Increasing the number of food training classes could cause a drastic increase in the food classification error. Using contextual dietary information the classifier can assign different weights to the food classes that are more relevant to what are commonly eaten by the individual at similar times, dates or locations. For example, we can learn that an individual is more likely to have scrambled eggs in the morning rather than in the evening from the temporal data. GPS information is able to indicate where a person has the meal, whether at home, at work or in a restaurant. Assuming that people consume different foods at home/work compared to any meal served in a restaurant, GPS data can be treated as a priori in the classification process. Thus, the contextual information can reduce the number of classes that the classifier has to select from and hence can learn the dietary habits of the participant.

There has been work in using contextual information in food image analysis. Matsuda et al. [187] proposed to use a manifold ranking method to improve food classification rate using food co-occurrence statistics. Beijbom et al [28] made use of geographic location as context and focused on identifying foods in restaurants. In previous work [184] we incorporated two types of contextual knowledge, food co-occurrence patterns and an individual's food consumption frequency for a week.

In this chapter, we extend our earlier work on food image classification [10] and on the use of contextual information [11, 45]. We show that both our segmentation-to-classification pipeline with handcrafted features and a region proposal based method with deep features benefit from the contextual data.

7.2 Image Segmentation and Food Identification

In this section, we overview how we refine the segments generated by local variation [188]. Local variation is a graph based segmentation method, in which two regions are segmented if the difference between the two regions is large relative to the internal difference within at least one of the two regions. The degree to which the difference between regions must be larger than minimum internal difference is controlled by a threshold β [188]. β roughly controls the size of the regions in the resulting segmentation. Smaller values of β yield smaller regions and favor over-segmentation. We use $\beta = 150$ in the segmentation experiment. Since the image segmentation method is limited by a particular choice of input parameters, some food items may be under-segmented, while others may be over-segmented. We seek

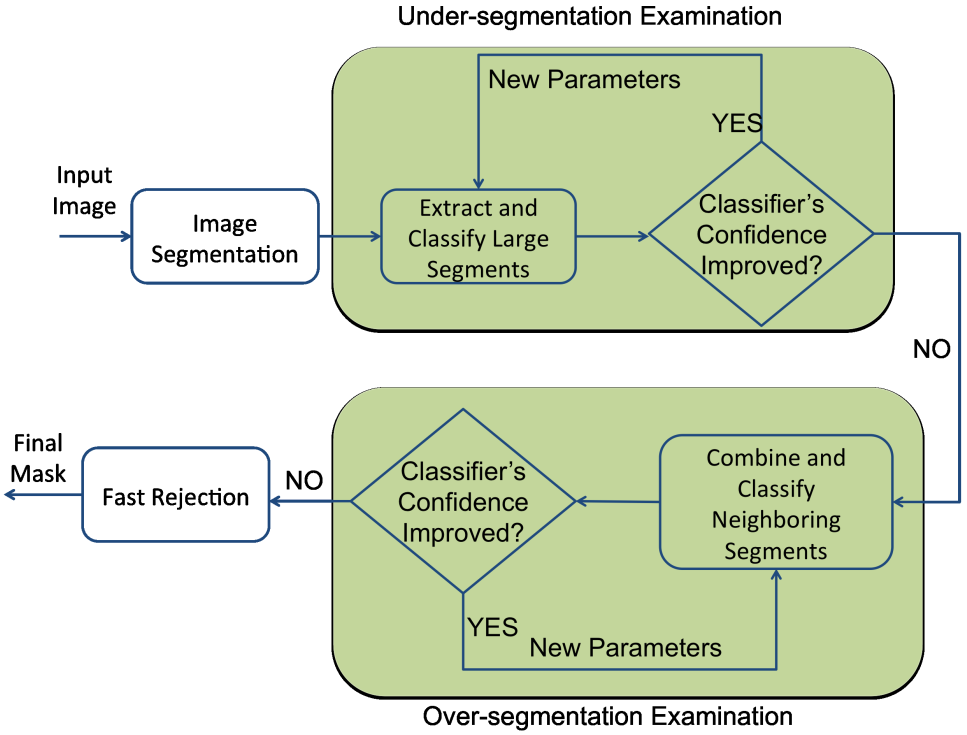

Under-segmentation Examination
New Parameters
YES
Input Image
Image Segmentation
Extract and Classify Large Segments
Classifier's Confidence Improved?
NO
Final Mask
Fast Rejection
NO
Classifier's Confidence Improved?
Combine and Classify Neighboring Segments
YES
New Parameters
Over-segmentation Examination

Fig. 7.1.: Segmentation refinement.

to overcome the segmentation problem by using classification feedback to refine the segmentation results. In our approach, the image segments are classified to a particular food label using the features extracted from that segment. The K most probable candidate classes along with their classification confidence scores are used to refine initial segmentation results. Figure 7.1 shows our segmentation refinement approach.

To detect under-segmentation, we first scan all the segments produced by the use of local variation in the image to filter out small segments. We define "small segments" as segments that contain less than 1/50 pixels of the original image. Each remaining segment is re-segmented and classified again. If the food classification confidence score is improved by re-segmentation, we accept the new segmentation; otherwise the original segmentation is kept as final segmentation. After under-segmentation examination, we update the label of the segments to $\{s_0, s_1, ..., s_{Q-1}\}$ and the corresponding food category label as $\{(c_{0,0}, c_{0,1}, ..., c_{0,K-1}), ..., (c_{Q-1,0}, c_{Q-1,1}, ..., c_{Q-1,K-1})\}$.

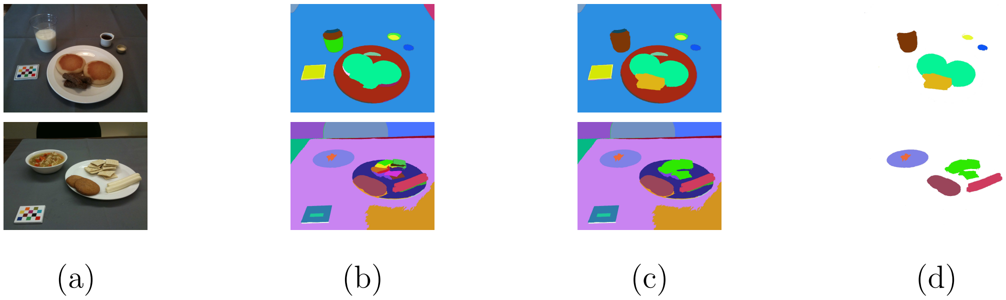


Fig. 7.2.: Examples of food image segmentation and segmentation refinement. (a) original food images, (b) initial segmentation results using the local variation segmentation, (c) segmentation refinement using food classification confidence score, and (d) final image segmentation results after fast rejection.

After under-segmentation examination, for each adjacent pair of segments, if a food category label in one segment equals to a label in the other segment, and the sum of the confidence score is greater than the highest individual score, we combine these two segments with their updated K category labels corresponding to the K largest confidence scores in the descending order. This process of over-segmentation examination is done iteratively until the overall confidence score of a segment cannot be improved. After under-segmentation and over-segmentation are examined, we may still have redundant segments, such as in the background area. We use a fast rejection step to remove these redundant segments [10]. We filter out the segments with low confidence scores from the classifier. Illustration of the complete image segmentation refinement process is shown in Figure 7.2.

Features are used for describing the characteristics of objects. An essential step in solving the food classification problem is to select suitable features to distinguish one food from another. Some foods may have very distinctive color or very distinctive patterns, but for most food items it is the combination of these aspects that make them distinctive. Previously, we have investigated various features for food classification [10, 11]. Here, we overview three types of features, color, texture and local region descriptors, among which we regard color and texture as the global features. Based on the evaluation of these feature descriptors and their combinations, we select the optimal strategy for our food image analysis system.

It should be noted that in this section we describe a set of features used in our contextual experiments. Other approaches such as deep networks [35, 189] could also be used to investigate the use of contextual information.

Color features have been extensively studied in image retrieval [190]. Some foods may exist in a wide variety of colors, but many have a distinctive color. Color information is sensitive to environmental conditions, such as changes in light source and shadows. We investigated two color descriptors, namely, Dominant Color Descriptor (DCD) and Scalable Color Descriptor (SCD) [190]. DCD is a vector of D representative colors from the *CIE-Luv* color space using the generalized Lloyd algorithm for color clustering [191, 192] and their corresponding percentages. SCD is determined by quantizing the colors in the HSV color space uniformly into 256 bins, which includes 16 levels in H, 4 levels in S, and 4 levels in V as suggested by the MPEG-7 standard [190].

Texture, similar to color, is a very descriptive low-level feature. In general, texture describes the arrangement of basic elements of a material on a surface [193, 194]. We selected two texture descriptors for food classification: Entropy-Based Categorization and Fractal Dimension Estimation (EFD) and Gabor-Based Image Decomposition and Fractal Dimension Estimation (GFD) [10]. EFD can be seen as an attempt to characterize the variation of roughness of homogeneous parts of the texture in terms of complexity [10]. GFD is based on fractal dimension estimation [10].

Local region features are described for points of interest and/or local regions. The idea is to find points in the object which can be reliably found in other samples of the same object regardless of variations between images. An invariant local region feature describes such points of interest in the same way in different images with illumination, scale and viewpoint changes. Many local region features have been proposed to represent the characteristics of points of interest [195–198]. We investigated the following two local region features for food classification: Scale Invariant Feature Transforms

Table 7.1.: List of features investigated and their types and dimensions.

Feature	**Feature Type**	**Dimension**
DCD	Color Feature	20
SCD	Color Feature	256
EFD	Texture Feature	120
GFD	Texture Feature	120
SIFT	Local Region Feature	128
MDSIFT	Local Region Feature	384

(SIFT) [195] and Multi-scale Dense SIFT (MDSIFT) [10]. Table 7.1 summarizes the features used in our experiments.

Once the food items are segmented from an eating occasion image and the features are extracted, we classify the color and texture features using K-Nearest Neighbors (KNN) [199] and the local features using the Vocabulary Tree (VT) classifier [11].

7.3 Region Proposal Based Approach with Deep Features

Since the interest in Convolutional Neural Networks (CNN) was rekindled by AlexNet [137] in 2012, the number of applications using deep networks has grown exponentially. CNNs have dominated many aspects of object classification and detection [32]. Besides, recent research indicates that the generic descriptors extracted from the convolutional neural networks are very effective [189]. The success of CNNs is largely attributed to big data and carefully designed models. In terms of food image analysis, some researchers [200] focused on improving the network structure by considering the food structure in the image, or "vertical food layer". However, they did not utilize any contextual information to improve classification for foods that do not have an obvious structure in their appearances.

In the section, we describe a region proposal based method to identify multiple food items in an image. In contrast to the segmentation-to-classification pipeline we discussed in Section 7.2, deep features are extracted from redundant proposed regions instead of segments. Then, support vector machine (SVM) [189] is used to classify each region as either a specific food item or background. Similar to RCNN [168], we adopted selective search [170] as the generic region proposal method. We finetuned VGG-16 [150] on Food-101 dataset [27] and used the output from the first fully connected layer as the deep features.

Before we forward propagate the region proposals through the network, we first run a fast rejection to eliminate tiny regions and regions with large aspect ratios. The fast rejection is based on the assumption that the food items in a food image usually occupies the majority of the scene.

We used the PyTorch [201] implementation of the VGG-16 [150] to obtain the 4096-dimensional features from each region proposal that was not rejected. In order to convert a region proposal to the dimension compatible with the deep network, we proposed a random 10-crop technique. Since VGG-16 requires the input image of 224×224, for any region proposal, we first resize it so that the shorter dimension of the region is 224. Then we randomly select 10 224×224 cropped regions from the resized proposal. Features are computed by propagating a mean-subtracted 224×224 RGB image through the network.

Finally, we used an SVM to classify food items and the background. For regions that are classified as a certain food class with greater than 75% confidence, we apply non-maximum suppression to select the best proposal. We combine the majority vote from the 10-crop technique with the confidence score from SVM to finalize our prediction. As improving deep features or region proposal methods is not the focus of this chapter, we only show the experimental result using our own dataset in Section 7.5 as a comparison to the method we discussed in Section 7.2. More importantly, we show that the contextual information can be integrated as easily in the region proposal based approach as in the previously discussed segmentation-to-classification pipeline.

7.4 Context Refinement

Contextual information has gained more attention in image analysis and computer vision in the past few years [182, 183, 202–206]. Context such as semantic, spatial images and poses has been proved to be effective for natural images. Examples of contextual information in the dietary applications include the date, time and location of the eating occasion, who the subject is eating with, and personal eating habits. In this section, we overview our previous approaches for integrating contextual information which the participant supplies to the system either explicitly or implicitly [45] and propose a new approach for combining temporal eating information and food co-occurrence into a personalized learning model. Note that the contextual information we investigate are independent of the classifier and can be used with other types of machine learning techniques (e.g SVM and deep networks [35, 189]).

7.4.1 Temporal Dietary Information

We explore temporal information of food images to generate the preference of different food classes based on time of an eating occasion. People usually eat different types of foods with regard to the time of a day, such as breakfast vs. dinner. We incorporate this contextual dietary information to assign a weight to different food classes.

We divide eating time into three time intervals: *12am - 11am*, *11am - 4pm*, and *4pm - 12am (midnight).* For example, from our "free-living" dietary assessment study [45] the food consumption frequency of these three time intervals are shown in Figure 7.3, Figure 7.4, and Figure 7.5, respectively. We can see unique food consumption patterns for different time intervals. For example, from *12am* to *11am*, "Bagel", "English Muffin" and "Pancake" are more likely to be consumed than other foods such as "Chicken Wrap," "Frozen Meal Meatloaf" and "Ham Sandwich." From *11am* to *4pm*, people tend to eat more "Ham Sandwich" and "Potato Chips" than earlier in the day. When it comes to *4pm* to 12am, there is a significant increase

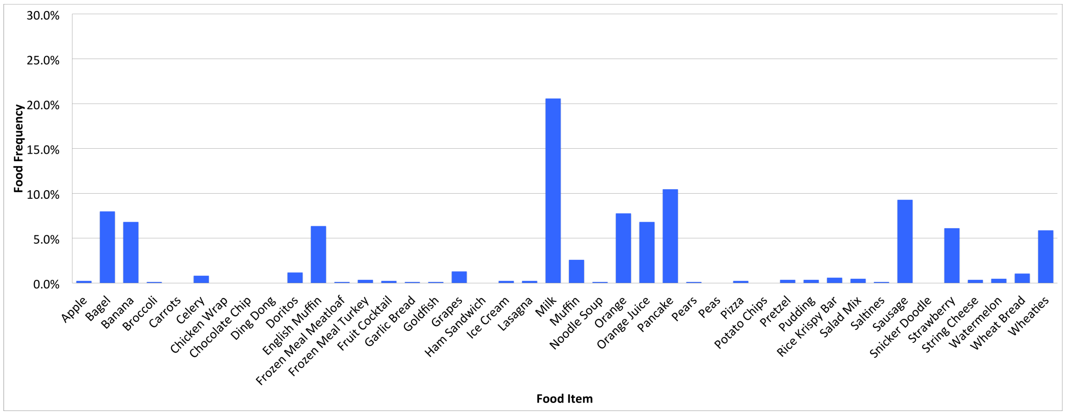


Fig. 7.3.: Examples of food consumption preference between 12am (midnight) and 11am.

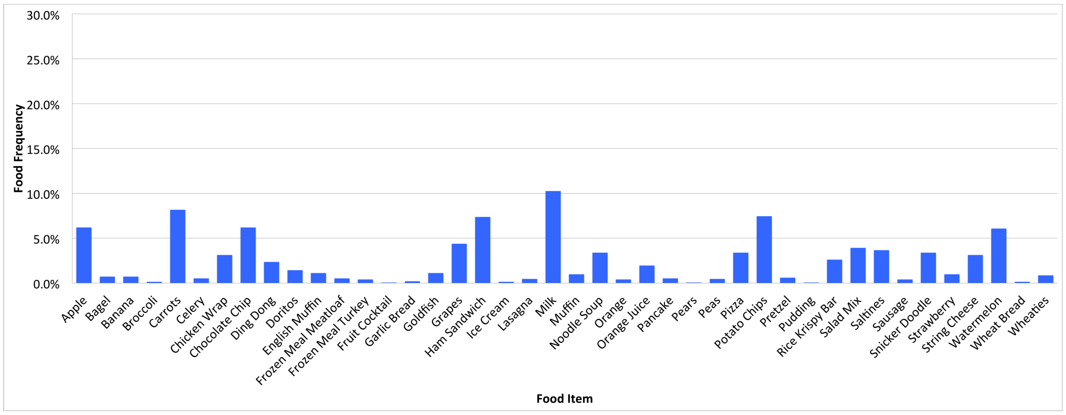


Fig. 7.4.: Examples of food consumption preference between 11am and 4pm.

in the consumption of “Garlic Bread” and “Lasagna.” Given a participant’s food consumption frequency over time, we assign different weights to different food classes according to the eating occasion time.

7.4.2 Food Co-Occurrence Patterns

A food co-occurrence pattern describes the likelihood of food combinations. It is the joint probability of food items existing together in a single eating occasion [45]. Semantic context can provide valuable information for improving classification. In

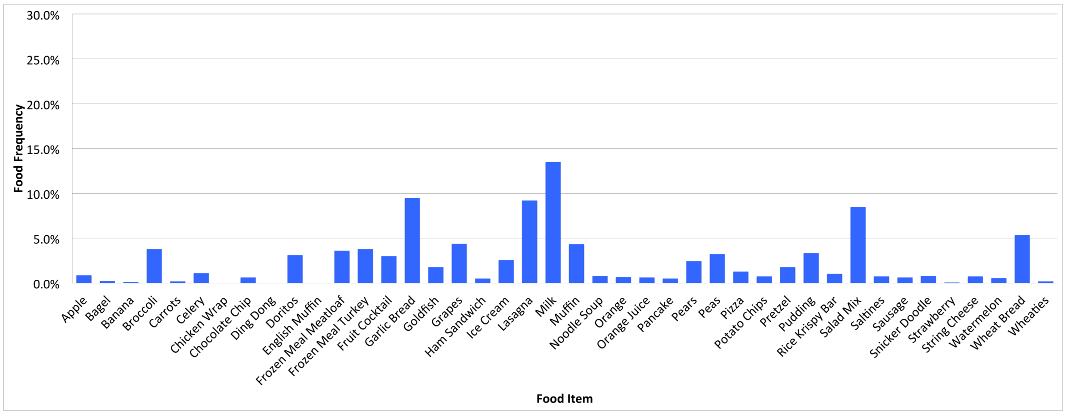


Fig. 7.5.: Examples of food consumption preference between 4pm and 12am (midnight).

this section, we describe the use of the co-occurrence of food items in order to reach a labeling agreement for all the segmented regions in an image. The goal is to detect potential misclassifications and refine the classification results that were obtained by using only visual features.

After image segmentation and food classification, an eating occasion image I is segmented into multiple regions $s_0, ..., s_q, ..., s_{Q-1}$. A segment s_q is assigned K food labels. A classification confidence score $\phi(c_{q,k})$ measuring the probability that any food label matches the segment s_q based on the distance of the visual features between the segmented region and the training data. We now want to adjust the food labels to achieve maximal global contextual agreement with respect to the food co-occurrence pattern given the constraints of the segments' visual features. For example, in most cases "fries" has a higher contextual agreement with "ketchup" than with "pepper."

Graphical models provide a simple way to visualize the structure of a probabilistic model. Since the number of food segments in an eating occasion is relatively small, we construct a weighted complete digraph between all segments [207]. In our graph, each node in the weighted complete digraph represents a segment and its associated food labels from the food classification results. Therefore, the graph contains Q nodes and each contains K food labels. Obviously, one node has $Q-1$ outgoing edges and $Q-1$ incoming edges.

The food co-occurrence probability of food label $c_{j,k}$ given food label $c_{i,k'}$, denoted as $P(c_{j,k}|c_{i,k'})$ is defined as follows,

$$P(c_{j,k}|c_{i,k'}) \quad = \max\{P(c_{j,k}|c_{i,1}), ..., P(c_{j,k}|c_{i,K-1})\} \tag{7.1}$$

where k and k' independently indeces K. Then the influence of segment i on the k^{th} food label of segment j, $v(c_{j,k}|s_i)$, is calculated as:

$$v(c_{j,k}|s_i) = \phi(c_{i,k'})P(c_{j,k}|c_{i,k'}) \tag{7.2}$$

where $\phi(c_{i,k'})$ is the classification confidence score of food label $c_{k'}$ in the i^{th} segment. Finally, the weight of the edge from node i to node j, $\mathbf{v}_{ij}$, is defined as an influence vector indicating how much influence the food labels in segment i have on all the food labels in segment j. An influence vector of K-dimension is computed from the food occurrence pattern as follows,

$$\mathbf{v}_{ij} = [v(c_{j,0}|s_i), ..., v(c_{j,k}|s_i), ..., v(c_{j,K-1})|s_i)] \tag{7.3}$$

To estimate the co-occurrence probability, we first construct a food co-occurrence matrix M_{FCO} that contains the food co-occurrence counts among food labels in the training set of the database [45]. Figure 7.6 shows an example of a food co-occurrence matrix. The entry (i,j) in a food co-occurrence matrix is the number of times that food λ_j is in an eating occasion image when food λ_i is in the image [45].

Figure 7.6 illustrates the structure and content of a food co-occurrence matrix. As we can see, some food items have a high probability of existing together in the same image, e.g. “Wheaties” with “Milk,” “Garlic Bread” with “Lasagna;” while some food items rarely appear together in the same image, e.g. “Carrots” with “Celery.” Note that the co-occurrence matrix is trained on training data, where we have perfect segmentation and food labels from our dietary studies. The matrix is only updated when we receive a participant’s confirmation from the review process of the TADA system where the participant can confirm, change or add food labels [45].

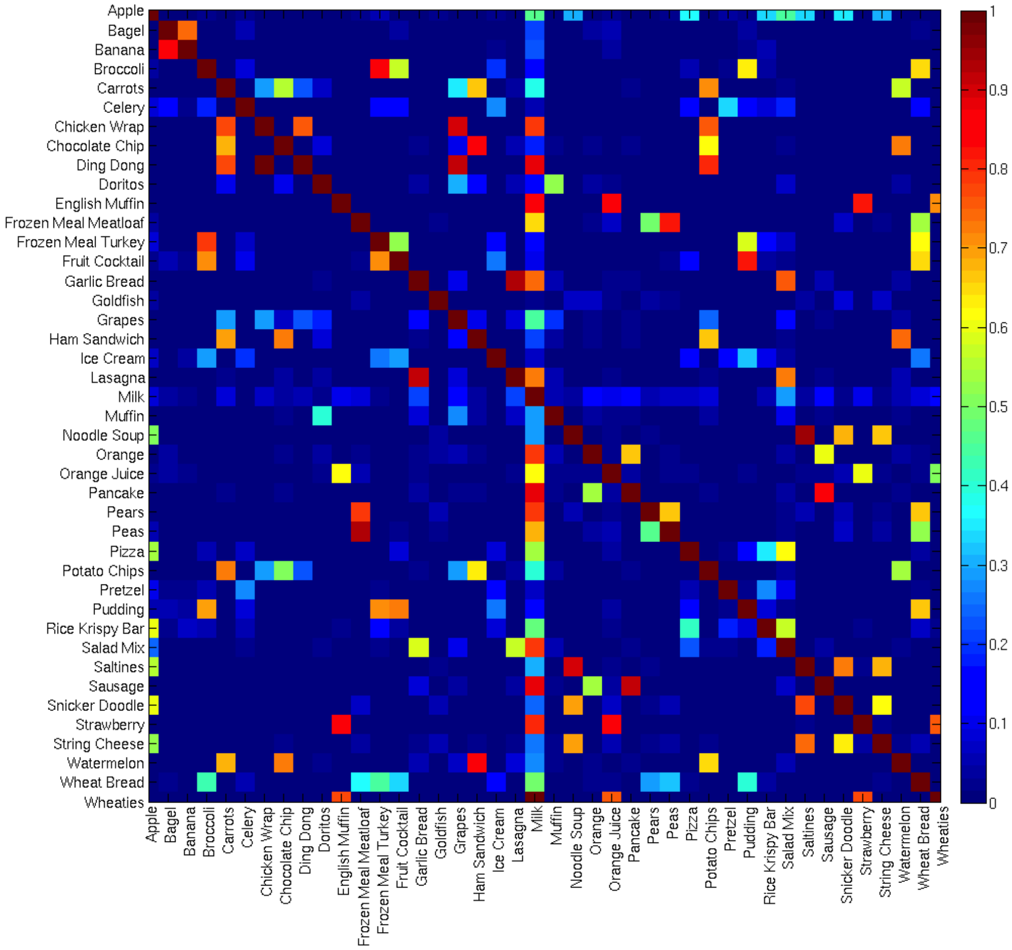


Fig. 7.6.: An example of food co-occurrence patterns.

So far we have found the influence vector from s_i to s_j. Following the same approach, we find the influence vectors from all other segments to s_j. The next step is to find the total influence on s_j from all other nodes in the graph. We propose to use the maximal influence vector $\mathbf{w}_j$:

$$\mathbf{w}_j = [\max(\{\mathbf{v}_{ij}(0), i \neq j\}), ..., \max(\{\mathbf{v}_{ij}(K-1), i \neq j\})]$$

where $\{\mathbf{v}_{ij}(k), i \neq j\}$ is the set containing the k^{th} element of each of the influence vectors that point to node j. We choose the largest influence given to each food label of node j as the final influence vector $\mathbf{w}_j$. We finally update the food classification confidence score of each food label of node j as follows:

$$\phi'(c_{j,k}) = \phi(c_{j,k})(w_j(k) + \epsilon) \tag{7.4}$$

where $\epsilon > 0$ and $w_j(k)$ is the k^{th} element of $\mathbf{w}_j$. The term ϵ is used to avoid setting $\phi'(c_{j,k})$ to zero when $w_j(k) = 0$. The max influence of one segment on another is constrained by the max confidence score of the food label. For each adjacent pair of segments, we only accept the new segmentation if a food category label in one segment equals to a label in the other segment, and the sum of the confidence score is greater than the highest individual score.

After the confidence scores for segment i is updated to $\{\phi'(c_{i,0}), ..., \phi'(c_{i,K-1})\}$, we update the order of food labels accordingly, with the Top 1 food label being the one associated with the largest updated confidence score, and the Top M food label being the one associated with the M^{th} largest updated confidence score.

7.4.3 Personalized Learning Model

The goal of a personalized learning model is to improve food classification by using dietary preferences. For example, if it learns that a person prefers diet coke and he/she never drinks regular coke from his/her dietary history, the personalized learning model will adjust the prediction of different Coke products if a classifier initially assigns similar confidence scores to those classes.

Figure 7.7 illustrates a list of food consumption frequency patterns for various participants in our free-living study. For example in Figure 7.7, most of the participants shown here drink milk quite frequently but participant 36 rarely drinks milk. We can also tell that the favorite fruit of participant 3 is “Grapes;” the favorite drink for participant 35 is “Orange Juice;” and the favorite food for participant 36 is “Lasagna.” The figure shows the differences in individual eating habits using food consumption

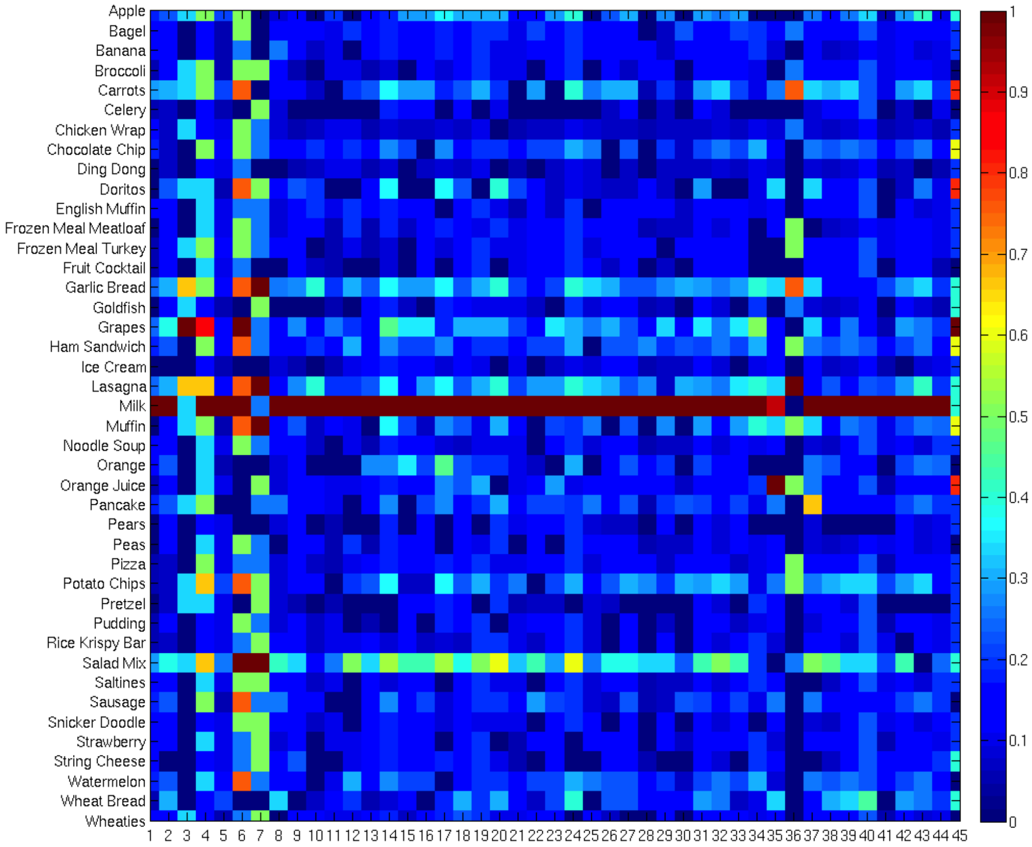


Fig. 7.7.: An example of food consumption frequency for various participants. **Horizontal axis** shows the IDs of participants and **vertical axis** represents various food items.

frequency for this subset of foods. The food consumption frequency of food item λ_i for a participant S_j is:

$$F(S_j, \lambda_i) = \frac{\gamma_i(S_j, \lambda_i)}{\sum_k \gamma_k(S_j, \lambda_i)} \text{ for } k = 1, ..., K \tag{7.5}$$

where γ_i is the food consumption counts of a participant and K is the number of food classes.

The personalized learning model takes into account both temporal dietary information and food co-occurrence patterns. Figure 7.8 shows how we propose to do context-based classification refinement. Given a set of labeled segments,

$$\{(c_{0,0}, ..., c_{0,K-1}), ..., (c_{q,0}, ..., c_{q,K-1})\}$$

with associated confidence scores

$$\{(\phi(c_{0,0}), ..., \phi(c_{0,K-1})), ..., (\phi(c_{q,0}), ..., \phi(c_{q,K-1}))\}$$

, the food co-occurrence pattern generates an updated confidence scores for each segment in the image,

$$\{(\phi'(c_{0,0}), ..., \phi'(c_{0,K-1})), ..., (\phi'(c_{q,0}), ..., \phi'(c_{q,K-1}))\}$$

.

In the temporal information block of Figure 7.8, we use recursive Bayesian estimation to incrementally learn a participant's dietary pattern [45,208,209]. We model whether a participant, S_j eats a particular food, λ_i in time internal, V, as a Bernoulli trial,

$$W = \begin{cases} 1, X \\ 0, 1 - X \end{cases} .$$

where $W = 1, X$ represents S_j eats λ_i in V with a possibility, X, and X is assumed to follow a Gaussian-like distribution with the support from 0 to 1. As discussed in Section 7.4-A, we used three time intervals *12am - 11am*, *11am - 4pm*, and *4pm - 12am (midnight)*.

We would like to estimate the probability, P_{λ_i}, that a participant, S_j, will eat a particular food, λ_i on the next day given the history [45]. Let $p_{\lambda_i}(x^n)$ be the probability density function (PDF) representing S_j eats λ_i, in the time interval V on the n^{th} day, and z^n be the observation whether S_j eats λ_i in V on the n^{th} day [45].

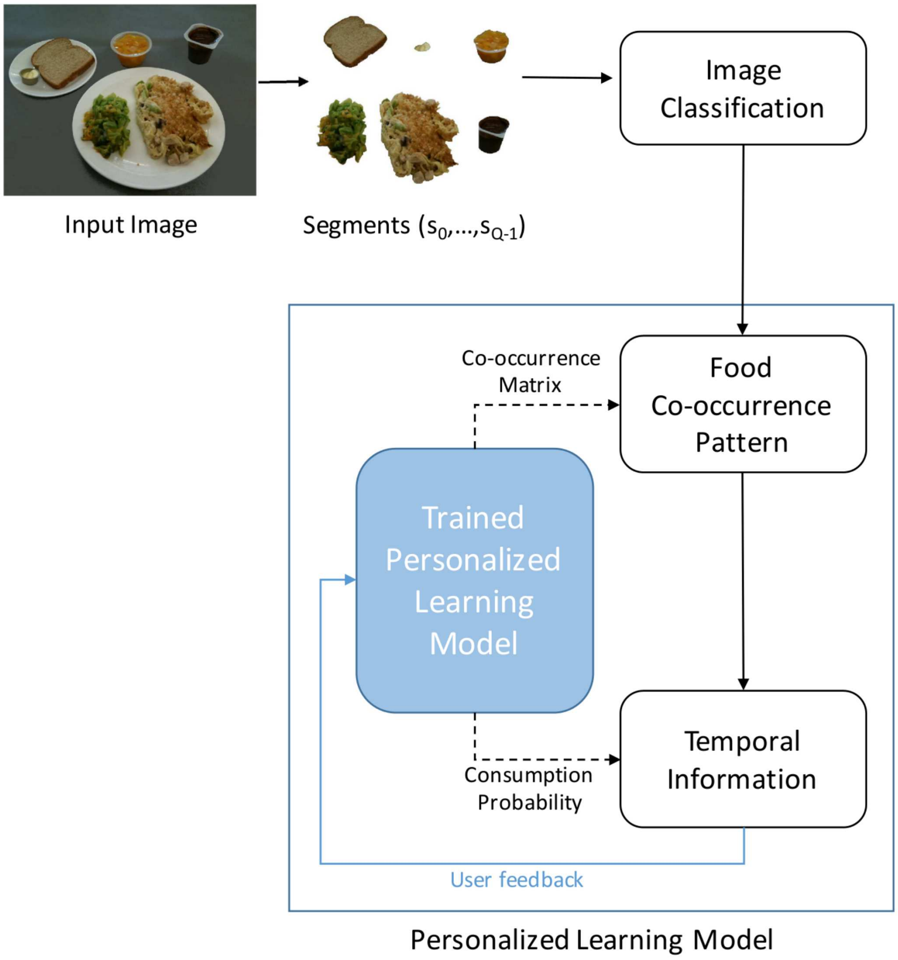

Input Image
Segments (s_0,…,s_{Q-1})
Image Classification
Co-occurrence Matrix
Food Co-occurrence Pattern
Trained Personalized Learning Model
Consumption Probability
Temporal Information
User feedback
Personalized Learning Model

Fig. 7.8.: The use of contextual dietary information in food classification refinement.

The following equations describe the posteriori update step in the recursive Bayesian network,

$$\begin{aligned} p_{\lambda_i}(x^n|z^{1:n},V) &= \frac{p_{\lambda_i}(z^n|x^n,V)p_{\lambda_i}(x^n|z^{1:n-1},V)}{p_{\lambda_i}(z^n|z^{1:n-1},V)} \\ &= \frac{\text{likelihood} \times \text{prior}}{\text{normalization}\ \ \text{term}} \ . \end{aligned} \tag{7.6}$$

Initially, $p_{\lambda_i}(x^1|V)$ is assumed to have a Gaussian-like distribution centered at 0.5 with unit variance. If the participant eats λ_i in V on the n^{th} day, $p_{\lambda_i}(z^n|x^n,V)$ becomes the Gaussian-like distribution centered at 1 with unit variance, otherwise the distribution centers at 0. $p_{\lambda_i}(x^n|z^{1:n},V)$ is used to predict $p_{\lambda_i}(x^{n+1}|z^{1:n},V)$ and the

PDF is computed by multiplying the likelihood and prior followed by normalization between 0 and 1. On the $n+1^{\text{th}}$ day, the optimal estimate of $P_{\lambda_i}(V)$ is computed as $P_{\lambda_i}(V) = \arg\max_x p_{\lambda_i}(x^n|z^{1:n}, V)$.

For all the foods in the training dataset, we have a set of probabilities,

$$\{P_{\lambda_0}^{n+1}(V), \ldots, P_{\lambda_{K-1}}^{n+1}(V)\}$$

where N is the total number of food categories. We further define the context-based confidence scores (CCS) to be:

$$\begin{aligned}\Psi^{n+1}(V) &= \left[\psi_{\lambda_0}^{n+1}(V), \ldots, \psi_{\lambda_{K-1}}^{n+1}(V)\right]^{\mathrm{T}} \\ &= \left[\omega P_{\lambda_0}^{n+1}(V), \ldots, \omega P_{\lambda_{K-1}}^{n+1}(V)\right]^{\mathrm{T}} \quad . \end{aligned} \tag{7.7}$$

where ω controls the trust weight we assigned to the context-based decisions. For each image segment, the CCS associated with the Top M food labels, $(\psi_{i,0}, ..., \psi_{i,M-1})$, can be obtained from $\Psi^{n+1}(V)$. The final confidence scores are calculated using a strategy of majority vote,

$$\begin{aligned}(\phi''(c_{i,0}), ..., \phi''(c_{i,M-1})) &= (\phi'(c_{i,0}), ..., \phi'(c_{i,M-1})) \\ &\quad + (\psi_{i,1}(V), ..., \psi_{i,M-1}(V)) \\ &= (\phi'(c_{i,0}), ..., \phi'(c_{i,M-1})) \\ &\quad + (\omega P_{i,1}(V), ..., \omega P_{i,M-1}(V)) \end{aligned} \tag{7.8}$$

ω, also in Equation 7.7, is set to be $1/h$ of the maximum automatic analysis based confidence score. In our experiments, we observed best results when h was set to 4-5. The food labels are updated again according to the new confidence scores to generate the final food labels $\{(c''_{0,0}, ..., c''_{1,M-1}), ..., (c''_{q,0}, ..., c''_{q,M-1})\}$.

7.5 Experimental Results

7.5.1 Experiment Setup and Datasets

We evaluate our system using a free-living dietary study where we provided some foods to the participants and were flexible relative to their preferences regarding how

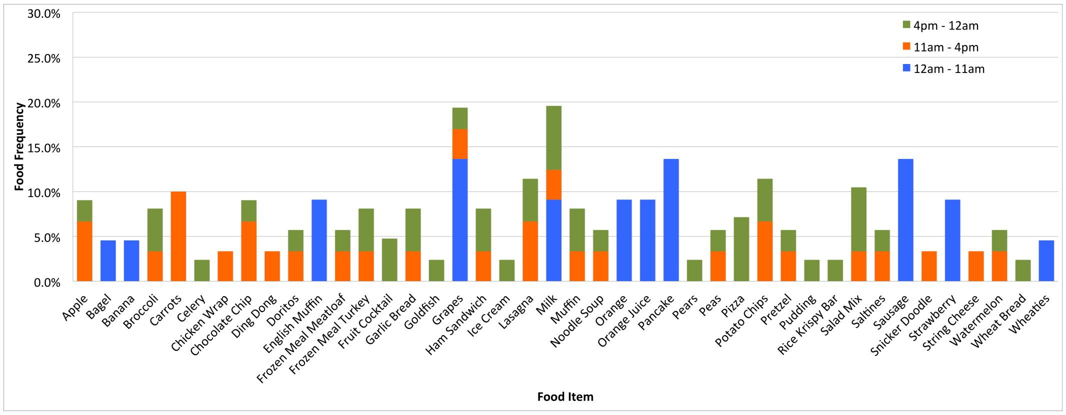


Fig. 7.9.: An example of food consumption frequency by a participant with the integrated time information integrated. The colors indicated different eating time intervals.

and when foods were eaten [45]. In addition, we encouraged the participants to select their own favorite foods if not provided [158]. They used the TADA mobile application to record their eating occasions and all the images were uploaded to our back-end server and then analyzed using the techniques we described in the previous sections.

This study consists of 45 participants. Each participant was asked to acquire eating occasion images at each eating occasion for a 7-day period. In total 1,453 eating occasion images were collected in the study and 42 commonly eaten food items were analyzed. Moreover, the dataset contains rich contextual information, such as participant feedback, temporal data, GPS location and nutrient information.

We used the free-living dataset as it is for evaluating features and classification. To evaluate the personalized learning model on a day-to-day basis, we selected participants in the free-living study with similar food consumption patterns to construct three datasets. We measure the similarity using the Euclidean distance between each food consumption pattern and used *K-means* for clustering. For example, one of the datasets contains 119 food images from participant 14, 17, 20 and 32. As illustrated in Figure 7.7, participant 14, 17, 20 and 32 all show relatively high consumption frequency of milk, mixed salad and lasagna. Each dataset features a different food consumption pattern and contains approximately 120 images. We labeled them as *Dataset 1, 2 and 3* corresponding to *User 1, 2 and 3.* Milk, lasagna, mixed salad and garlic bread are the most frequently consumed foods in *Dataset 1*, while *Dataset 2* does not have any frequently consumed foods except milk. *Dataset 3* represents a significant dietary pattern change within a month. The first three weeks in *Dataset 3* have similar food consumption style as *Dataset 1.* However, the eating pattern of the last week was selected to be noticeably different.

In addition, we use a subset of the 1453 images for evaluating the performance of the region proposal method with deep features. Due to the limitation of the bounding box groundtruth we have in the free-living dataset, the subset consists of 60 images of 24 food classes and we denote it as *Dataset 4*. On average, each image contains 4 instances of the 24 food classes. The bounding box groundtruth of the 24 food classes from the rest of the free-living dataset are used for training.

7.5.2 Feature and Classification

As we discussed in Section II, we classify food items based on the automatic segmentation result and the features extracted from each segment. The food identification accuracy is defined as: $accuracy = TP/(TP + FP/M + FN)$ where TP indicates True Positives (correctly detected food segments); FP indicates False Positives (incorrectly detected food segments or misidentified foods); FN indicates False Negatives (food not detected). Finally, M refers to the identification accuracy order. If one food classification label is generated for each image segment, then $M = 1$. In our implementation, after image segmentation and food classification, each image segment is assigned 4 food categories (or classes) with the 4 largest class labeling confidence scores ($M = 4$).

The classification accuracy of a single feature is discussed in [11]. From the results, we see that color features achieve better classification accuracy compared to texture

Table 7.2.: Food classification accuracy from feature combination.

Features	Top 1 Accuracy	Top 4 Accuracy
DCD + MDSIFT	60.9%	83.27%
DCD + MDSIFT + SCD	62.9%	85.1%
DCD + MDSIFT + SCD + SIFT	64.5%	84.2%
DCD + MDSIFT + SCD + SIFT + EFD	63.5%	83.4%
DCD + MDSIFT + SCD + SIFT + EFD + GFD	62.9%	82.8%

features. DCD outperformed all other features. This suggests that in general we can represent the color content of food items using only a few colors, which is consistent with color representation of general objects [210]. As to local features, the *MDSIFT* feature achieves better food classification results than the *SIFT* feature.

We combine the confidence scores of food labels that belong to the same food class and choose M food classes with Top M confidence scores. Each feature is assigned a weight from 0 to 1 based on our training experiment and the final confidence score is obtained by a weighted sum model. The food classification accuracy of Top 1 and Top 4 most probable food classes is shown in Table 7.2.

Table 7.2 summaries the food classification results after combining multiple features. Based on the performance of feature combinations and complexity consideration, we choose three features, namely, DCD, MDSIFT and SCD, in our food classification system. The Top 1 and Top 4 food classification accuracy for each food item using the combination of these three features is shown in Figure 7.10. When tested on Intel Xeon X5550 CPU, it usually takes 30-70s for one image to complete segmentation and classification depending on the number of food items in the image.

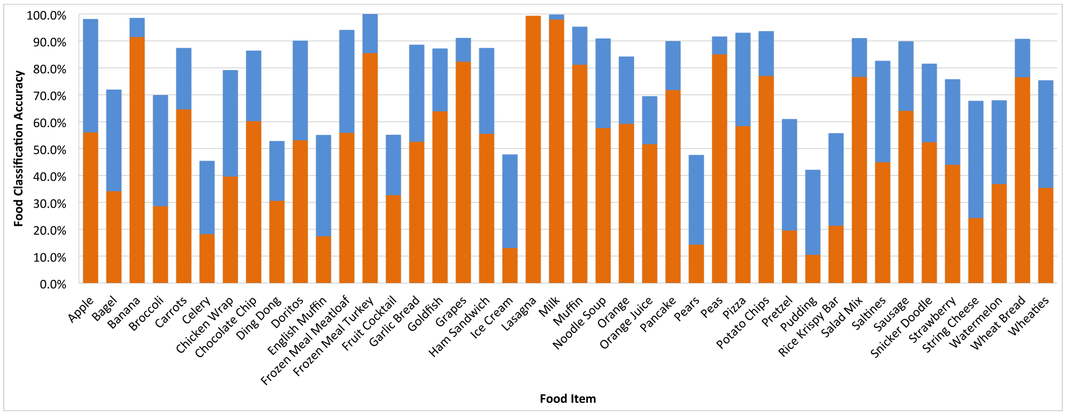


Fig. 7.10.: Top 1 and Top 4 food classification accuracy for each food item with three features fused together, DCD, MDSIFT and SCD. The top of the orange bar: Top 1 classification accuracy; the top of the blue bar: Top 4 classification accuracy.

We fully understand that automatic identification of food items in an image is not an easy problem and we will not be able to recognize every type of food. The way of packaging or the way the food is served will present problems for automatic recognition. Also, some food items are inherently difficult to identify due to their visual similarity in the feature space. In some cases, even if a food is undetected or not correctly identified, it may not make much difference with respect to the energy or nutrients consumed. For example, if we fail to detect water in an eating occasion image, it will have little impact on the estimate of the energy or nutrients consumed in the meal due to the low energy content of water. Similarly, if our system identifies a “brownie” as “chocolate cake,” there is no significant difference in energy or nutrients consumed.

7.5.3 Region Proposal Based Approach with Deep Features

To evaluate the performance of the region proposal based method, we selected 24 food classes with relatively sufficient bounding box groundtruth as mentioned in Section 7.5.1. We start with 20 bounding boxes on average for each class and we augment the training set by applying minor shifting to the original groundtruth. We end up getting more than 100 bounding boxes for each class. We validate the SVM model using a 5-fold cross-validation. For each bounding box groundtruth, we first resize it so that the shorter dimension of the region is 224 while keeping the aspect ratio. Then the deep features are extracted from the 224×224 center-cropped region. In our experiment, we chose the SVM model with the Radial Basis Function kernel where $C = 1000$ and $\gamma = 0.001$. *Dataset 4* was used to evaluate the proposed method. For each detected region, we consider it correct if it has more than 80% overlap with the groundtruth. We report 52.4% detection and classification accuracy.

During training time, we validate the SVM model using a 5-fold cross-validation. For each bounding box groundtruth, we first resize it so that the shorter dimension of the region is 224 while keeping the aspect ratio. Then the deep features are extracted from the 224×224 center-cropped region. In our experiment, we chose the SVM model with the Radial Basis Function kernel where $C = 1000$ and $\gamma = 0.001$. Figure 7.11 demonstrates the precision and recall [159] of the classification accuracy using the 5-fold cross-validation. To summarize, we achieve approximately 84% Top 1 accuracy for both precision and recall.

At test time, *Dataset 4* was used to evaluate the proposed method. For each detected region, we consider it correct if it has more than 80% overlap with the groundtruth. We report 52.4% detection and classification accuracy.

7.5.4 Contextual Refinement

We conducted two experiments to validate the personalized learning model. First, we tested on the same 1453 eating images used in the feature and classification experiment by assuming the food assumption frequency and co-occurrence pattern are known. If the food co-occurrence pattern is given, the Top 1 and Top 4 food identification accuracy increased to 65.3% and 85.9% compared to 62.9% and 85.1% without contextual information. The accuracy is further improved to **71.4%** and **88.3%** with

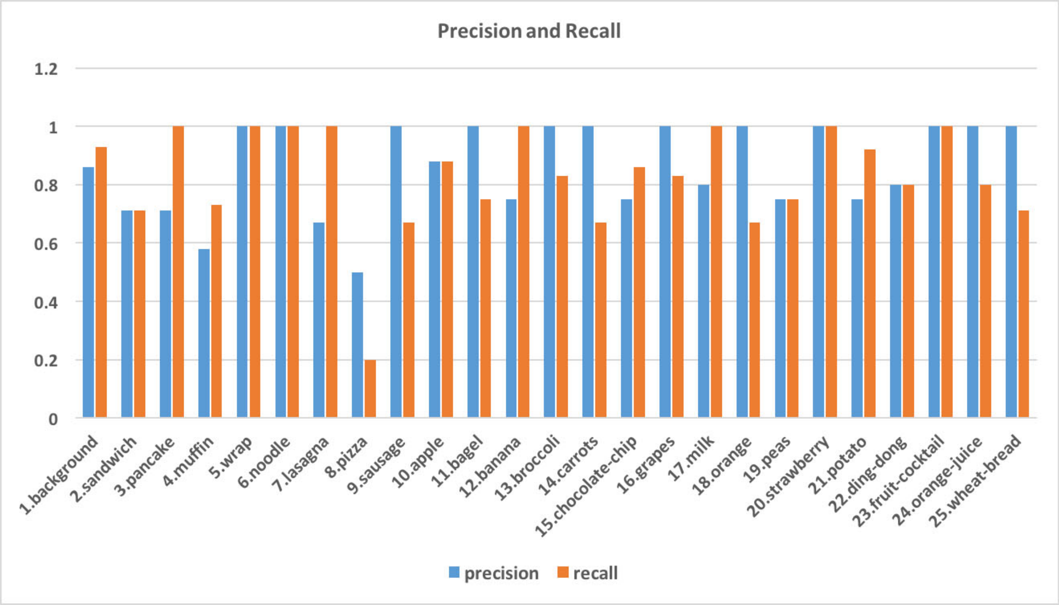


Fig. 7.11.: Precision and recall of each food class using 5-fold cross-validation.

both food assumption frequency and co-occurrence pattern. The food identification accuracy for each food item is shown in Figure 7.12. Comparing food classification accuracy obtained after contextual refinement (Figure 7.12) and before contextual refinement (Figure 7.10), we can see that most of the food items in our dataset achieve a better classification accuracy. Similarly, we tested the region proposal based method on *Dataset 4*. We achieved 57.5% detection and classification accuracy with contextual information by assuming the food assumption frequency and co-occurrence pattern are given compared to 52.4% without contextual information.

Next, we would like to examine how the personalized learning model behaves day by day over a month. As we mentioned in Section 7.5.1, we have created three datasets from the free-living study, each of which features a different food consumption pattern. In the following experiment, we use the method discussed in Section 7.2 as it shows better result than the region proposal based method in the previous evaluation.

Figure 7.13(a) shows how the recursive Bayesian network updates the prediction probabilities for three example food items in *Dataset 1* from 12 am to 11am. On Day 1, every food has the same prediction. In the end, the prediction of milk, orange juice

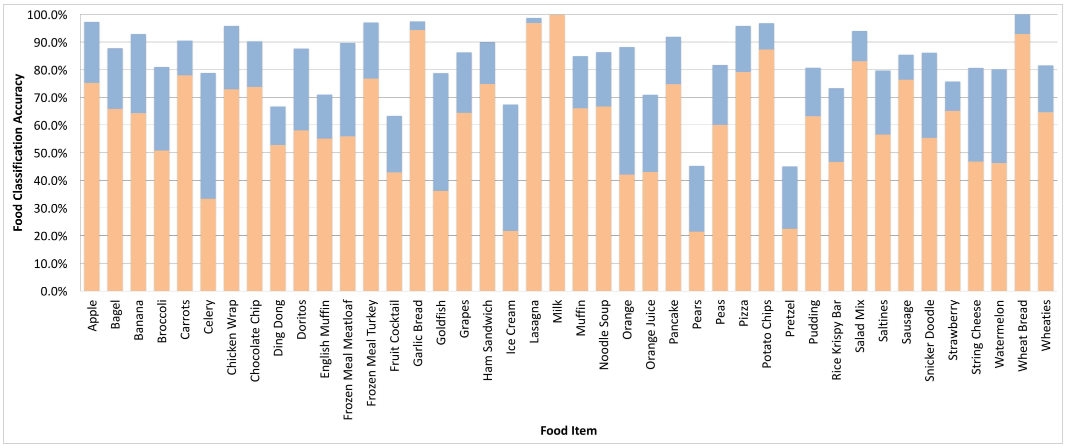


Fig. 7.12.: Food identification accuracy for each food item after integrating contextual dietary information.

and muffin converges to 0.51, 0.19 and 0.06 respectively. Figure 7.13(b) compares the prediction of milk among all three datasets from 11am to 4pm. It is clear that *User 1* consumes milk during lunch time more frequently than other *Users*.

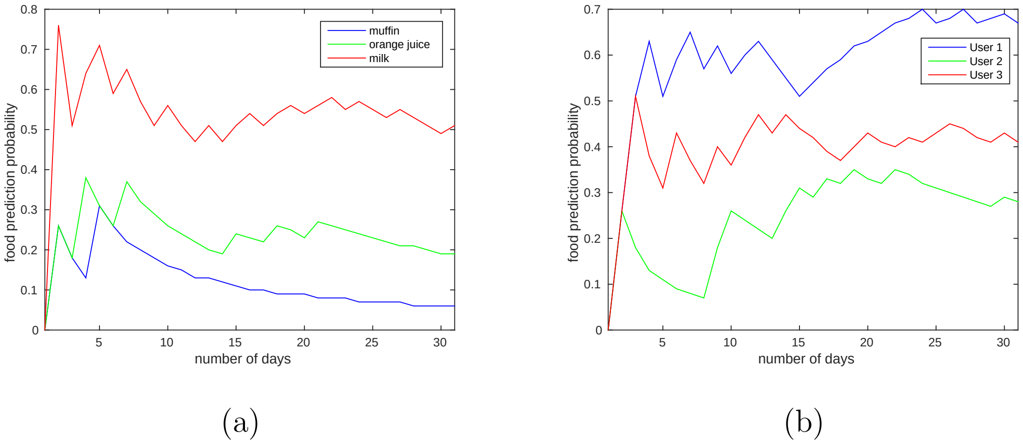


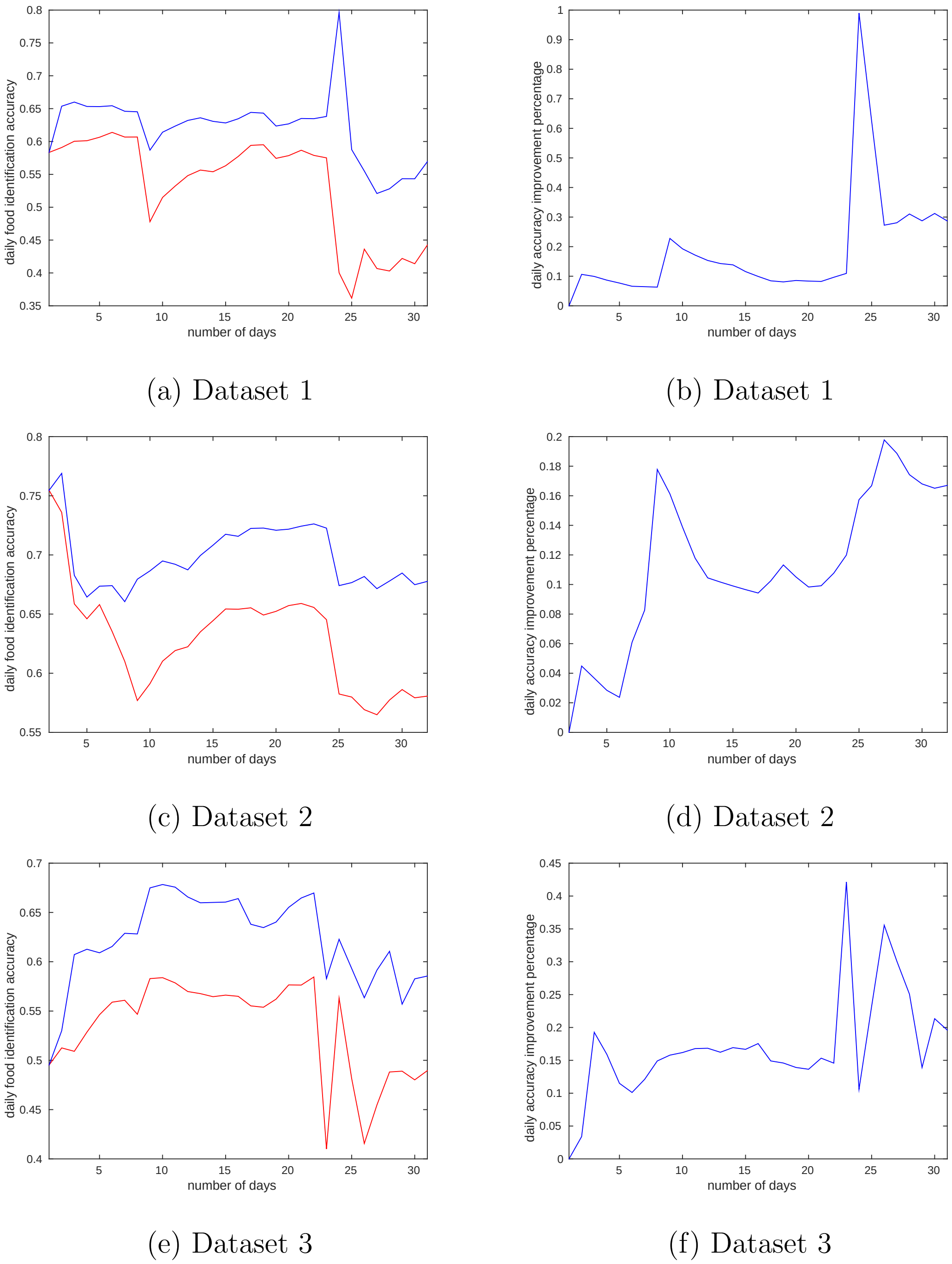
Fig. 7.13.: (a) Food occurrence prediction of three food items, (b) Prediction of milk among three datasets

Note we use the food label with the highest confidence score (top 1) from the classifier. As we show below the classification accuracy from the highest confidence score is in the range of 50-65%. In the TADA system we report the top 4 food labels and have a classification accuracy of 80-85% [11].

Figure 7.14 demonstrates the food classification accuracy improvement. The blue lines in Figure 7.14(a), 7.14(c) and 7.14(e) indicate the average daily food classification accuracy with temporal context, Θ_{context}, while the red lines indicate the one without, Θ_{auto}. The accuracy improvement is illustrated in Figure 7.14(b) and it is defined as, $improvement = (\Theta_{\text{context}} - \Theta_{\text{auto}})/\Theta_{\text{auto}}$.

As shown in Figure 7.14(b), the accuracy improvement drops from Day 10 to Day 20 as the baseline (classification accuracy without context) increases from 47% to 57%. This implies that the proposed method is more effective when the automatic image analysis does not work well. The 80% accuracy rate achieved with temporal context on Day 25 in Figure 7.14(a) demonstrates the effectiveness of the proposed method when the automatic image analysis result is poor (36%). In *Dataset 2*, the classification accuracy without context is always above 55% (see the red line in Figure 7.14(c)). The drop in the first few days shown in Figure 7.14(d) implies the undergoing learning process. Nevertheless, Figure 7.14(b) and Figure 7.14(d) both illustrate an ascent trend of accuracy improvement.

We selected images of the last 7 days to have a noticeably different food consumption pattern compared to the first 23 days in *Dataset 3*. We would like to verify the behavior of our training model under the circumstance where a participant may change their eating style. We witnessed a huge drop in Figure 7.14(f) followed by the re-learning state. The accuracy improvement is the minimum on Day 24 after the first week, because the context-based prediction puts more confidence in the specific food, which *Dataset 3* no longer contains after the user 3 changes eating habit. For example, milk is not consumed on Day 24. Due to the dietary change in *Dataset 3*, the increasing trend of classification accuracy is not as obvious. Table 7.3 compares



(a) Dataset 1

(b) Dataset 1

(c) Dataset 2

(d) Dataset 2

(e) Dataset 3

(f) Dataset 3

Fig. 7.14.: Learning curves for one month. Daily classification rates with(blue) and without(red) temporal context are illustrated in (a),(c) and (e). Corresponding accuracy improvements are shown in (b),(d) and (f).

the average daily classification accuracy with and without contextual information for each user. The average daily accuracy improvement is calculated as:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\Theta_{context}^{(i)}-\Theta_{auto}^{(i)}}{\Theta_{auto}^{(i)}} \tag{7.9}$$

Due to our dataset selection, the classification accuracy using automatic image analysis alone in *Dataset 2* is significantly higher than other datasets. The lower accuracy in *Dataset 1* and *Dataset 3* reflects the variation in the subset of the total 1453 testing images. Thus, the accuracy improvement for *Dataset 2* is expected to be lower (10.94%). The fact that *Dataset 2* has less frequently consumed foods also contributed to the lower accuracy improvement. When a person has a more consistent eating pattern, such as User 1, the classification accuracy gain using temporal contextual information is higher (18.69%). On average, the proposed method of utilizing temporal context shows 15.56% improvement. In the end, three datasets obtain roughly 65% accuracy with contextual information, which slightly lower than 71.4% we reported in the first experiment. This is because the classifier gradually learns the personalized eating patterns throughout 30 days, therefore the accuracy improvement in the earlier days is expected to be relatively lower than the one of the later days.

Table 7.3.: Food classification with contextual information

statistics	user ID	with context	without context
average daily classification accuracy(%)	user1	62.90	53.23
	user2	69.81	62.90
	user3	62.12	53.28
average daily accuracy improvement(%)	user1	18.69	
	user2	10.94	
	user3	17.05	

8. SUMMARY AND FUTURE WORK

8.1 Conclusions

In this thesis, we described the TADA system with the focus on some new components including the Android mobile application, the new implementation of the TADA web inference and the CTADA crowdsourcing tool. We introduced a color correction method with saliency based deblurring. We described a graph based image segmentation method using superpixels. We investigated the use of deep features and deep networks to improve food recognition. We also proposed an end-to-end food analysis pipeline. To deal with the sparsity of the segmentation groundtruth, we proposed a weakly supervised segmentation method using class activation maps. The weakly supervised technique only requires the label of the input image. Finally, we integrated contextual information into the TADA system and introduced the personalized learning model to further improve the food recognition accuracy. The result indicates that our contextual models are promising and further investigation is warranted.

The main contributions of this work are as follows:

- We created the TADA Android application to assist our user studies. Several user-friendly features were implemented, such as automatic update, background image uploading and crash report. A legacy version with newer Android API was developed to improve the user experience and prepare for the Android system update in the future.
- We improved the TADA web interface with a modular and secure design. The current implementation is much easier for maintenance and cloning to different servers.

- We implemented a crowdsourcing tool in an effort to build a large-scale food image dataset with richer groundtruth information.

- We proposed a polynomial color correction model in the LMS color space. A fiducial marker was previously designed as a color and distance reference. We computed a color correction matrix based on the 11 detected colors in the unknown environment and the same colors measured in the D65 lighting condition.

- We again made use of the fiducial marker present in the scene and introduced a de-blurring method based on a saliency map. The deconvolution kernel was estimated using the area that contains the fiducial marker.

- We proposed a graph based segmentation method which combines the normalized cut with a superpixel technique. We used superpixels to extract higher level features and reduce the number of nodes and connections in the constructed graph. There is a trade-off between using more complex features and processing speed. In the current implementation, we used color and texture features. Our method achieved competitive results on both food and non-food datasets.

- We described a weakly supervised segmentation technique which only requires image level annotation. We introduced a new type of pooling layer and coupled it with a modified VGG-16 network to improve class activation maps. We evaluated the proposed network for both classification and segmentation tasks and achieved competitive performance.

- We proposed a 3-stage food analysis pipeline powered by deep networks. We designed a food-vs-nonfood localizer based Faster RCNN and we adopted the deformable convolution to improve the localization accuracy. We evaluate several modern classification networks and achieve the top1 and 5 single-model accuracy of 83.27%/96.35%.

- We investigated adversarial examples in the domain of food images.
- We described a few scenarios where any traditional classifiers struggles. We proposed to use contextual information as the postprocessing step to further refine the food recognition results. We also integrated food co-occurrence pattern and temporal context into a personalized learning model.

8.2 Future Work

Potential topics for the future work include:

- Although our mFR system has been tested by many users studies, improvement is definitely welcome at both the frontend and backend. We should consider some UI modifications on both the iOS and Android applications as the design code at the system level has been changing rapidly. It will be great if we can collect more feedback from the review process on the mobile phone without putting too much burden on the user. On the backend server, we implemented a queuing mechanism for the automatic image analysis. However, as more users start to use our TADA system, stacking the incoming images in a single queue is no longer an option. We should investigate the scalability of the food image analysis by introducing queuing management and allocating tasks to multiple cloud resources.
- In the current context-based model, we predict a user's diet mainly based on the food consumption frequency in the breakfast, lunch and dinner. And we consider the user's eating pattern from the first day when he/she is registered in our system. However, a more sophisticated model is possible such that it has both short-term and long-term memory of the user's dietary history, as sometimes a person may change his/her daily food consumption based on his/her location or even emotion. With more user feedback and groundtruth data, we

should expand the food co-occurrence model and the personalized model to accommodate more food categories.

- To compute the edge weights in SNcut, we combined the confidence scores from different color and texture features by thresholding in a certain interval. Similarly in the context-based learning model, we used a weighted sum of confidence scores from the automatic predictions and the inference from our context models. In the future, we would like to incorporate a probabilistic model that adapts to the optimal weights.

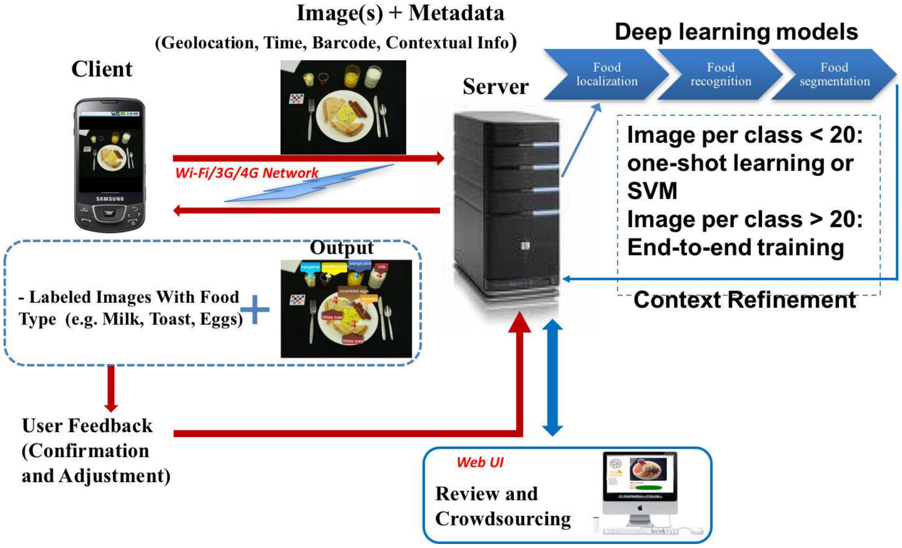

Image(s) + Metadata
(Geolocation, Time, Barcode, Contextual Info)
Client
Server
Deep learning models
Food localization
Food recognition
Food segmentation
Image per class < 20: one-shot learning or SVM
Image per class > 20: End-to-end training
Context Refinement
Wi-Fi/3G/4G Network
Output
- Labeled Images With Food Type (e.g. Milk, Toast, Eggs)
+
User Feedback (Confirmation and Adjustment)
Web UI
Review and Crowdsourcing

Fig. 8.1.: The potential system architecture powered by deep learning.

- To select effective contextual data is also an interesting problem. Apart from the context information we are currently using, i.e. date, time, and geolocation, there are more to explore, for example contextual location that describes not only the GPS coordinates of the location but location type, i.e. a restaurant versus the users' home or more subtlely whom the user is eating with. For example, if the user is eating at home, then our personalized eating model can be deployed with more confidence. All the information help us to constrain the classifier decisions.

- To take advantage of big data and deep learning, we should first have a large-scale food image dataset with sufficient groundtruth. Thus, we will continue our work on crowdsourcing.

- Incorporating user feedback into the image analysis system is a challenging problem, especially when the quality of the user feedback is unknown. If we assume the user feedback is always correct, which means the user provides accurate food labels and bounding boxes, how can we make the data trainable? Figure 8.1 shows a potential system to fully utilize the traditional machine learning, deep learning and the user feedback. When the data of a certain food class is insufficient, we can explore the possibility of using one-shot learning or simply training a one-vs-all SVM. When enough data is collected for a class, maybe it is feasible to retrain the end-to-end classification network.

REFERENCES

REFERENCES

[1] F. E. Thompson, A. F. Subar, C. M. Loria, J. L. Reedy, and T. Baranowski, “Need for technological innovation in dietary assessment,” *Journal of the American Dietetic Association*, vol. 110, no. 1, pp. 48–51, January 2010.

[2] A. Smith, “U.S. smartphone use in 2015,” PewInternet, Tech. Rep., April 2015. [Online]. Available: http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/

[3] “Tuingle.” [Online]. Available: http://tuingle.com/

[4] C. J. Boushey, D. A. Kerr, J. Wright, K. D. Lutes, D. S. Ebert, and E. J. Delp, “Use of technology in children’s dietary assessment,” *European journal of Clinical Nutrition*, vol. 63, pp. S50–S57, February 2009.

[5] B. L. Six, T. E. Schap, F. Zhu, A. Mariappan, M. Bosch, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey, “Evidence-based development of a mobile telephone food record,” *Journal of the American Dietetic Association*, vol. 110, no. 1, pp. 74–79, January 2010.

[6] C. Xu, N. Khanna, C. Boushey, and E. Delp, “Low complexity image quality measures for dietary assessment using mobile devices,” *Proceedings of the IEEE International Symposium on Multimedia*, pp. 351–356, December 2011, Dana Point, CA.

[7] C. Xu, F. Zhu, N. Khanna, C. Boushey, and E. Delp, “Image enhancement and quality measures for dietary assessment using mobile devices,” *Proceedings of the IS&T/SPIE Conference on Computational Imaging X*, vol. 8296, pp. 82 960Q–1–10, January 2012, San Francisco, CA.

[8] Y. Wang, C. Xu, C. Boushey, F. Zhu, and E. J. Delp, “Mobile image based color correction using deblurring,” *Proceedings of the IS&T/SPIE Conference on Computational Imaging*, vol. 9401, pp. 940 107–940 107–12, Feburary 2015, San Francisco, CA.

[9] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp, “The use of mobile devices in aiding dietary assessment and evaluation,” *IEEE journal of Selected Topics in Signal Processing*, vol. 4, no. 4, pp. 756 –766, August 2010.

[10] F. Zhu, M. Bosch, N. Khanna, C. Boushey, and E. Delp, “Multiple hypotheses image segmentation and classification with application to dietary assessment,” *IEEE journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 377–388, January 2015.

[11] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, “Analysis of food images: Features and classification,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 2744–2748, October 2014, Paris, France.

[12] "USDA food and nutrient database for dietary studies, 1.0." Beltsville, MD: Agricultural Research Service, Food Surveys Research Group, 2004.

[13] M. Bosch, T. Schap, F. Zhu, N. Khanna, C. Boushey, and E. Delp, "Integrated database system for mobile dietary assessment and analysis," *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1 – 6, July 2011, Barcelona, Spain.

[14] F. Kong, H. He, H. A. Raynor, and J. Tan, "Dietcam: Multi-view regular shape food recognition with a camera phone," *Pervasive and Mobile Computing*, vol. 19, pp. 108–121, 2015.

[15] T. F. Aflague, C. J. Boushey, R. T. L. Guerrero, Z. Ahmad, D. A. Kerr, and E. J. Delp, "Feasibility and use of the mobile food record for capturing eating occasions among children ages 3–10 years in guam," *Nutrients*, vol. 7, no. 6, pp. 4403–4415, June 2015.

[16] K. Kitamura, T. Yamasaki, and K. Aizawa, "Foodlog: Capture, analysis and retrieval of personal food images via web," *Proceedings of the ACM multimedia workshop on Multimedia for cooking and eating activities*, pp. 23 – 30, November 2009, Beijing, China.

[17] M. Sun, L. E. Burke, Z.-H. Mao, Y. Chen, H.-C. Chen, Y. Bai, Y. Li, C. Li, and W. Jia, "ebutton: a wearable computer for health monitoring and personal assistance," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference*, pp. 1–6, June 2014, San Francisco, CA.

[18] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," *Proceedings of the IEEE International Conference on Image Processing*, pp. 285–288, November 2009, Cairo, Egypt.

[19] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," *Proceedings of the IEEE International Conference on Image Processing*, pp. 289–292, November 2009, Cairo, Egypt.

[20] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2249–2256, June 2010, San Francisco, CA.

[21] Z. Zong, D. T. Nguyen, P. Ogunbona, and W. Li, "On the combination of local texture and global structure for food classification," *Proceedings of the IEEE International Symposium on Multimedia*, pp. 204–211, December 2010, Taichung, Taiwan.

[22] M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, "A food recognition system for diabetic patients based on an optimized bag-of-features model," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1261–1271, July 2014.

[23] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 25–30, July 2012, Melbourne, Australia.

[24] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008, Anchorage, AK.

[25] Y. Deng and B. Manjunath, “Unsupervised segmentation of color-texture regions in images and video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, August 2001.

[26] Y. Kawano and K. Yanai, “Foodcam-256: a large-scale real-time mobile food recognitionsystem employing high-dimensional features and compression of classifier weights,” *Proceedings of the ACM International Conference on Multimedia*, pp. 761–762, November 2014, Orlando, FL.

[27] L. Bossard, M. Guillaumin, and L. Van Gool, “Food-101 – mining discriminative components with random forests,” *European Conference on Computer Vision*, vol. 8694, pp. 446–461, September 2014, Zurich, Switzerland.

[28] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menu-match: Restaurant-specific food logging from images,” *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 844–851, January 2015, Waikoloa Beach, HI.

[29] V. Bettadapura, E. Thomaz, A. Parnami, G. Abowd, and I. Essa, “Leveraging context to support automated food recognition in restaurants,” *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 580–587, January 2015, Waikoloa Beach, HI.

[30] W. Ouyang and X. Wang, “Joint deep learning for pedestrian detection,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2056–2063, December 2013, Darling Harbour, Sydney.

[31] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, June 2014, Columbus, OH.

[32] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[33] S. Ao and C. X. Ling, “Adapting new categories for food recognition with deep representation,” *Proceedings of the IEEE International Conference on Data Mining Workshop*, pp. 1196–1203, November 2015, Atlantic City, NJ.

[34] K. Yanai and Y. Kawano, “Food image recognition using deep convolutional network with pre-training and fine-tuning,” *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, pp. 1–6, July 2015, Torino, Italy.

[35] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy, “Im2calories: Towards an automated mobile vision food diary,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1233–1241, December 2015, Santiago, Chile.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, June 2015, Boston, MA.

[37] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *arXiv preprint arXiv:1606.00915*, June 2016.

[38] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, “Recipe recognition with large multimodal food dataset,” *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, pp. 1–6, June 2015, Torino, Italy.

[39] Y. Kawano and K. Yanai, “Automatic expansion of a food image dataset leveraging existing categories with domain adaptation,” *Proceedings of European Conference on Computer Vision Workshop*, pp. 3–17, September 2014, Zurich, Switzerland.

[40] J. Zheng, Z. J. Wang, and C. Zhu, “Food image recognition via superpixel based low-level and mid-level distance coding for smart home applications,” *Sustainability*, vol. 9, no. 5, pp. 856:1–17, May 2017.

[41] J. Shang, M. Duong, E. Pepin, X. Zhang, K. Sandara-Rajan, A. Mamishev, and A. Kristal, “A mobile structured light system for food volume estimation,” *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 100–101, November 2011, Barcelona, Spain.

[42] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition and volume estimation of food intake using a mobile device,” *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 1–8, December 2009, Snowbird, UT.

[43] F. Kong and J. Tan, “Dietcam: regular shape food recognition with a camera phone,” *Proceedings of the IEEE International Conference on Body Sensor Networks*, pp. 127–132, May 2011, Dallas, TX.

[44] M. Sun, J. Fernstrom, W. Jia, S. Hackworth, N. Yao, Y. Li, C. Li, M. Fernstrom, and R. Sclabassi, “A wearable electronic system for objective dietary assessment,” *Journal of the American Dietetic Association*, vol. 110, no. 1, p. 45, January 2010.

[45] Y. Wang, Y. He, F. Zhu, C. Boushey, and E. Delp, “The use of temporal information in food image analysis,” *New Trends in Image Analysis and Processing*, ser. Lecture Notes in Computer Science, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Springer International, 2015, vol. 9281, pp. 317–325.

[46] S. Fang, C. Liu, F. Zhu, E. Delp, and C. Boushey, “Single-view food portion estimation based on geometric models,” *Proceedings of the IEEE International Symposium on Multimedia*, pp. 385–390, December 2015, Miami, FL.

[47] D. A. Kerr, C. Pollard, P. A. Howat, E. J. Delp, M. Pickering, K. R. Kerr, S. S. Dhaliwal, I. S. Pratt, J. Wright, and C. J. Boushey, “Connecting health and technology (chat): Protocol of a randomized controlled trial to improve nutrition behaviours using mobile devices and tailored text messaging in young adults,” *BMC Public Health*, vol. 12, no. 447, pp. 1–10, June 2012.

[48] D. A. Kerr, A. J. Harray, C. M. Pollard, S. S. Dhaliwal, E. J. Delp, P. A. Howat, M. R. Pickering, Z. Ahmad, X. Meng, I. S. Pratt, J. L. Wright, K. R. Kerr, and C. J. Boushey, “The connecting health and technology study: a 6-month randomized controlled trial to improve nutrition behaviours using a mobile food record and text messaging support in young adults,” *International Journal of Behavioral Nutrition and Physical Activity*, vol. 13, no. 1, pp. 52:1–14, April 2016.

[49] B. L. Daugherty, T. E. Schap, R. Ettienne-Gittens, F. Zhu, M. Bosch, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey, “Novel technologies for assessing dietary intake: Evaluating the usability of a mobile telephone food record among adults and adolescents,” *Journal of Medical Internet Research*, vol. 14, no. 2, pp. e58:1–12, April 2012.

[50] C. Boushey, D. Kerr, T. Schap, and B. Daugherty, “Importance of user interaction with automated dietary assessment methods,” *European Journal of Clinical Nutrition*, vol. 66, no. 5, p. 648, May 2012.

[51] T. Schap, B. Six, E. Delp, D. Ebert, D. Kerr, and C. Boushey, “Adolescents in the United States can identify familiar foods at the time of consumption and when prompted with an image 14 h postprandial, but poorly estimate portions,” *Public Health Nutrition*, vol. 14, no. 7, pp. 1184–1191, February 2011.

[52] C. J. Boushey, E. J. Delp, Z. Ahmad, Y. Wang, S. M. Roberts, and L. M. Grattan, “Dietary assessment of domoic acid exposure: what can be learned from traditional methods and new applications for a technology assisted device,” *Harmful Algae*, vol. 57, pp. 51–55, July 2016.

[53] J. Howe, *Crowdsourcing: How the power of the crowd is driving the future of business.* Random House, 2008.

[54] ——, “The wisdom of the crowd resides in how the crowd is used,” *Nieman Reports*, vol. 4, no. 62, pp. 47–50, 2008.

[55] A. Doan, R. Ramakrishnan, and A. Y. Halevy, “Crowdsourcing systems on the world-wide web,” *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, April 2011.

[56] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, *et al.*, “Predicting protein structures with a multiplayer online game,” *Nature*, vol. 466, no. 7307, pp. 756–760, June 2010.

[57] C. B. Eiben, J. B. Siegel, J. B. Bale, S. Cooper, F. Khatib, B. W. Shen, B. L. Stoddard, Z. Popovic, and D. Baker, “Increased diels-alderase activity through backbone remodeling guided by foldit players,” *Nature biotechnology*, vol. 30, no. 2, pp. 190–192, January 2012.

[58] G. M. Turner-McGrievy, E. E. Helander, K. Kaipainen, J. M. Perez-Macias, and I. Korhonen, "The use of crowdsourcing for dietary self-monitoring: crowdsourced ratings of food pictures are comparable to ratings by trained observers," *Journal of the American Medical Informatics Association*, vol. 22, no. e1, pp. amiajnl–2014, April 2015.

[59] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos, "Platemate: crowdsourcing nutritional analysis from food photographs," *Proceedings of the ACM symposium on User Interface Software and Technology*, pp. 1–12, October 2011, Santa Barbara, CA.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009, Miami, FL.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *Proceedings of European Conference on Computer Vision*, pp. 740–755, September 2014, Zurich, Switzerland.

[62] G. Sharma, *Digital Color Imaging Handbook*. Boca Raton, Florida: CRC Press, 2002.

[63] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, September 2010.

[64] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons, "Moment invariants for recognition under changing viewpoint and illumination," *Computer Vision and Image Understanding*, vol. 94, no. 1, pp. 3–27, April 2004.

[65] G. Hong, R. Luo, and P. Rhodes, "A study of digital camera colorimetric characterisation based on polynomial modelling," *Color Research and Application*, vol. 26, no. 1, pp. 76–84, February 2001.

[66] R. Bala, G. Sharma, V. Monga, and J.-P. Van de Capelle, "Two-dimensional transforms for device color correction and calibration," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1172–1186, August 2005.

[67] V. Cheung, S. Westland, and M. Thomson, "Accurate estimation of the nonlinearity of input/output response for color cameras," *Color Research & Application*, vol. 29, no. 6, pp. 406–412, December 2004.

[68] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms. II. experiments with image data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 985–996, September 2002.

[69] S. Bianco and R. Schettini, "Two new von kries based chromatic adaptation transforms found by numerical optimization," *Color Research & Application*, vol. 35, no. 3, pp. 184–192, Janurary 2010.

[70] B. Funt and H. Jiang, "Nondiagonal color correction," *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 481–484, September 2003, Barcelona, Spain.

[71] A. Gijsenij, R. Lu, and T. Gevers, “Color constancy for multiple light sources,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 697–707, February 2012.

[72] S. Bianco, F. Gasparini, and R. Schettini, “Consensus-based framework for illuminant chromaticity estimation,” *Journal of Electronic Imaging*, vol. 17, no. 2, pp. 023 013–023 013–9, May 2008.

[73] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, September 2001.

[74] S. Srivastava, C. Xu, and E. Delp, “White synthesis with user input for color balancing on mobile camera systems,” *Proceedings of the IS&T/SPIE Conference on Multimedia on Mobile Devices*, vol. 8304, pp. 83 040F–1–8, Jan 2012, San Francisco, CA.

[75] G. Buchsbaum, “A spatial processor model for object colour perception,” *Journal of the Franklin Institute*, vol. 310, no. 1, pp. 1–26, July 1980.

[76] G. D. Finlayson, S. D. Hordley, and P. M. Hubel, “Color by correlation: A simple, unifying framework for color constancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209–1221, November 2001.

[77] J. Von Kries, “Sources of color science,” *Chromatic adaptation*, pp. 109–119, 1970, Cambridge, MA.

[78] A. Rizzi, C. Gatta, and D. Marini, “A new algorithm for unsupervised global and local color correction,” *Pattern Recognition Letters*, vol. 24, no. 11, pp. 1663–1677, July 2003.

[79] A. Moreno, B. Fernando, B. Kani, S. Saha, and S. Karaoglu, “Color correction: a novel weighted Von Kries model based on memory colors,” *Computational Color Imaging*. Springer, April 2011, pp. 165–175, Milan, Italy.

[80] S. Bianco and R. Schettini, “Color constancy using faces,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 65–72, June 2012, Providence, Rhode Island.

[81] X. Wang and D. Zhang, “An optimized tongue image color correction scheme,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 6, pp. 1355–1364, September 2010.

[82] A. Ilie and G. Welch, “Ensuring color consistency across multiple cameras,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1268–1275, October 2005, Beijing, China.

[83] A. Munsell and M. Color, *Munsell soil color charts.* Munsell Color, 2000, New Windsor, NY.

[84] F. Zhu, M. Bosch, C. Boushey, and E. Delp, “An image analysis system for dietary assessment and evaluation,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 1853 –1856, September 2010, Hong Kong, China.

[85] B. Wandell, *Foundations of Vision.* Sinauer Associates, Inc., 1995, Sunderland, MA.

[86] M. D. Fairchild, *Color appearance models.* John Wiley & Sons, 2013.

[87] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, June 1963.

[88] D. Pascale, "RGB coordinates of the Macbeth color checker," *The BabelColor Company*, pp. 1–16, June 2006.

[89] P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications.* CRC press, 2007, Boca Raton, FL.

[90] G. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," *Optics Letters*, vol. 13, no. 7, pp. 547–549, July 1988.

[91] A. Katsaggelos and K.-T. Lay, "Maximum likelihood blur identification and image restoration using the EM algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 729–733, March 1991.

[92] C. Likas and N. Galatsanos, "A variational approach for bayesian blind image deconvolution," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2222–2233, August 2004.

[93] R. Lane and R. Bates, "Automatic multidimensional deconvolution," *Journal of the Optical Society of America A*, vol. 4, no. 1, pp. 180–188, January 1987.

[94] T. S. Cho, S. Paris, B. K. Horn, and W. T. Freeman, "Blur kernel estimation using the radon transform," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 241–248, June 2011, Colorado Spring, CO.

[95] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," *Proceedings of European Conference on Computer Vision*, pp. 157–170, September 2010, Crete, Greece.

[96] J. Miskin and D. MacKay, "Ensemble learning for blind image separation and deconvolution," *Advances in Independent Component Analysis.* Springer, 2000, pp. 123–141, London, UK.

[97] A. Levin, "Blind motion deblurring using image statistics," *Processings of Advances in Neural Information Processing Systems*, pp. 841–848, December 2006, Vancouver, Canada.

[98] M. Bronstein, A. Bronstein, M. Zibulevsky, and Y. Zeevi, "Blind deconvolution of images using optimal sparse representations," *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 726–736, June 2005.

[99] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 787–794, July 2006.

[100] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971, June 2009, Miami, FL.

[101] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 233–240, June 2011, Colorado Spring, CO.

[102] S. Cho and S. Lee, "Fast motion deblurring," *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 145:1–8, December 2009.

[103] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–10, August 2008.

[104] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, July 2012.

[105] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, July 1991.

[106] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," *Proceedings of Advances in Neural Information Processing Systems*, pp. 1033–1041, December 2009, Vancouver, Canada.

[107] S. Beucher, "Watershed, hierarchical segmentation and waterfall algorithm," *Mathematical Morphology and Its Applications to Image Processing.* Springer, 1994, pp. 69–76.

[108] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 315–327, February 2012.

[109] P. Arbelaez, "Boundary extraction in natural images using ultrametric contour maps," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pp. 182–182, June 2006, New York, NY.

[110] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011.

[111] M. Donoser and D. Schmalstieg, "Discrete-continuous gradient orientation estimation for faster image segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3158–3165, June 2014, Columbus, OH.

[112] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, September 2004.

[113] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.

[114] P. Arbel'aez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2294–2301, June 2009, Miami, FL.

[115] B. Catanzaro, B.-Y. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer, "Efficient, high-quality image contour detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2381–2388, June 2009, Miami, FL.

[116] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, "Food image analysis: Segmentation, identification and weight estimation," *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1–10, July 2013, San Jose, CA.

[117] M. Bosch, F. Zhu, N. Khanna, C. Boushey, and E. Delp, "Combining global and local features for food identification and dietary assessment," *Proceedings of the IEEE International Conference on Image Processing*, pp. 1789–1792, September 2011, Brussels, Belgium.

[118] F. Zhu, M. Bosch, N. Khanna, C. Boushey, and E. Delp, "Multilevel segmentation for food classification in dietary assessment," *Proceedings of the IEEE International Symposium on Image and Signal Processing and Analysis*, pp. 337–342, September 2011, Dubrovnik, Croatia.

[119] Y. He, N. Khanna, C. Boushey, and E. Delp, "Image segmentation for image-based dietary assessment: A comparative study," *Proceedings of the IEEE International Symposium on Signals, Circuits and Systems*, pp. 1–4, July 2013, Iasi, Romania.

[120] X. Ren and J. Malik, "Learning a classification model for segmentation," *Proceedings of IEEE International Conference on Computer Vision*, pp. 10–17, June 2003, Madison,WI.

[121] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, November 2012.

[122] D. R. K. Brownrigg, "The weighted median filter," *Communications of the ACM*, vol. 27, no. 8, pp. 807–818, August 1984.

[123] S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer, "Independence properties of directed markov fields," *Networks*, vol. 20, no. 5, pp. 491–505, August 1990.

[124] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.

[125] L. Wang and D.-C. He, "Texture classification using texture spectrum," *Pattern Recognition*, vol. 23, no. 8, pp. 905–910, June 1990.

[126] F. Yates, "Contingency tables involving small numbers and the χ 2 test," *Supplement to the Journal of the Royal Statistical Society*, vol. 1, no. 2, pp. 217–235, January 1934.

[127] D. R. Martin, "An empirical approach to grouping and segmentation," Ph.D. dissertation, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, August 2003.

[128] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," *Proceedings of the British Machine Vision Conference*, pp. 1–10, September 2007, Coventry, UK.

[129] U. Weidner, "Contribution to the assessment of segmentation quality for remote sensing applications," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. B7, pp. 479–484, July 2008.

[130] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, May 2004.

[131] F. J. Estrada and A. D. Jepson, "Benchmarking image segmentation algorithms," *International Journal of Computer Vision*, vol. 85, no. 2, pp. 167–181, November 2009.

[132] M. Meilă, "Comparing clusterings: an axiomatic view," *Proceedings of the International Conference on Machine Learning*, pp. 577–584, August 2005, Bonn, Germany.

[133] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, April 1960.

[134] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, June 2007.

[135] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[136] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 416–423, July 2001, Vancouver, Canada.

[137] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of Advances in Neural Information Processing Systems*, pp. 1097–1105, December 2012, Lake Tahoe, NV.

[138] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, June 2015, Boston, MA.

[139] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proceedings of Advances in Neural Information Processing Systems*, pp. 91–99, December 2015, Montreal, Canada.

[140] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, December 2015, Santiago, Chile.

[141] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1796–1804, December 2015, Santiago, Chile.

[142] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," *Proceedings of European Conference on Computer Vision*, pp. 1–15, May 2006, Graz, Austria.

[143] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, August 2013.

[144] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," *Proceedings of European Conference on Computer Vision*, pp. 297–312, September 2014, Zurich, Switzerland.

[145] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 685–694, June 2015, Santiago, Chile.

[146] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, Las Vegas, NV.

[147] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," *Proceedings of European Conference on Computer Vision*, pp. 695–711, October 2016, Amsterdam, Netherlands.

[148] N. Pourian, S. Karthikeyan, and B. Manjunath, "Weakly supervised graph based semantic segmentation by learning communities of image-parts," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1359–1367, December 2015, Boston, MA.

[149] S. Maji, N. K. Vishnoi, and J. Malik, "Biased normalized cuts," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2057–2064, June 2011, Colorado Spring, CO.

[150] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[151] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Proceedings of European Conference on Computer Vision*, pp. 346–361, September 2014, Zurich, Switzerland.

[152] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, “Automatic salient object segmentation based on context and shape prior,” *Proceedings of British Machine Vision Conference*, vol. 6, no. 7, pp. 9:1–12, September 2011, Dundee, Scotland.

[153] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, “Global contrast based salient region detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, March 2015.

[154] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, August 2004.

[155] J. C. Climaco and C. H. Antunes, “Implementation of a user-friendly software package - a guided tour of TRIMAP,” *Mathematical and Computer Modelling*, vol. 12, no. 10-11, pp. 1299–1309, January 1989.

[156] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” California Institute of Technology, April 2007.

[157] “TensorFlow: Large-scale machine learning on heterogeneous systems,” software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[158] T. Schap, F. Zhu, E. Delp, and C. Boushey, “Merging dietary assessment with the adolescent lifestyle,” *Journal of Human Nutrition and Dietetics*, vol. 27, no. s1, pp. 82–88, January 2014.

[159] D. M. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” *Journal of Machine Learning Technologyies*, vol. 2, no. 1, pp. 37–63, Feburary 2011.

[160] Y. Wang, C. Liu, F. Zhu, C. J. Boushey, and E. J. Delp, “Efficient superpixel based segmentation for food image analysis,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 2544–2548, September 2016, Pheonix, AZ.

[161] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, December 2015.

[162] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[163] “Inception in TensorFlow.” [Online]. Available: https://github.com/tensorflow/models/tree/master/inception

[164] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, December 2013.

[165] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, June 2016, Las Vegas, NV.

[166] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016, Las Vegas, NV.

[167] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," *to appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, Honolulu, HI.

[168] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, June 2014, Columbus, OH.

[169] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, January 2015.

[170] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, September 2013.

[171] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Proceedings of European Conference on Computer Vision*, pp. 21–37, October 2016, Amsterdam, Netherland.

[172] K. H. Jifeng Dai, Yi Li and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, May 2016.

[173] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, June 2016, Las Vegas, NV.

[174] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv preprint arXiv:1703.06870*, May 2017.

[175] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, December 2016.

[176] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, Feburary 2015.

[177] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *arXiv preprint arXiv:1703.06211*, March 2017.

[178] M. Bolanos and P. Radeva, "Simultaneous food localization and recognition," *arXiv preprint arXiv:1604.07953*, April 2016.

[179] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, December 2014.

[180] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *Proceedings of the IEEE European Symposium on Security and Privacy*, pp. 372–387, March 2016, Saarbrücken, Germany.

[181] M. Bar, "Visual objects in context," *Nature Reviews Neuroscience*, vol. 5, no. 8, pp. 617–629, August 2004.

[182] B. McFee, C. Galleguillos, and G. Lanckriet, "Contextual object localization with multiple kernel nearest-neighbor," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 570–585, February 2011.

[183] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, October 2007, Rio de Janeiro, Brazil.

[184] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, "Context based food image analysis," *Proceedings of the IEEE International Conference on Image Processing*, pp. 2748–2752, September 2013, Melbourne, Australia.

[185] M. Fink and P. Perona, "Mutual boosting for contextual inference," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1515–1522, December 2003, Vancouver, Canada.

[186] A. Torralba, K. P. Murphy, and W. T. Freeman, "Using the forest to see the trees: exploiting context for visual object detection and localization," *Communications of the ACM*, vol. 53, no. 3, pp. 107 – 114, March 2010.

[187] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 2017–2020, June 2012, Providence, Rhode Island.

[188] P. Felzenszwalb and D. Huttenlocher, "Image segmentation using local variation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 98–104, June 1998, Santa Barbara, CA.

[189] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, June 2014, Columbus, OH.

[190] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703 –715, June 2001.

[191] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin, "An efficient color representation for image retrieval," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 140–147, January 2001.

[192] C. Kenney, Y. Deng, B. S. Manjunath, and G. Hewer, "Peer group image enhancement," *IEEE Transactions on Image Processing*, vol. 10, no. 2, p. 326334, February 2001.

[193] M. Amadasun and R. King, "Textural features corresponding to textural properties," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 1264 – 1274, September 1989.

[194] B. Julesz, “Textons, the elements of texture perception and their iteractions,” *Nature*, vol. 290, no. 5802, pp. 91 – 97, March 1981.

[195] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, January 2004.

[196] K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 1, no. 60, pp. 63–86, January 2004.

[197] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (SURF),” *Journal of Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346– 359, June 2008.

[198] E. Tola, V. Lepetit, and P. Fua, “DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, May 2010.

[199] R. Duda and P. Hart, *Pattern classification and scene analysis.* John Wiley & Sons, February 1973.

[200] N. Martinel, G. L. Foresti, and C. Micheloni, “Wide-slice residual networks for food recognition,” *arXiv preprint arXiv:1612.06543*, December 2016.

[201] “Pytorch,” tensors and Dynamic neural networks in Python with strong GPU acceleration. [Online]. Available: http://www.pytorch.org/

[202] I. Biederman, R. Mezzanotte, and J. Rabinowitz, “Scene perception: detecting and judging objects undergoing relational violations,” *Cognitve Psychology*, vol. 14, no. 2, pp. 143–177, April 1982.

[203] K. Murphy, A. Torralba, and W. Freeman, “Using the forest to see the trees: a graphical model relating features, objects and scenes,” *Proceedings of Advances in Neural Information Processing Systems*, vol. 16, pp. 1499–1506, December 2003, Vancouver, Canada.

[204] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, “Context-based vision system for place and object recognition,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 273–280, October 2003, Nice, France.

[205] A. Oliva and A. Torralba, “The role of context in object recognition,” *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, December 2007.

[206] C. Galleguillos and S. Belongie, “Context based object categorization: A critical survey,” *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 712–722, February 2010.

[207] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine Learning*, vol. 37, no. 2, pp. 183–233, November 1999.

[208] S. Sarkka, *Bayesian Filtering and Smoothing.* Cambridge University Press, 2013, Cambridge, UK.

[209] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[210] W. Y. Ma, Y. Deng, and B. Manjunath, "Tools for texture- and color-based search of images," *Proceedings of the SPIE Human Vision and Electronic Imaging II*, vol. 3016, pp. 496–507, June 1997, San Jose, CA.

VITA

VITA

Yu Wang was born in Jinan, China. He received the B.E. in Electrical Engineering and Automation from Nanjing University of Aeronautics and Astronautics (NUAA), China.

Mr. Wang joined the Ph.D. program at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana in August 2013. He worked at the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Edward J. Delp.

Mr. Wang is a recipient of multiple Excellent Student Scholarship awarded by NUAA. He participated in BoilerMake Hackathon in 2015 and his team won the top award among over 600 selected students.

His research interests are image processing, computer vision and deep learning. He is a student member of the IEEE, the IEEE Signal Processing Society, SPIE, IS&T.

Yu Wang's publications are:

Journal Paper

1. **Yu Wang**, Ye He, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. "Context based image analysis with applications in dietary assessment and evaluation" *Multimedia Tools and Applications*, under review, Nov 2016.
2. Carol J. Boushey, Edward J. Delp, Ziad Ahmad, **Yu Wang**, Sparkle M. Roberts, and Lynn M. Grattan. "Dietary assessment of domoic acid exposure: What can be learned from traditional methods and new applications for a technology assisted device." *Harmful Algae*, vol.57 pp.51-55, 2016.

Conference Paper

1. **Yu Wang**, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. "Weakly supervised food image segmentation using class activation maps", *Proceedings of the IEEE International Conference on Image Processing*, to appear, September, 2017
2. **Yu Wang**, Shaobo Fang, Chang Liu, Fengqing Zhu, Deborah A Kerr, Carol J Boushey, and Edward J Delp. "Food image analysis: the big data problem you can eat!" *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, November 2016.
3. **Yu Wang**, Chang Liu, F. Zhu, C. J. Boushey, and E. J. Delp, "Efficient superpixel based segmentation for food image analysis, *Proceedings of the IEEE International Conference on Image Processing*, September 2016.
4. **Yu Wang**, Ye He, Fengqing Zhu, Carol J. Boushey, and Edward J. Delp. "The use of temporal information in food image analysis," *New Trends in Image Analysis and Processing - ICIAP 2015 Workshops, Lecture Notes in Computer Science*, Vol. 9281, Springer International, pp. 317-325, September 2015.

5. **Yu Wang**, Chang Xu, Carol J. Boushey, Fengqing Zhu, and Edward J. Delp. "Mobile image based color correction using deblurring," *Proceedings of the IS&T/SPIE Conference on Computational Imaging*, pp. 940107-940107, February 2014.

6. **Yu Wang**, Javier Ribera, Sri Yarlagad, Chang Liu, Fengqing Zhu. "Pill recognition using minimal labeled data," *Proceedings of the IEEE International Conference on Multimedia Big Data*, April, 2017.

7. David Gera, **Yu Wang**, Luca Bondi, Paolo Bestagini, Stefano Tubaro and Edward Delp. "A Counter-Forensic Method for CNN-Based Camera Model Identification", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, to appear, July, 2017