Pervasive Quantified-Self using Multiple Sensors

by

Junghyo Lee

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved May 2019 by the
Graduate Supervisory Committee:

Sandeep K.S. Gupta, Chair
Ayan Banerjee
Baoxin Li
Erin Chiou
Yogish C. Kudva

ARIZONA STATE UNIVERSITY

August 2019

ABSTRACT

The advent of commercial inexpensive sensors and the advances in information and communication technology (ICT) have brought forth the era of pervasive Quantified-Self. Automatic diet monitoring is one of the most important aspects for Quantified-Self because it is vital for ensuring the well-being of patients suffering from chronic diseases as well as for providing a low cost means for maintaining the health for everyone else. Automatic dietary monitoring consists of: a) Determining the type and amount of food intake, and b) Monitoring eating behavior, i.e., time, frequency, and speed of eating. Although there are some existing techniques towards these ends, they suffer from issues of low accuracy and low adherence. To overcome these issues, multiple sensors were utilized because the availability of affordable sensors that can capture the different aspect information has the potential for increasing the available knowledge for Quantified-Self. For a), I envision an intelligent dietary monitoring system that automatically identifies food items by using the knowledge obtained from visible spectrum camera and infrared spectrum camera. This system is able to outperform the state-of-the-art systems for cooked food recognition by 25% while also minimizing user intervention. For b), I propose a novel methodology, IDEA that performs accurate eating action identification within eating episodes with an average F1-score of 0.92. This is an improvement of 0.11 for precision and 0.15 for recall for the worst-case users as compared to the state-of-the-art. IDEA uses only a single wrist-band which includes four sensors and provides feedback on eating speed every 2 minutes without obtaining any manual input from the user.

ACKNOWLEDGMENTS

First and foremost, I would like to express deepest gratitude to my adviosr, Dr. Sandeep K.S. Gupta. His guidance, pragmatism, patience, encouragement, and strategic insights were key motivations to my whole Ph.D. studies. I appreciate all his contributions of time, idea, funding, and discussion to achieve the my Ph.D.

I am greatly thankful to Dr. Ayan Banerjee who has been my mentor for technical discussion and big help to reach this dissertation. Also, I want to express my appreciation to my Ph.D. committee members (Dr. Baoxin Li, Dr. Erin Chiou, and Dr. Yogish C. Kudva). Moreover, I thank supports of all my previous and current lab mates, Priyanka, Madhurima, Koosha, Prajwal, Javad, Imane, Vinaya, Subhasish, Apurupa, Azamat, and Bernard.

Finally, I am grateful to my wife, Hyejeong Kim, for supporting me faithfully throughout my Ph.D. study. And I would thank my mother, Dalmak Kim, for her constant love and devotion in my life.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Machine learning, internet-of-things (IoT), and artificial intelligence (A.I.) promise a new era for smart healthcare. Also, the advances in smart devices and the emergence of inexpensive sensors have enabled to collect and analyze the health conditions from anyone, anywhere, and anytime. These developments provide a pervasive healthcare framework to allow users to manage and assess their own health conditions. This is often called Quantified-Self or Life-logging. This trend in Quantified-Self is an important research topic because it allows the prevention of illnesses by monitoring individual health conditions such as diet, blood pressure, and activity in daily life. Quantified-Self is specially important for patients of chronic diseases and elderly people because sustained monitoring is critical to preserve their health. The analysis of the data obtained from continuous monitoring can be utilized for automatic and intelligent decisions such as a balanced diet recommendation using calorie estimation and fast eating speed warning using eating action recognition. These decisions can enhance the quality of healthcare service, which in-turn can lead to many beneficial health impacts such as low cost illness prevention.

Knowledge discovery is important since quality and quantity of discovered knowledge critically affect the performance of such decisions. Recent advances in feature engineering techniques for sensor data have greatly contributed to the increase in the quality of knowledge. However, the quantity of knowledge extracted also depends on the amount of information present in the sensor data source. An intuitive way to increase the quantity of knowledge would be to utilize multiple sensors. Theoretically, utilizing sensors that capture information from multiple sensors can lead to increase in the quantity of discovered knowledge and also increase the overall quality of knowledge leading to a synergy effect. However, there are many application-specific challenges that need to be overcome. To explore these challenges, advanced dietary monitoring techniques are required. It is an important aspect of treatment plans for many common health problems [1, 2]. Diet monitoring has several components including, dietary preference logging, maintenance of a diet plan, food allergy information logging, and eating activity monitoring. On a broad level, for diet monitoring system to have high usability, they should be able to accurately and automatically i) identify food items and ii) monitor eating behavior.

For i) the food items identification, utilizing a color and thermal camera sensor was presented to increase the quantity of knowledge because approaches based on only color cameras are not accurate. However, in order to utilize sensors from different spectra they must be properly caliberated. Multiple sources of visual information implies multiple viewpoints since different camera pin-holes cannot be co-located. Before the various sources can be utilized as inputs to an intelligent system, a pixel-by-pixel correspondence between them is required as a preprocessing step. This preprocessing step is called image registration which is typically achieved by utilizing a perspective projective transformation called Homography. This technique requires at least four matching feature points between two images and has only been shown to work for images in the visible spectrum. Wang et al. [3] have proposed a image registration method for satellite images using ASIFT [4]. ASIFT utilizes convolution of neighborhood texture features to obtain a good matching between corresponding pixels in color images from two fixed cameras. Then, a random selection of at least four pixels from one of the images, using algorithms such as RANSAC [5] allows successful image registration. However, this technique does not work properly for images in the non-visible spectrum because it may not be feasible to match at least four feature points. Even if four candidate feature points are identified, it is usually not possible to complete the registration because of the significant differences in textures between images from different spectra.

For ii) the eating behavior monitoring, a wristband which includes four different sensors was utilized. In order to increase the quantity of usuable knowledge by using these multiple sensors, proper segmentation must be achieved which can be challenging. A valid segment for the purpose of this system is required to have the following properties: P1) A segment should have continuous data from the start to the end of only one eating action. P2) An eating action should be contained in only one segment (i.e. no overlapping segments). To achieve such properties in the segments, we need to specify exact time duration of eating actions in the sliding-window-based algorithm. This is not possible in a User-Independent setting because the average duration of eating action varies significantly from person to person. Hence, an alternative segmentation method based on extrema points was performed. To obtain accurate segmentation using extrema points, the most appropriate sensor(s) must be selected.

In this dissertation, a computer vision based system was presented which fuses color and thermal images of cooked food to obtain an accuracy of nearly 90% for a variety of cooked food as discussed in Chapter 3. Several works have developed self-monitoring techniques for diet monitoring such as manual paper-based records (food diaries) and 24-hour dietary recalls. The purpose of these techniques has been to assess the amount and type of food eaten. However, these techniques suffer from three important drawbacks: a) adherence to self-monitoring for the prescribed period of intervention is low (nearly 60%) [6], b) self-reporting is prone to underreporting by amounts ranging from 20%-50%, especially in individuals with obesity [6, 7, 8, 9], and c) recall error while reporting food intake [10]. These drawbacks have prevented a wide-scale adoption of these automatic dietary monitoring which would otherwise benefit the users. Some state-of-the-art smartphone camera based monitoring systems [11, 12, 13, 14, 15, 16, 17] have been suggested to automate dietary monitoring in order to solve the problems associated with self-reporting. In addition to making the systems more usable, automatic food identification techniques based on computer vision also mitigate underreporting and recall error to some extent. However, the reported accuracy of such techniques are only about 70% for identification of cooked food which forms a significant portion of every day diet for most people. One main reason for the low performance of these systems for cooked food is their inability to differentiate between food-items in a plate that are not well separated.

Also, we present a wristband based system to accurately estimate the eating speed of users in real-time and in daily usage scenarios without restrictions on type of food. This is discussed in Chapter 4. Successful diet monitoring systems also benefit from automatic monitoring of eating behavior. This monitoring can provide several important insights into an eating episode like picking times, eating times, types of utensils used, the portion sizes for type of food consumed, and finally speed of eating. Information like this can be utilized in various ways to analyze the eating behavior and also to provide automatics feedback to the user. However, automatically and accurately monitoring the different components of an eating activity with minimal manual intervention is challenging. Recent survey [18] shows that lack of accurate estimate of food intake portions reduces the effectiveness of dietary monitoring systems for fostering a healthy eating behavior. Eating speed recognition is one of the most important analytics that can be derived from automatic eating activity monitoring. This is because, real-time feedback can be given to users if the eating speed can be accurately determined. Such feedback has been shown to positively impact the eating behavior of users [19].

Chapter 2

QUANTIFIED-SELF

The Quantified-Self (QS) is an approach for tracking and measuring individuals′ daily life activities and states. Information such as physical activity level, heartbeat, blood pressure, blood glucose, and diet can be collected using wearable sensors for this purpose. Many researchers have proposed the QS-based systems which allows users to manage and assess their own health conditions. Oh et al. [20] have categorized tracking for QS system into five themes: 1) Body information, 2) Psychological State and Traits, 3) Activity, 4) Social Interaction, and 5) Environmental & Property States. Among these five categories, the activity category is a representative of QS research. The tracking of physical activity level, eating, sleeping, taking pills, watching TV, and studying are examples of the category. The advances in smartphones and the emergence of inexpensive sensors have contributed to the pervasive data collection, which leads to the intelligent and pervasive diet monitoring.

Researchers have developed several self-monitoring techniques for diet, such as manual paper-based records (food diaries) and 24-hour dietary recalls that assess the amount and type of food eaten. However, these techniques suffer from three important drawbacks: a) adherence to self-monitoring for the prescribed period of intervention is low (nearly 60%) [6], b) self-reporting is prone to underreporting (especially in individuals with obesity for whom the amounts range from 20% to 50%) [6, 7], and c) recall error while reporting food intake [10]. Further, in self-reported dietary assessment, where a 0.5 to 0.7 correlation with actual intake would be considered good [21]; many studies have found a 0.4 correlation with self-reported dietary assessment and intake [9]. The misclassification of caloric intake and nutrient profiles

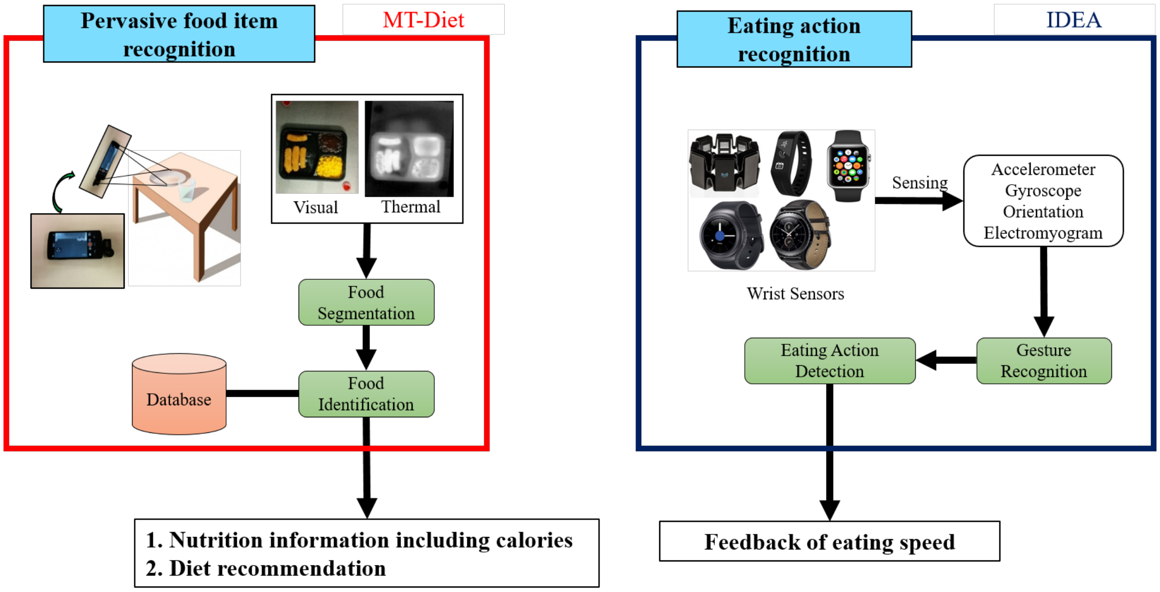


Figure 2.1: Proposed Fundamental Dietary Monitoring Systems.

tends to be different based on weight status and/or overall energy intake [8]. Moreover, after a weight loss program involving self-monitoring of diet, there is a high rate of relapse [22, 23]. Hence, we need objective, usable techniques to automatically assess dietary intake which makes this a very compelling use-case for intelligent QS systems.

Automatic diet monitoring can help in assessment of nutritional status, monitoring of compliance with dietary regimes, evaluation of outcome of dietary recommendations, and self-monitoring by motivated individuals. The close relationship between daily nutrient intakes and certain chronic diseases such as cardiovascular disease [24], diabetes [25], cancer [26], and obesity [27] necessitates accurate measurement of dietary intakes. These diet related chronic diseases cost the US economy nearly $1 trillion a year [28]. Dietary intervention in conjunction with physical activity and socio-economic environment changes is considered to be one of the significant factors in prevention of such chronic diseases, especially problems related to overweight and obesity [28]. Dietary interventions concentrate on two important aspects: a) the type and amount of food intake, and b) eating behavior, i.e., time, frequency, and speed of eating [28]. There have been significant efforts in designing interventions that consider various aspects of nutritional intake such as number and type of calories. However, such interventions often have financial impacts, especially on low-income populations [29, 30]. On the other hand, interventions that consider speed, time, duration and frequency of eating are behavioral in nature and are thus more affordable. In this dissertation, we present these two fundamental dietary monitoring systems as seen in Fig. 2.1: a) MT-Diet which utilizes a long wavelength infrared camera interfaced with a smartphone and b) IDEA that can operate with a commodity wristband sensor to instantly identify eating actions without any manual input from the user.

Chapter 3

FOOD ITEM RECOGNITION

Increased usage of smartphones with high resolution cameras and powerful processers provides the opportunity to employ intelligent systems for long-term, non-invasive, and automatic diet monitoring. Indeed, a recent study on university students found image based food recognition applications to be easier to use, more fun and were more likely to have continued usage when compared with a text based food log [31]. Another study of 50 university student reported that nearly 83% stated that they will not be too lazy to use an image based diet monitoring app for long terms and 50% intended to use them on a daily basis [32]. Practical deployment of image based diet monitors such as Portal [33] or MyFitnessPal [34] demonstrated ease of use but also highlighted concerns of low adherence, under-reporting and recall error. These studies also report that subjects were interested in getting additional information from the food monitoring app regarding type of food and calorie intake. In theory, intelligent diet monitoring systems should be able to provide these types of additional information, however in practice, there are multiple challenges that need to be addressed.

Cooked food tends to contribute to a significant proportion of calories in daily diets [35]. Multiple items of cooked food tends to be served in the same plate which makes it into a multi-object recognition challenge. The system has to not only identify the various types of food in an image, but it also has to identify the location for these food items (for eg. steak and mashed potatoes).

Most works on food identification consider food-items in isolation. The works that do consider multiple-items of food together have additional requirement for users to

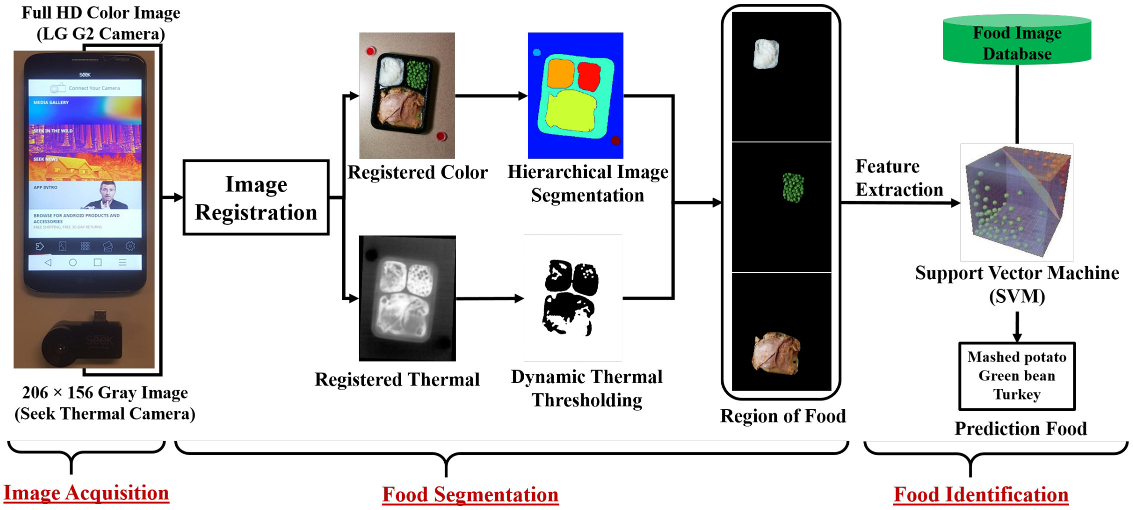


Figure 3.1: Overview Of *MT-DIET*.

draw a bounding box which makes the system less user friendly. Most works in this area report higher accuracy for fresh food (93%) as compared to cooked food (63%) [36, 37, 38, 39]. However, a significant portion of daily calorie intake consists of cooked food which makes their accurate recognition very important [35]. Additionally, cooked food has multiple items served in the same plate (for e.g. steak and mashed potatoes) which recognition using color images more challenging [40].

To solve this, in this dissertation, we present an intelligent and pervasive food image recognition system, MT-Diet [41] [1] which utilizes a long wavelength infrared camera interfaced with a smartphone. This system utilizes knowledge obtained from two different cameras: 1) in the visible spectrum and 2) in the infrared spectrum as shown in Fig. 3.1.

3.1 Overview

Before two spectra images can be utilized, an image registration between the thermal and visible spectra needs to be performed. Successful image registration requires

[1] Published in PerCom2016

high accuracy for feature matching between corresponding pixels of the two images. However, the features from thermal and color images are quite disparate, making it difficult to match corresponding features. To achieve high performance for feature matching, Jarc et. al [42] and Istenic et. al [43] proposed image registration methods that use texture and Hough transformation based features. These techniques assume the availability of high resolution infrared images, however, most commercial infrared cameras for smartphones typically have a very low resolution. In this dissertation, we presented an image registration approach that is successful even for low resolution images discussed in Sec. 3.5.1.

With cooked food, the food plate is much cooler than the food itself; as a result, the thermal image gives a better opportunity to accurately segment different food portions on a plate. Further, the same amount of heat will yield different temperature increases for different food items. Thus, even if two food types are mixed, a thermal image can distinguish between them. The segmented area is then applied to the color image after the image registration processing between color and thermal image. The main purpose of image registration is to find optimal spatial and intensity transformations such that the images get aligned into the same coordinate frame [44]. Unwanted portions of the food plate in segmentations are further removed using the GrabCut method [45]. MT-Diet then uses color histogram based analysis of each segment to determine the actual number of food items on the plate and the area covered by them. The food segments from both the thermal and color images are used to extract five features, a) relative temperature difference of each food item with respect to food plate (thermal), b) bag of features [46] (color), c) histogram of oriented gradients [47] (color), d) histogram of color map (color), and e) texture information (color) [48]. These features are then provided as input to a support vector machine (SVM) based classifier to identify the type of food.

If accurate and efficient, MT-Diet will be a significant improvement from the current state-of-the-art in smartphone based diet monitoring.

There are significant challenges in creating an automated camera based caloric intake measurement system:

(a) Inaccurate segmentation of food items: The average accuracy in automatically segmenting the food plate from the food using a color image is very poor (nearly 60%) [39]. Further, when different food items are mixed it is nearly impossible to distinguish them using just the visual image of the food plate. Use of thermal images along with visual images improves segmentation accuracy. Preliminary results from a database of 80 different types of frozen food warmed in a microwave for the recommended time show that use of thermal images in addition to visual images increases the segmentation accuracy to 92%.

(b) Inaccurate identification of food items: Existing food recognition has been successful in identifying raw food such as fruits or vegetables with accuracy nearly 100%. However, the state-of-the-art cooked food identification techniques from visual images have an accuracy of only 63% as shown in Table 3.1. MT-Diet augments machine learning classifiers with a temperature map of the food items. Preliminary data on identification of the food items from a frozen food plate warmed in a home-grade microwave show that inclusion of a thermal map of the food plate increases identification accuracy to 87%.

(c) Estimation of caloric intake: A nutrition expert charts the calorie value per gram of consumed food. Based on the amount and the type of food eaten, standard equations [49] can be used to compute caloric intake. The estimated cost of the MT-Diet prototype is around $140 for the Seek thermal camera (excluding the cost of smartphone). However, MT-Diet is not limited to the specific devices and it can operate with any smartphones and infrared camera.

Table 3.1: Camera-based Approaches That Use Images In The Visible Spectrum.

Group Parameter	UEC [11, 12, 13, 50, 14]	DCVER [36, 38, 37, 51]	He, Y. et al. [16]	Yang, S. et al. [17]	U of Bern [52]	GoCARB [53]	Bolanos et al. [54]	MT-Diet
Segmentation method	Graph-cut	Graph-cut	Local variation	Manual segmentation	Mean-shift with Thresholding	Pyramidal mean-shift	Fast R-CNN	Proposed method
Identification Accuracy	Avg: 65.4%	Avg: 99%	63%	Avg: 78%	Avg: 87%	88.5%	Avg: 75.13%	Avg: 88.93%
Multiple foods count Auto-level	Auto	Manual	Auto	Not Require	Manual	Auto	Auto	Auto
Segmentation Auto-level	None	Semi-Auto	Auto	Not Require	Semi-Auto	Auto	Auto	Auto
Food Type	100 types Japanese cooked foods	39 raw food 1 cooked vegetable	96 American food items	Fast food, fixed portions	Six major foods	24 foods	multiple dataset	33 types of cooked food
Plate Type	Various shape and color	White round dish	White round dish	Variable	White round dish	White round dish	Various shapes and colors	Various shapes and colors

3.2 Related Work

Several researchers have attempted to develop camera-based solutions for automated caloric intake estimation, but almost all of them have considered only images in the visible spectrum. Table 3.1 compares the recent works in this area in terms of several important parameters. In summary, the following characteristics are observed: i) segmentation algorithms that are semi-automatic, i.e., have some form of guidance from the user, have a better accuracy than fully automatic algorithms, ii) automatic or semi-automatic identification algorithms have excellent accuracy for identifying raw uncooked food but have poor accuracy (63%) for cooked food, and iii) the number of food items on a plate may not be computed automatically with high accuracy from an image in the visible spectrum and requires manual guidance. *MT-Diet* has the following advantages:

***MT-Diet* works regardless of the food plate shape or color:** There are two different types of attempts at food plate separation: a) assumption of a circular plate [36, 38, 37, 53] allows the usage of contour detection using Hough transform to identify the plate boundary. However, this approach can only remove background and does not separate food from the plate, and b) assumption that the plate is white, and food is non-white [13]. However, under these assumptions, distinction between certain food items such as rice from the plate can be difficult. Typically, the above two methods work well for raw food but they have very poor accuracy for cooked food (Table 3.1). Bolanos et al. [54] present a food segmentation approach based on Faster R-CNN [**?**], however they provide results only for coarse level bounding boxes. This kind of segmentation can be useful to identify the type of food, but obtaining information about the quantity of food can be problematic for which a pixel-based 'semantic' segmentation is desirable. Segmentation is an important step in food recognition and an inaccurate segmentation can lead to inaccuracies in food identification. In addition to the image in the visible spectrum, *MT-Diet* also captures the infrared image of the food plate. This results in more accurate separation of the food and the plate irrespective of the color or the shape, since the plate can be assumed to be much cooler than hot cooked food.

***MT-Diet* improves food identification by using thermal images:** Image identification is typically performed using machine learning algorithms, which are trained to recognize certain identifying features of a given food item. Commonly used features include statistical measures of color in different domains such as Red Green Blue (RGB) or Hue Saturation Value (HSV), texture, and edges. When a new food item is obtained the extracted features are used to classify it to the correct class of food items. Our preliminary data suggest that existing machine learning methods with features extracted in the visible spectrum have high accuracy for identifying raw food (around 97%), as also observed in previous research [36]. However, with cooked food, the accuracy reduces significantly mostly due to two reasons: a) cooked food may not have uniform color and texture, and b) different food types may by similar in color when cooked (e.g., brown rice and chicken). *MT-Diet* improves identification accuracy by increasing the feature dimension with color histogram information from the infrared images. Preliminary data shows that incorporation of infrared color histogram improves the identification accuracy to nearly 100% for different colored food and 74% for food with the same color.

***MT-Diet* maintains privacy of user data:** *MT-Diet* does not need crowdsourcing to train its machine learning approach for segmentation of the food plate or identification of food items. Hence it does not need to share personal food intake information with the community and supports user privacy.

3.3 Problem Statement

The problem statement is defined as follows:

Definition

Inputs:

a) A plate full of hot food,

b) color image from smartphone camera, and

c) thermal image from infrared camera.

Platform:

a) A smartphone interfaced with thermal camera, and

b) Reliable connection of the smartphone with a cloud server.

Assumptions:

a) Food temperature $\gg$ Plate temperature,

b) Plate temperature $>$ Background temperature,

c) The plate is not overflowing with food,

d) Offline food database is available, and

e) Users wait at least 5 min for food items to cool down before eating.

Outputs:

a) Food type in plate, and

b) Calories estimation into USDA website.

3.4 System Architecture

LG G2 smartphone and Seek thermal camera [55] are used to capture the food color images and thermal images. LG G2 is interfaced with the Seek thermal camera [55] through micro-USB as seen in Fig. 3.1. The thermal image is the gray scale image whose intensity is directly proportional to a temperature degree. Also, we implemented MT-Diet android application to demonstrate the pervasive food item recognition system as seen in Fig. 3.2. This application has two buttons for initiating the color image camera and the thermal camera. A user is asked to take two pictures of the food plate. These two images are transferred to the cloud server. This server starts three core tasks sequentially: a) image registration, b) food segmentation, and c) food identification. After competition in server-side, the MT-Diet application will provide users the food types by a spinner button. The users can click on the button for each food, which accesses to the USDA website that shows the nutritional information for the food.

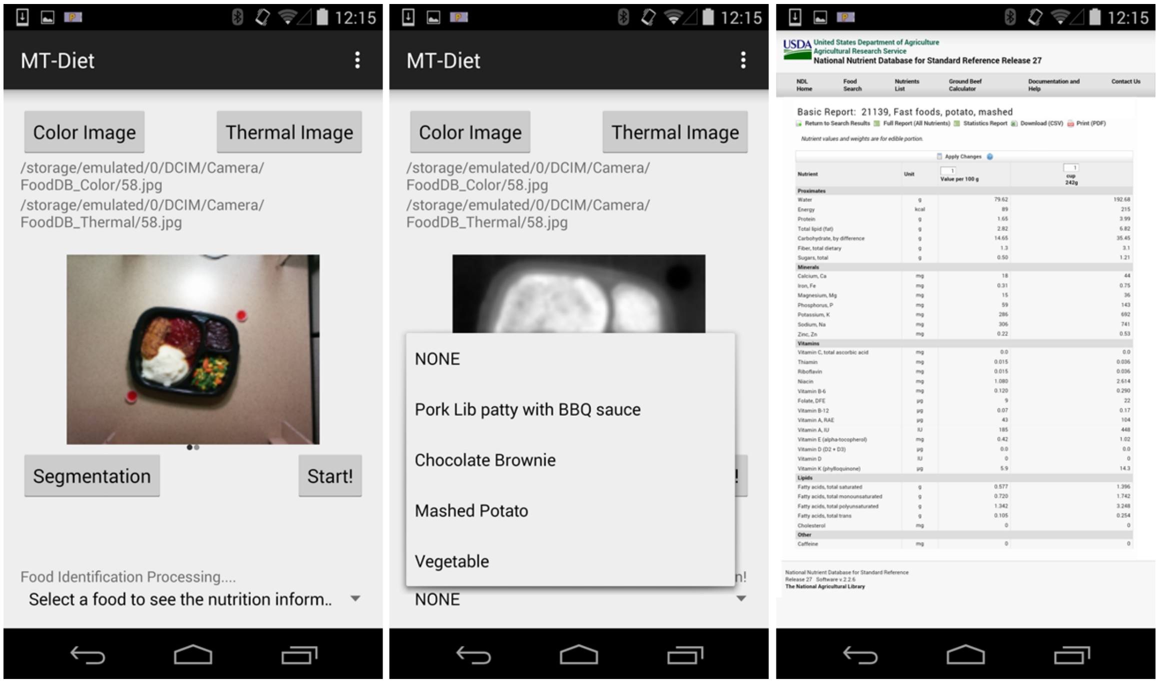
Data collection: we collected 80 frozen foods. The food were heated using a microwave as recommended by the manufacturer. The 80 foods consists of overall 33 different types as seen in Tab. 3.2.

3.5 Methodology

The system follows a pipeline which consists of three core components: i) image registration, ii) food segmentation, and iii) food type identification. In the section, we will describe each component$'$ method in detail.

Table 3.2: Food Database With 33 Different Food Types.

ID	Name
1	Cheese and Macaroni
2	Turkey and Mashed potato
3	Vegetable
4	Chocolate pudding
5	Fish stick
6	Chicken nugget
7	Chicken nugget and Rice
8	French fries
9	Chicken nugget and French fries
10	Swedish meatball with egg noodle
11	Mashed Potato
12	Pork and Turkey patty with cheese sauce
13	Corn
14	Fried chicken with white source
15	Gravy meat Loaf
16	Fettucine alfredo
17	Turkey with bread
18	Pea
19	Spaghetti with meatballs
20	Meatballs
21	Tender chicken patty
22	Chocolate brownie
23	Chicken
24	Port lib patty with BBQ sauce
25	Apple crumble
26	Beef enchilada with chili and cheese sauce
27	Authentic refried bean
28	Cocada pudding
29	Chicken with BBQ sauce
30	Pork Rib and Mashed potato
31	Green bean
32	White meat chicken



/storage/emulated/0/DCIM/Camera/FoodDB_Color/58.jpg
/storage/emulated/0/DCIM/Camera/FoodDB_Thermal/58.jpg

Food Identification Processing....
Select a food to see the nutrition inform..

United States Department of Agriculture
Agricultural Research Service
National Nutrient Database for Standard Reference Release 27

Basic Report: 21139, Fast foods, potato, mashed

Nutrient values and weights are for edible portion.

Nutrient	Unit	1 Value per 100 g	1 cup 242g
Proximates			
Water	g	79.62	192.68
Energy	kcal	89	215
Protein	g	1.65	3.99
Total lipid (fat)	g	2.82	6.82
Carbohydrate, by difference	g	14.65	35.45
Fiber, total dietary	g	1.3	3.1
Sugars, total	g	0.50	1.21
Minerals			
Calcium, Ca	mg	18	44
Iron, Fe	mg	0.31	0.75
Magnesium, Mg	mg	15	36
Phosphorus, P	mg	59	143
Potassium, K	mg	286	692
Sodium, Na	mg	306	741
Zinc, Zn	mg	0.22	0.53
Vitamins			
Vitamin C, total ascorbic acid	mg	0.0	0.0
Thiamin	mg	0.015	0.036
Riboflavin	mg	0.015	0.036
Niacin	mg	1.080	2.614
Vitamin B-6	mg	0.120	0.290
Folate, DFE	µg	9	22
Vitamin B-12	µg	0.07	0.17
Vitamin A, RAE	µg	43	104
Vitamin A, IU	IU	185	448
Vitamin E (alpha-tocopherol)	mg	0.42	1.02
Vitamin D (D2 + D3)	µg	0.0	0.0
Vitamin D	IU	0	0
Vitamin K (phylloquinone)	µg	5.9	14.3
Lipids			
Fatty acids, total saturated	g	0.577	1.396
Fatty acids, total monounsaturated	g	0.720	1.742
Fatty acids, total polyunsaturated	g	1.342	3.248
Fatty acids, total trans	g	0.105	0.254
Cholesterol	mg	0	0
Other			
Caffeine	mg	0	0

National Nutrient Database for Standard Reference
Release 27 Software v.2.2.6
The National Agricultural Library

Figure 3.2: MT-Diet Android Application.

3.5.1 Image Registration

A perspective projective transformation (Homography), which requires at least four matching feature pairs between two images, is the popular technique for the image registration process. Since the Homography works under an assumption that four corresponding feature point pairs between two images are correctly identified, the feature matching process is very critical for successful image registration. Initial research has presented a manual process for selecting and matching these four feature point pairs but this requires a high user-intervention. An alternative is to extract many feature point candidates and match these features using RANSAC [5], which leads to an automatic image registration process. In this method, the various feature points are matched by taking into account the texture of the neighbors for each of these points. However, since the intensity of the various pixels in images of different spectra are quite different, it is impossible to match the feature points based on the

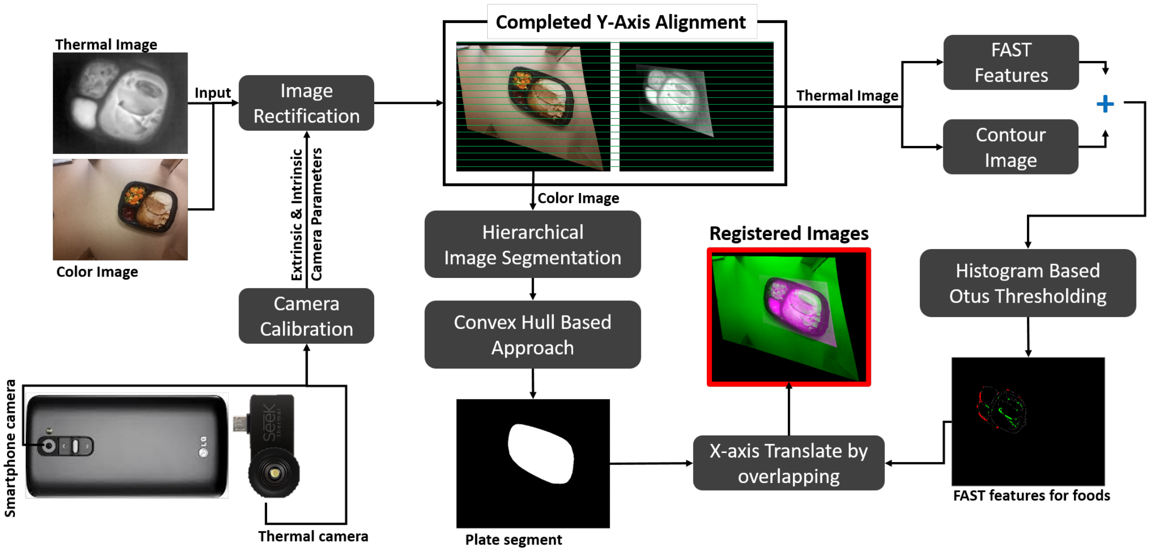


Figure 3.3: Image Registration Overview.

extracted texture information from these images. To circumvent this problem, several researchers have presented alternative techniques that rely on finding objects in the two images with a definite shape such as circles or lines [43] or a human body [56]. However, it is not feasible to apply such techniques in this project because food items on a plate of cooked food may not have items of any definite shape. Thus, in this section, we introduce my novel image registration approach between thermal food plate image and color food plate image. Intuitively, this approach is to align Y-axis using an existing image rectification technique at first and X-axis using a presented translation as seen in Fig. 3.3.

Y-axis alignment (Image Rectification)

The color and thermal images are transformed such that: a) for both color and thermal all epipolar lines are parallel to the horizontal axis, and b) corresponding feature points in both the images have same y co-ordinate as seen in Fig. 3.5. Image rectification consists of: a) camera calibration and b) planar rectification. Camera calibration is a one-time pre-processing task since the relative locations of the thermal

Algorithm 1 Image Registration Algorithm

```
Input:  camera_Parameters, Color, Thermal
Output:  Registered_color, Registered_thermal, Registered_contour_color
#Y-axis alignment
Rectified_color, Rectified_thermal ← rectify(camera_Parameters, Color, Thermal)
#Pre-processing for X-axis alignment
Color_contour ← gPb-owt-ucm(color)
HE_thermal ← Histogram_Equalization(thermal)
Thermal_contour ← gPb-owt-ucm(HE_thermal)
Thin_thermal_contour ← Zhang_Suen_thin(Thermal_contour)
Rectified_color_contour, Rectified_thermal_contour ← rectify(camera_Parameters, thin_color_contour, thin_thermal_contour)
#Thermal image process for X-axis alignment
FAST_features_candidate ← find_FAST(Rectified_thermal)
FAST_features_locations ← filtering(FAST_features_candidate, Rectified_thermal_contour)
FAST_foods, FAST_non_foods ← Otsu(FAST_features_locations, Rectified_thermal)
#Color image process for X-axis alignment
Plate_candidates ← find_Plate_Candidate(Rectified_color_contour)
#Find for X-axis translate parameter
for i ← 1 to size(Plate_Candidates) do
    N_FeatureMatching, X_translateValue ← Overlap(Plate_Candidates[i], FAST_foods)
end for
Translate_Para ← X_translateValue[ max_index(N_FeatureMatching) ]
#Clarify output
Registered_color ← Rectified_color
Registered_contour_color ← Rectified_color_contour
Registered_thermal ← translate(Rectified_thermal, Translate_Para)
```

camera and the embedded smartphone camera are not expected to change during usage. In the other words, camera calibration is done in order to obtain the intrinsic and extrinsic parameters of the camera. These parameters can then be fed to the planar rectification algorithm. For the camera calibration process, the two cameras are used to take images of the same setup. The setup is such that edge features can be easily determined from the two images. Once the edge features are determined, the calibration algorithm proposed by Zhang et al. [57] is used. However, the problem

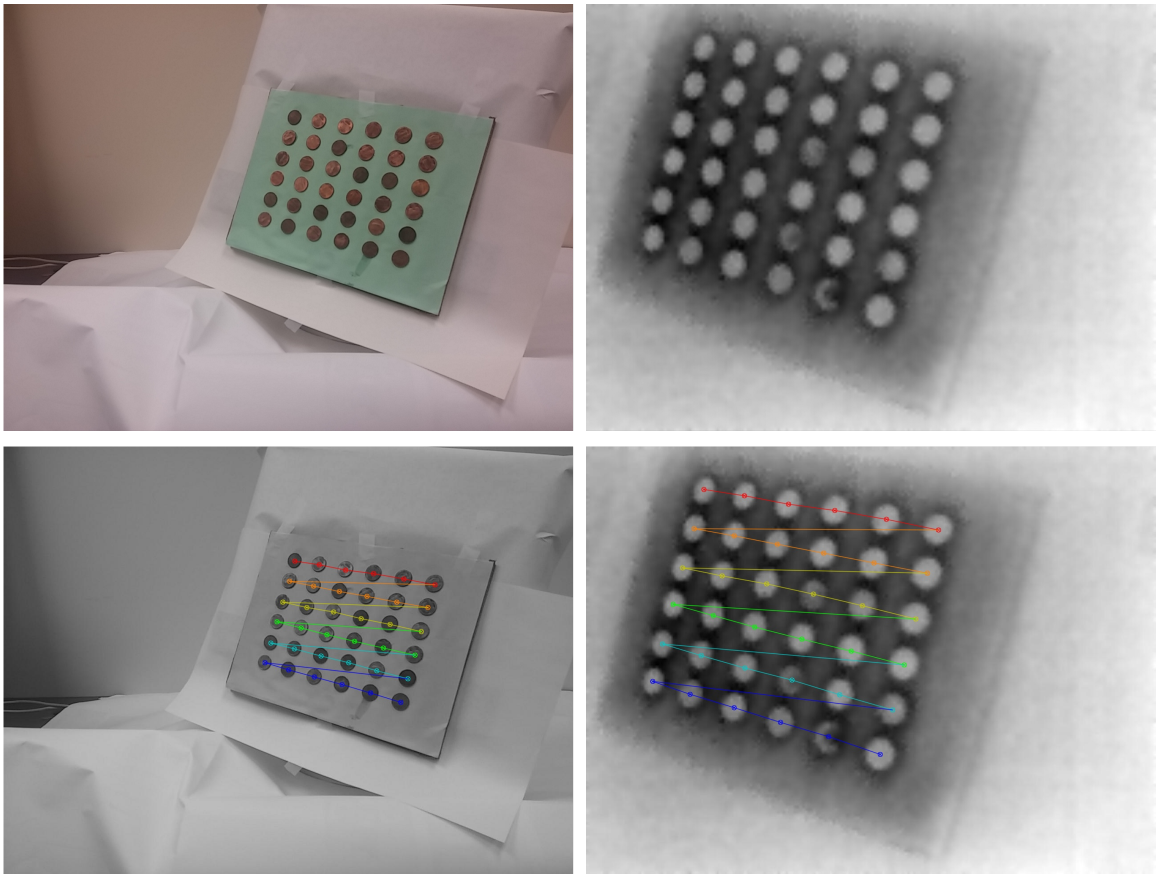


Figure 3.4: Top Two Images Are The Camera Calibration Tool By Taking Color Camera (left) And Thermal Camera (right). Down Two Images Are The Circle Detection Output Of Top Images.

is that the setup of a checkerboard typically used for camera calibration does not apply for thermal cameras because the corner edges in the checkerboard are not visible due to the lack of difference in temperatures. Hence, we use a setup that emulates the checkerboard using coins. The centers of the coins are utilized as feature points instead of the corner edge in the checkerboard. The setup is then cooled in a refrigerator. The resulting thermal image as seen in Fig. 3.4, is used to obtain the edges of the circle using Hough circle detection [58]. As seen in Fig. 3.5, intrinsic and extrinsic parameters obtained from the camera calibration process were utilized for rectification.

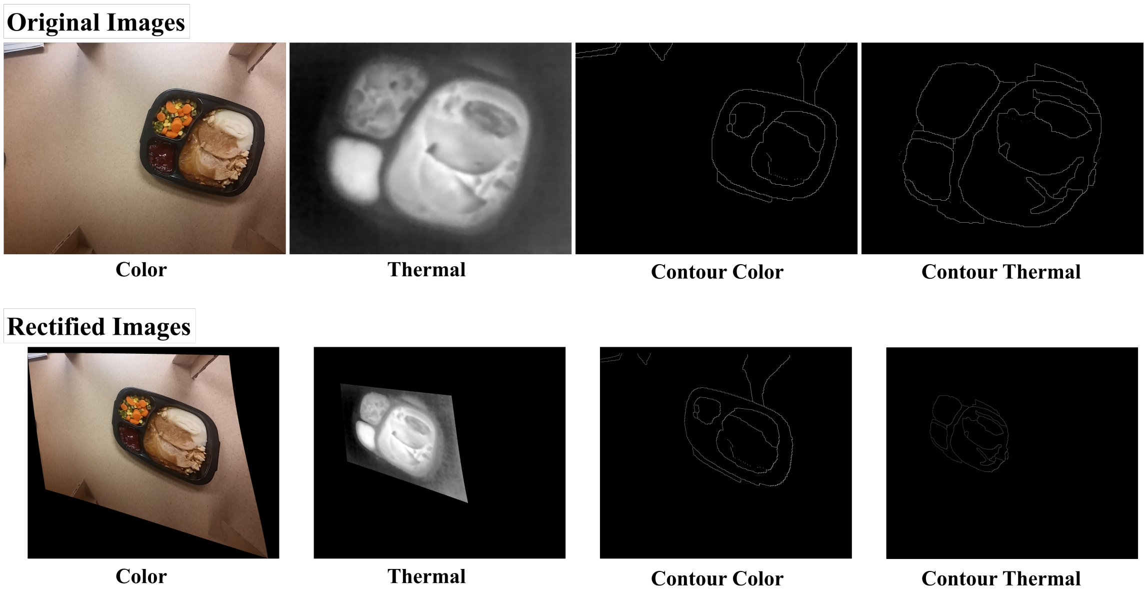


Figure 3.5: Rectification Processing Using Intrinsic And Extrinsic Parameters Of Camera.

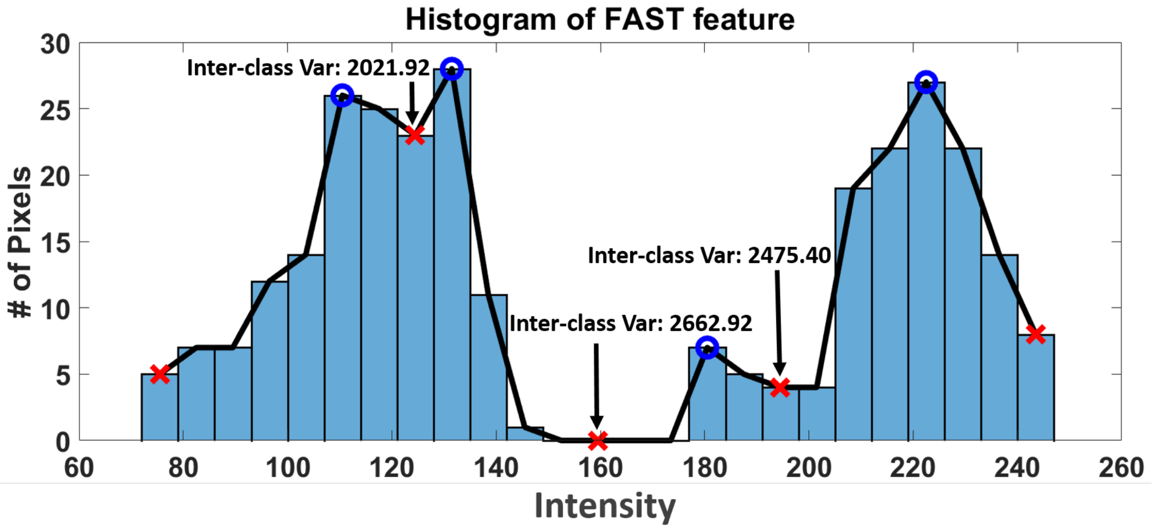


Figure 3.6: FAST Feature Histogram.

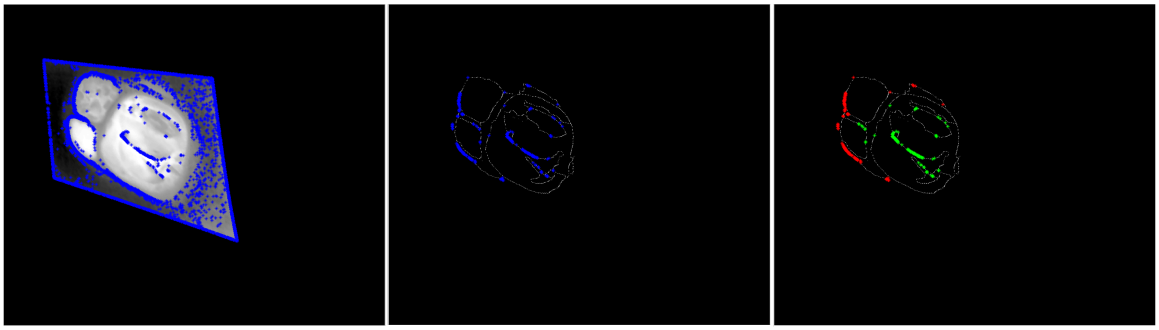


Figure 3.7: Food FAST Feature Identification Based On Thermal Image. Left Is All FAST Features Obtaining From Rectified Thermal Image. Center Is FAST Features On Rectified Contour Image. Right Is Classified FAST Features. Red Points In Right Is Non-food FAST Features And Green Points Is Food FAST Features.

X-axis alignment (Translation)

A highly accurate contour detection for both color and thermal images is required for X-axis alignment pre-processing task. To achieve the contour image, we use a combination of three algorithms, Global Probability of Boundary (gPb), Oriented Watershed Transform (OWT), and Ultra-metric Contour Map (UCM) as suggested by Arbelaez et al. [59]. After performing the watershed algorithm to the contour image, food, plate and background segments in the image can be assigned different labels. However, we found that the contour detection (gPb-owt-ucm [59]) is not able to obtain accurate boundaries in thermal images specially between the plate and the background. This is possibly because the temperature difference between the plate and the background may not be considerable when compared to the temperature difference with food portions which results in high pixel intensity for food segments but not for the plate or background segments. However, this boundary information is essential for proper X-axis alignment.

Thus, we first perform histogram equalization and then do contour detection (line 2-3 in Algorithm 1). Next, we find feature points in thermal image using FAST algorithm [60]. The FAST feature is a neighbor pixel intensity based feature. In this step (line 7-9 in Algorithm 1), two processes were performed as seen in Fig. 3.7: 1) Applying FAST feature detection to histogram equalized thermal image, 2) Checking whether the detected FAST feature points are on the boundary of thermal contour image or not. In order to perform this step in a time efficient manner, contour image thinning is required. This is because the execution time increases with the number of thermal feature points. Hence, Zhang-Suen thinning [61] was performed before this step (line 4 in Algorithm 1). The pixels in the thermal image can be roughly clustered as food or non-food features candidates. For this purpose, the Otsu′s method [62] was utilized. First of all, we draw a histogram of the intensity for all feature locations in the thermal image. Then, a line-graph based on the values of this histogram was drawn as seen in Fig. 3.6. The minima values of this graph become the candidates for the thresholding value (θ). Among these candidates, the value with the highest inter-class variable is selected as the thresholding value. The formula for the inter-class variable is given by the Equation 3.1. In the equation, W_p is the probability of plate, $W_p = \sum_{i=0}^{\theta-1} H(i)$, where H is the histogram intensity, and W_f is the probability of food, $W_f = \sum_{i=\theta}^{255} H(i)$. μ_p is the plate class mean, $\mu_p = \sum_{i=0}^{\theta-1} i\frac{H(i)}{W_p}$, and μ_f is the food class mean, $\mu_p = \sum_{i=\theta}^{255} i\frac{H(i)}{W_f}$.

$$Variable = W_p W_f (\mu_p - \mu_f)^2 \tag{3.1}$$

Thus, by this procedure, we have identified the FAST features that cluster the food pixels and plate pixels separately using the thermal image as seen in Fig. 3.7.

The next step in X-axis translation is to match the portion of the image that has the food between thermal and color images. For the color image, the largest convex hull (line 10 in Algorithm 1) obtained within the image based contour segmentation results is selected as the plate portion as seen in Fig. 3.8. This is based on the justified assumption that food does not overflow the plate.

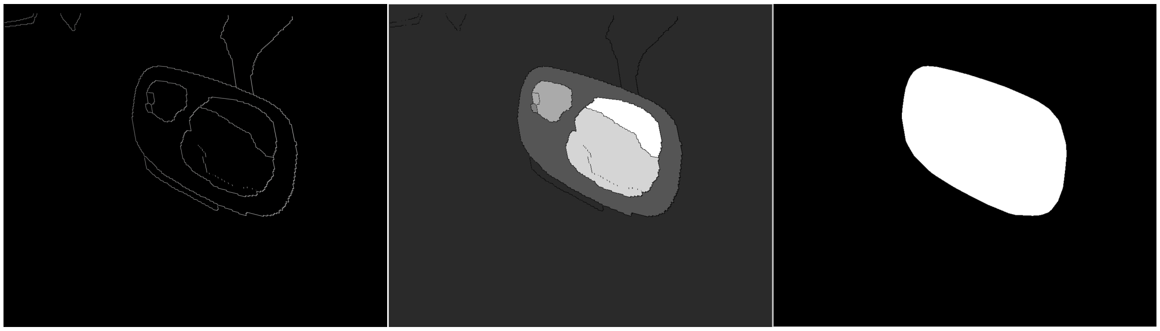


Figure 3.8: Plate Segmentation Identification From Rectified Contour Color Image.

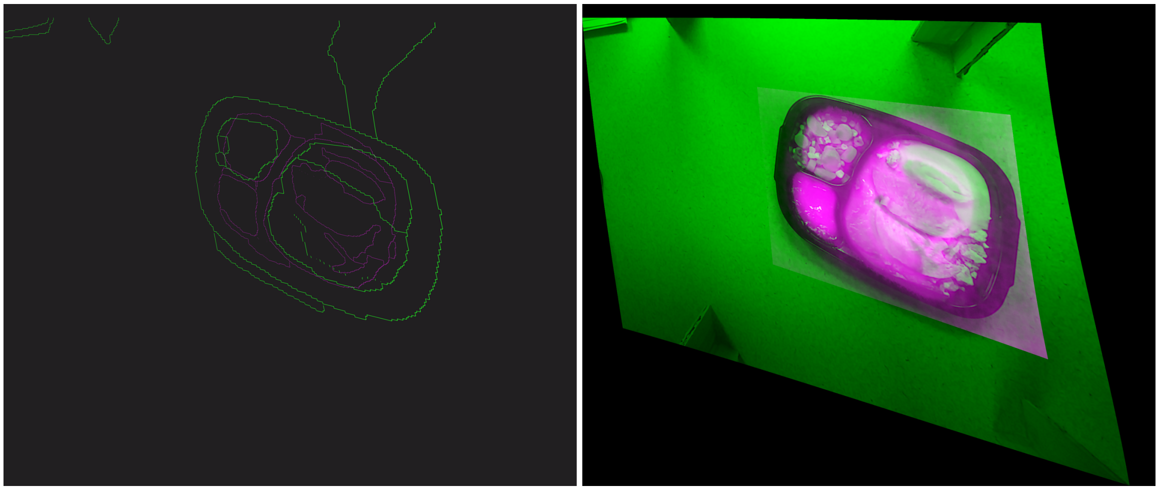


Figure 3.9: Image Registration Output. Left Is Contour Image Based Output And Right Is Color Image Based Output.

Finally, the plate portion obtained from the color image was overlapped to the FAST features for food obtained from thermal image (line 11-14 in Algorithm 1). We then compute the pixel by pixel overlap for the neighborhood of pixels related to the plate from both the images. Hence, we obtain the pixel by pixel correspondence for the color and thermal images as seen in Fig. 3.9.

3.5.2 Food Segmentation

The process of food segmentation is defined as follows:

Definition

Food Segmentation is a process that takes the following inputs:

a) Registered color image of served hot food and food plate in a background

b) Registered thermal image of served hot food and food plate in a background.

It assumes that

a) Background temperature $<$ Food plate temperature

b) Food plate temperature $\ll$ Food temperature.

It outputs the following:

a) Number of different food items on plate

b) Cropped region of each unique food item.

The main goal for the food segmentation method is to enhance the segmentation accuracy by extracting and merging knowledge obtained from cameras in different spectra. To be specific, food items that have the same color as the plate cannot be distinguished well using color images alone. Adding knowledge from thermal images helps in these cases due to the difference in temperature between the food and the plate. If, however, the food items are not sufficiently heated then the segmentation done using thermal images can be enhanced using the information from color images.

The inputs for this method are the image registration output such as registered color food image, registered contour image obtained from color food image, and registered thermal food image. The outputs of this method is an image with only food pixels, excluding the background and the plate. Also, the method consists of three steps: i) Hierarchical Image Segmentation (HIS) for the color food image, ii) Dynamic Thermal Thresholding (DTT) for the thermal food image, and iii) Region of Foods (ROF) detection.

Hierarchical Image Segmentation (HIS)

Using the registered color contour image that was obtained from the image registration, the watershed algorithm segments the registered color image based on identified contours and provides different labels to each segment as proposed by Arbelaez et al. [59].

Dynamic Thermal Thresholding (DTT)

In practice all cooked food may not be heated to the same temperature. Hence, in the gray scale thermal image, the intensity of food plate and background will vary from image to image, but there is a consistent trend that the food temperature is higher than food plate temperature and the plate temperature is higher than background temperature. Therefore, the DTT algorithm was developed to utilize this observation and segment food items from the plate and background.

The first step of DTT is to seek the temperature of the background pixels. In the experiments, we assume that the background is cooler than the cooked food and the food plate. This assumption is reasonable since no restaurant table will be as hot as the plate or the food. This enables a simple threshold based elimination. Based on the properties of the thermal camera, intensity value less than 150 was considered as background. In Equation (3.2), the thermal image with background cancellation

(rBP) given by Equation 3.2.

$$rBP(i,j) = \begin{cases} 0, & \text{if } ThImg(i,j) < 150 \\ ThImg(i,j), & \text{else} \end{cases} \tag{3.2}$$

The next step in DTT is to find the plate temperature. In the background eliminated image, we search for pixels with the highest difference in gray scale intensity from its neighbors. Hence for each pixel $rBP(i,j)$, 3 × 3 window ($W(i,j)$) is utilized as shown in Equation 3.3 to generate a differential matrix ($diff\_mat(i,j)$) as shown in Equation 3.3. The members of the $diff\_mat$ means the difference between maximum and minimum element in each W. In $diff\_mat(i,j)$ if at least one member of the $W(i,j)$ window is a background pixel, then this difference is assigned to zero.

$$W(i,j) = \begin{bmatrix} \text{rBP (i-1 , j-1)} & \text{rBP (i-1 , j)} & \text{rBP (i-1 , j+1)} \\ \text{rBP (i , j-1)} & \text{rBP (i , j)} & \text{rBP (i , j+1)} \\ \text{rBP (i+1 , j-1)} & \text{rBP (i+1 , j)} & \text{rBP (i+1 , j+1)} \end{bmatrix}$$

$$diff\_mat(i,j) = \begin{cases} 0, & \text{if } Min(W(i,j)) = 0 \\ Max(W(i,j)) & -Min(W(i,j)), \text{else} \end{cases} \tag{3.3}$$

The (x,y) position with the maximum value in $diff\_mat$ represents a window $W(x,y)$ that has both food portions and plate. This window $W(x,y)$ was utilized to compute a threshold gray scale intensity value. Any pixel with greater intensity than this threshold can be classified as food, while any other pixel as plate. The threshold (T_p) is considered to be the median in $W(x,y)$ as seen in Equation 3.4.

$$T_p = median(\ W(x,y)\ ) \tag{3.4}$$

Although DTT successfully removes the plate and background pixels, it does not segment individual food items. Further, the result of DTT thresholding has salt-and-pepper noises and may also remove certain food pixels, which are not heated

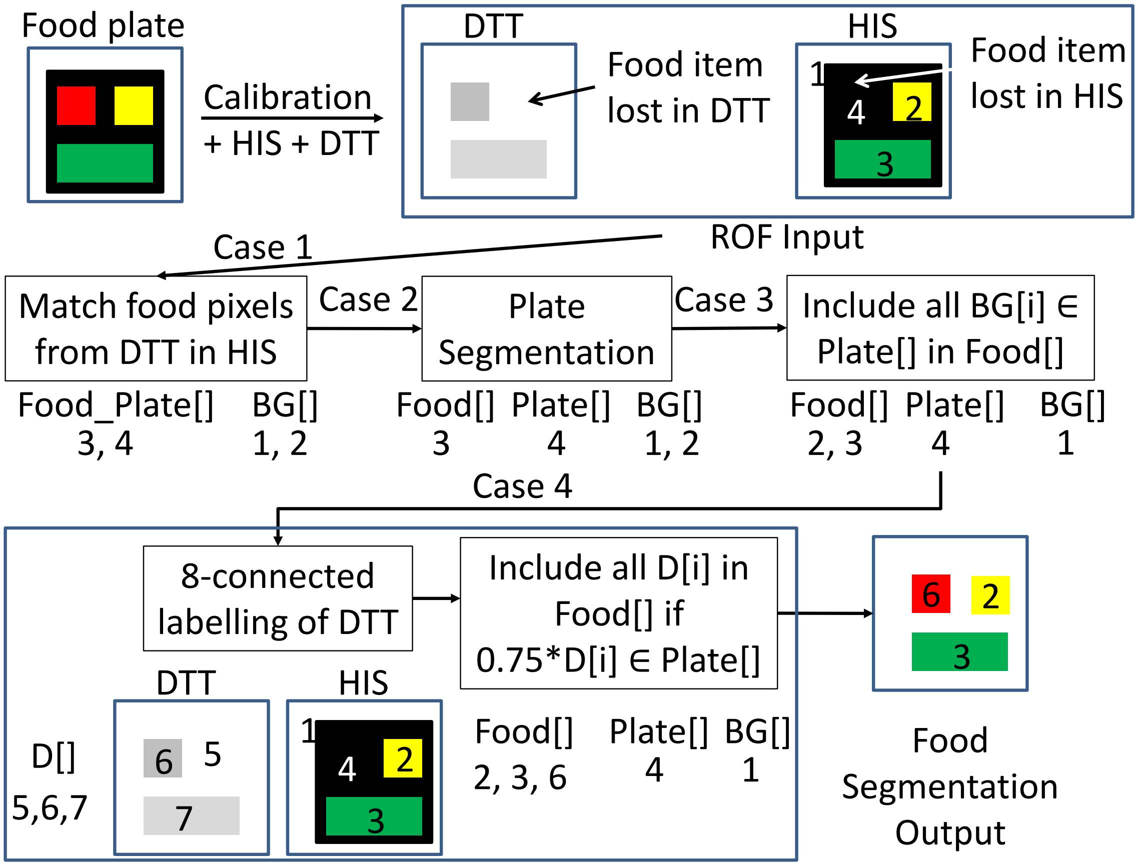


Figure 3.10: The ROF Algorithm Showing Four Cases That Might Cause Errors And Their Corresponding Solutions.

to sufficiently high temperatures. Morphology techniques such as opening and closing [63] was performed to remove the salt-and-pepper noises, but it is hard to recover removed food pixels. Using the Region of Foods (*ROF*), we combine the color and thermal images to reduce noise and also accurately identify regions of food in the image.

Region of Foods (ROF)

Although the DTT method is successful in removing background and majority of food plate, four problems exist which are solved using the presented region of food (ROF) detection as shown in Fig. 3.10.

Case 1: Missing labels - The HIS generates only segments, does not specify whether it is a food portion or not.

Solution: We combine the HIS with the DTT, which has already separated the food portions from the plate and the background based on thermal threshold. The HIS and the DTT segmentation portions are compared with respect to pixel indexes. If the indexes of each HIS segmentation portion match with indexes of food portion as identified in DTT, the HIS segmentation portion becomes candidates of ROF. This step outputs background segments and those that are either food or plate (Case 1 in Fig. 3.10).

Case 2 Wrong labeling - Plate portions near food items get heated enough to be included in food segment outputs of the DTT algorithm. In such a scenario, in the solution for Case 1, the entire HIS segmentation, which corresponds to the plate area, may be wrongly classified as food (Fig. 3.11).

Solution: Case 1 has isolated the background portion. We scan the edge image starting from the four corner faces of the image to identify four corner pixels of plate. If three or more pixels amongst these have the same label numbers as assigned by the HIS algorithm, the corresponding HIS segment is considered as plate. The assumption is that the plate is not overflowing with food. The output of this case is segments which are only food items, segments which have majority of plate, and background segment (Case 2 in Fig. 3.10).

Case 3: Missing food items in Thermal image - Food portions which are not sufficiently heated may be removed in the thermal image after the execution of the DTT algorithm and hence may not be detected by the solution of Case 1 as shown in Fig. 3.12.

Solution: To solve this problem, we utilize the assumption that all food portions are contained within the plate. For each segmented portion of the HIS algorithm, we

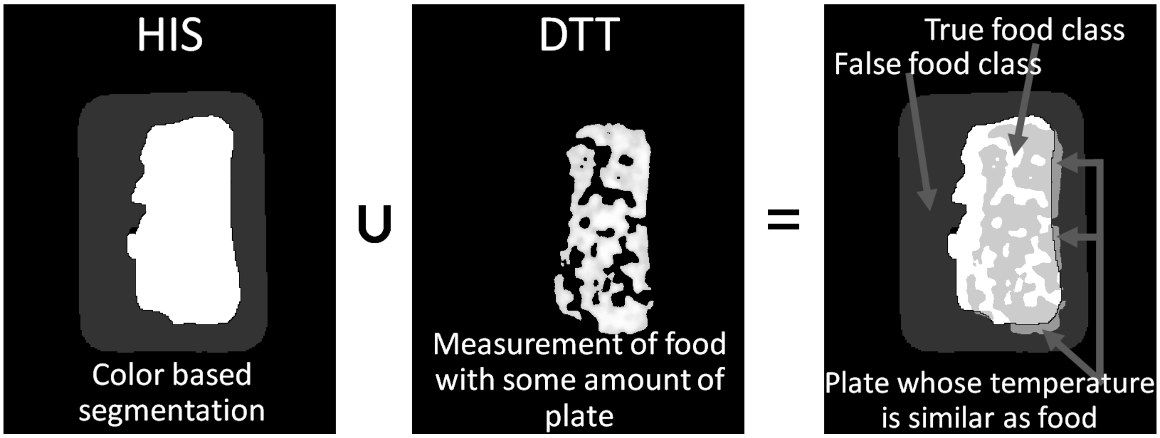

HIS
Color based segmentation
U
DTT
Measurement of food with some amount of plate
=
True food class
False food class
Plate whose temperature is similar as food

Figure 3.11: Case 2: Elimination Of Plate Portions That Are Heated To Nearly Similar Temperatures As Food.

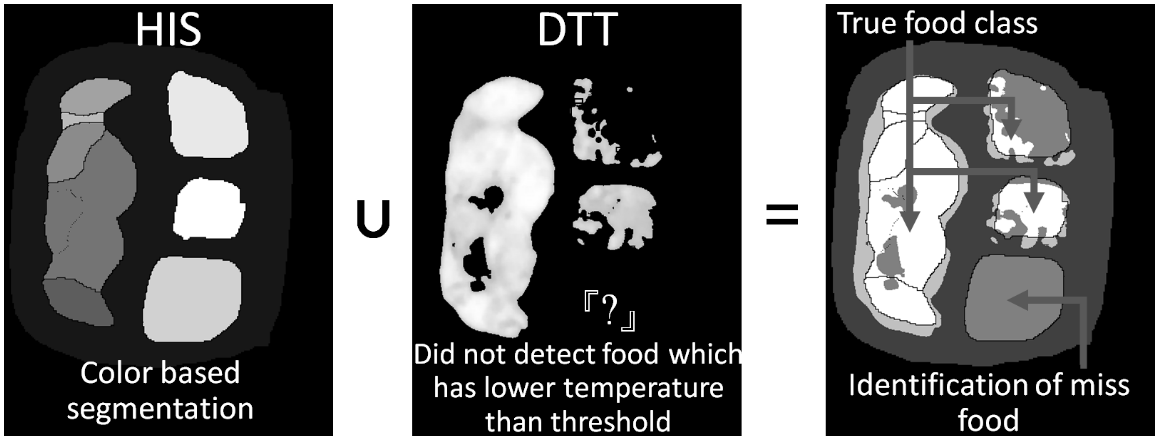

HIS
Color based segmentation
U
DTT
『?』
Did not detect food which has lower temperature than threshold
=
True food class
Identification of miss food

Figure 3.12: Case 3: Recovery Of Food Portions That Are Not Sufficiently Heated.

compute the median pixel. If the median pixel is within the plate but not included in the candidate ROF regions as derived from Case 1, then we consider the segmented portion as a candidate ROF.

Case 4: Missing food items in color image - Food items, which have the same color as the plate are eliminated in the HIS image segmentation method as shown in Fig. 3.13.

Solution: This problem can be solved using the thermal image, because the even if the food color is the same as the plate, in the thermal image, the food temperature will be higher than the plate. To retrieve such food portions, each segmented portion

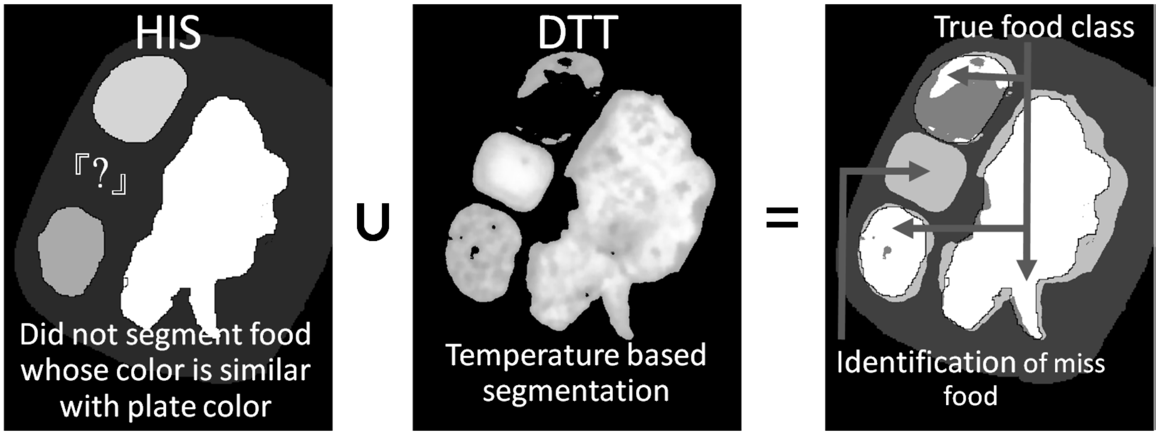

HIS
『?』
Did not segment food whose color is similar with plate color
U
DTT
Temperature based segmentation
=
True food class
Identification of miss food

Figure 3.13: Case 4: Recovery Of Food Portions With The Same Color As The Plate.

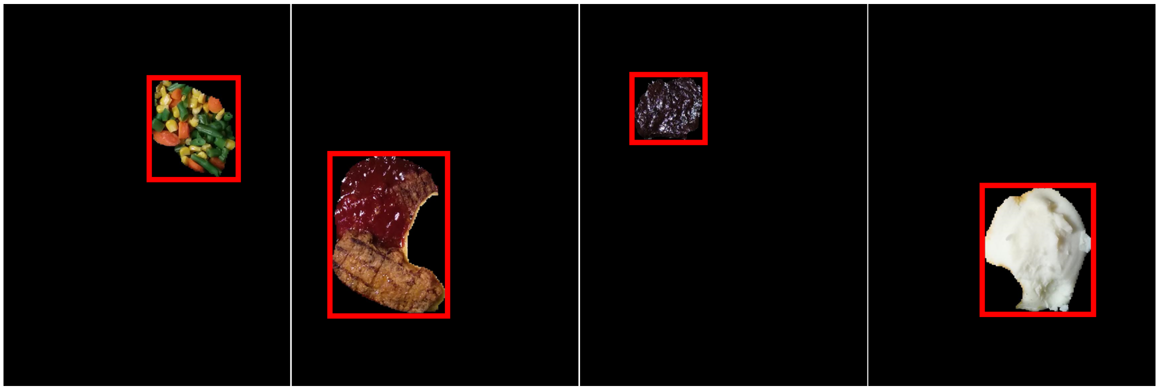


Figure 3.14: Example Of ROF Output With A Red Bounding Box: There Are Four Types Of Foods Such As Vegetable, Pork Lib Patty With BBQ Sauce, Chocolate Brownie, And Mashed Potato In The Rectangle.

from DTT output is labeled using connected component labeling[64]. If over 75% of the connected component from DTT output is considered as plate portion in HIS, the DTT labeled portion is considered as the candidates of ROF. As a result of the four above-mentioned solutions, we obtain the candidates of ROF, however these candidates still have noises because HIS is an approximate segmentation method. To get accurate food portions, we use the Grabcut algorithm [45]. Given a selection of potential object and background, Grabcut provides more accurate object boundary based on the color distributions of the object and background. Typically, Grabcut is used in visual image based food segmentation in a semi-automatic setting, where the user is asked to select potential food portions also noted as region of interest. MT-Diet considers ROFs as region of interest in Grabcut and hence eliminates the need for user intervention.

3.5.3 Food Identification

The process of food identification is defined as follows:

Definition

Food identification is a process that takes the following an input:

- A food portion image obtained from food segmentation.

It performs

a) feature extraction from the food image with features such as RGB color image, Gabor texture data, and histogram of oriented gradients,

b) reduce the number of features if needed using principal component analysis (PCA) and kernel principal component analysis (KPCA), and

b) Support Vector Machines based food classification.

It outputs:

- type of food items on the plate.

The food identification process takes the ROF output from the food segmentation process and outputs the food type by matching the image with a food image database. The first step in the food identification process is feature extraction.

Feature Extraction

For feature extraction, three features extraction methods are performed: color, texture and histogram of oriented gradients [47] from color and thermal images.

First of all, RGB histogram is applied to extract color feature. In RGB image, each color channel (Red, Blue, Green) has an intensity within a range from 0 to 255. We generate 32 histogram bins of each color channel so that the dimension of the color feature vector is 32768 ($32 \times 32 \times 32$).

The Gabor filter method is employed to extract the texture feature. The segmented ROF portions should be resized to a standard size, since different image sizes affect the texture feature vector size. Therefore, the ROFs were resized to a 400×400 image.

As shown in [65], we extract the variations in different frequencies and orientations in the images. The size of the texture feature vector is the size of each food image (400×400) multiplied by the number of scales and orientations (5×8) divided by the row and column down-sampling factors (4×4). Therefore, the dimension of a texture feature vector is $400 \times 400 \times 5 \times 8 \ / \ (4 \times 4) \ = \ 400000$.

Finally, we extract the histogram of oriented gradients (HOG) feature [47]. The cropped and resized images were utilized again to decrease an impact of the black background feature. Then, each food image is divided into 16 windows and oriented gradients for each of the windows are calculated using the histogram of the 36 bins. Therefore, the size of HOG feature vector is $16 \times 36 \ = \ 576$.

Feature Fusion

The identification process requires a database of features of different types of food. The SVM is used to learn how to differentiate between different foods types in the database. After the learning phase, the SVM is provided with the feature vector of an unregistered (not in the training database) food image. The SVM then attempts to classify the given input feature vector into a particular class using different distance functions known as the kernel. Feature selection is an important trade-off between identification accuracy and response time of the dietary feedback.

Usage of a single feature such as either RGB or Gabor texture or HOG, may be computationally efficient, however, it has a drawback that the accuracy is greatly influenced by the nature of the database. For example, when the color feature is employed to classify, it is hard to classify foods with the same color like corn and cheese macaroni. To overcome such inaccuracies, one can fuse several features as shown in Fig. 3.15. However, feature size drastically increases resulting in higher computational time, and increasing response time of MT-Diet. Further a simple concatenation of the color, texture and the HOG feature vector results in high feature size (433344) hence increases the SVM execution time (Fig. 3.15). Therefore, we employ the dimensionality reduction techniques such as Principal Component Analysis (PCA) [66] and PCA with Gaussian Kernel Principal Component Analysis (KPCA) [67]. In our first feature reduction attempt, we consider the concatenated feature vector and perform PCA and KPCA, and select the top 200 Eigen vectors from the feature matrix (Fig. 3.15). Using this Eigen vector, we recreate the feature vector. Therefore, the feature dimension reduces to 200. The accuracy of the method, however, actually does not increase compared to the accuracy by using single feature vector because the color feature vector dominates the other feature vectors. In the second method, PCA and KPCA are applied separately to each feature before concatenating the feature vectors. This not only decreases feature vector size but also increase accuracy. The size of each feature is reduced to 100 and the whole feature vector size is 300 (Fig. 3.15).

3.6 Evaluations

We evaluate MT-Diet food segmentation and identification method using experiments on cooked meals. Each food plate can have either single or multiple food items.

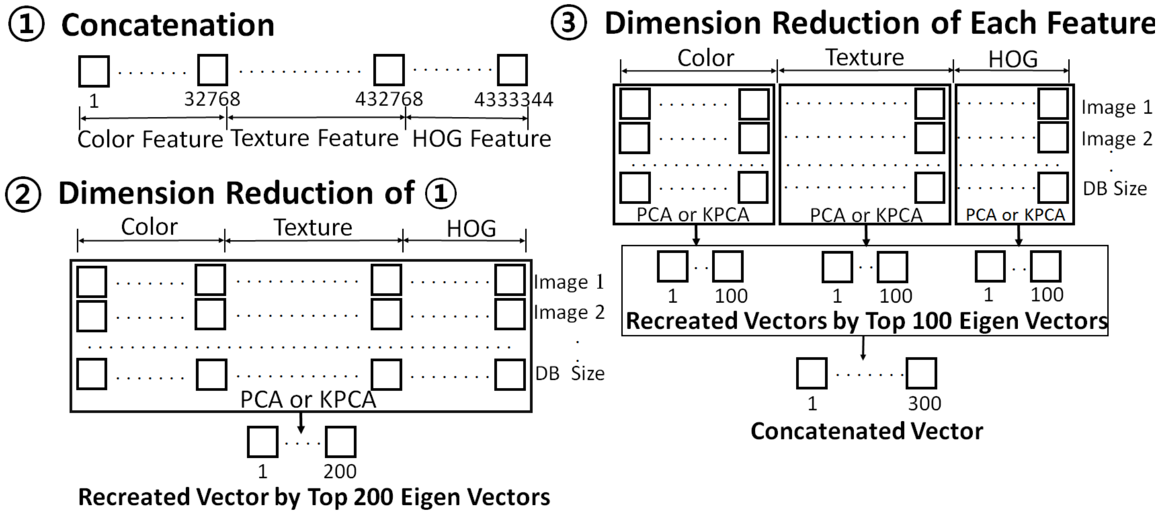


Figure 3.15: Three Feature Fusion Methods.

The shape of the plates can be different based on the number of food items available. Table 3.2 shows the food database with type of food and quantity of each food item in the plate type is shown. The database consists of ROF images from all 80 food plates. The total size of the database is 244 different images and Mashed potato has the largest number of appearances since it is the most popular in frozen food plates.

To implement the SVM classifier, the libsvm[68] API was utilized. The software supports not only kernels such as Polynomial (Poly), Radial Basis Function (RBF), and Sigmoid but also k-fold cross validation which helps to ensure the statistical reliability of the results. In our experiments, all kernels were evaluated with 5-fold cross validation with the three types of individual features and three feature fusion methods.

There are three important metrics for evaluating MT-Diet: a) accuracy for food segmentation with respect to food portions identified by human judgment, b) accuracy of food identification with respect to known items on the food plate, c) execution time of food segmentation and food identification process.

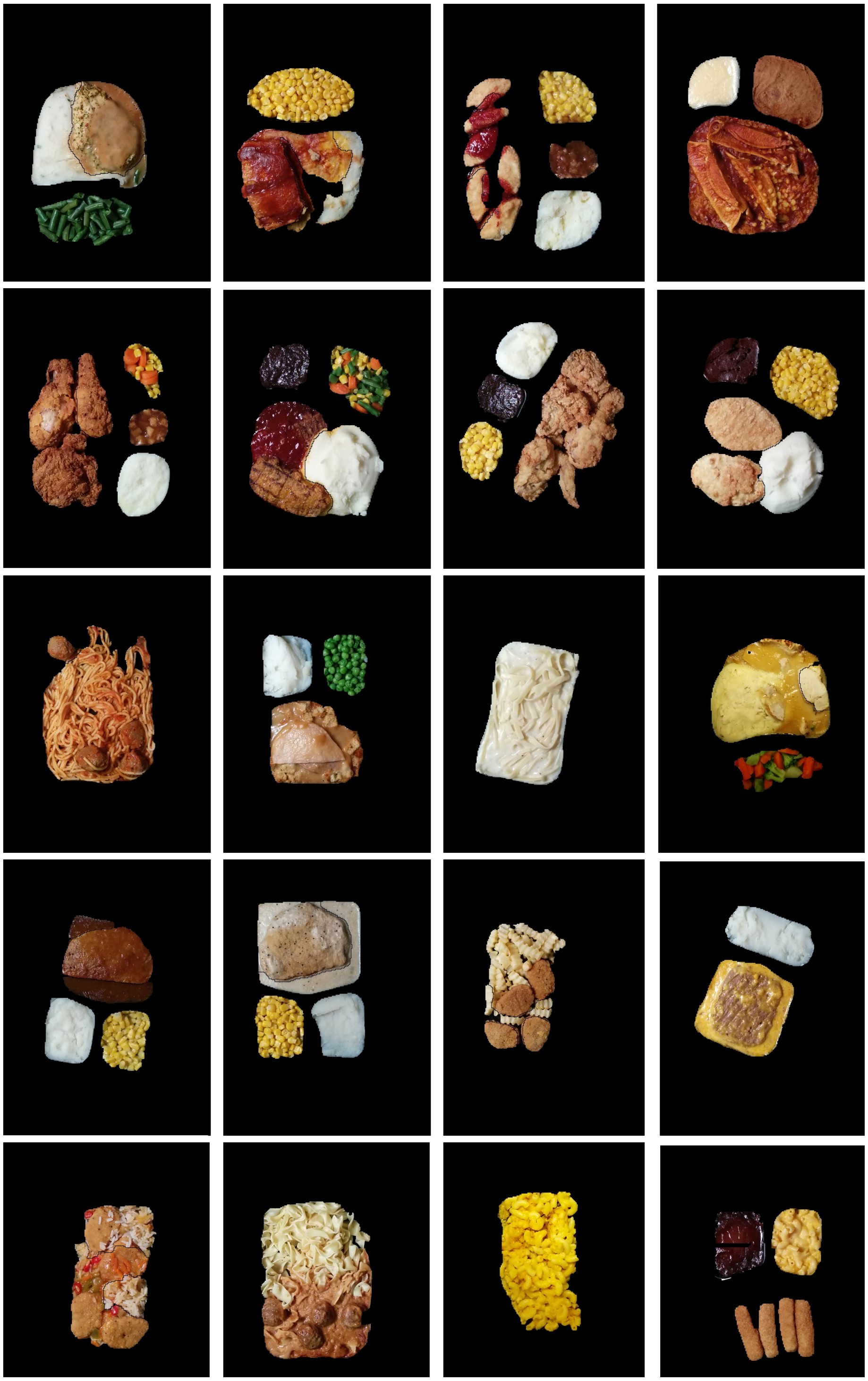


Figure 3.16: Segmented Food Images: 20 Different Type Of The Integrated Version.

3.6.1 Food Segmentation

To evaluate, the segmentation method, we focused on two issues: how well the background and plate pixels were removed and the accuracy of the food item count. An accurate output from food segmentation is critical for food identification because noises in segmented portions can lead to corrupted food features and hence reduce food identification accuracy. Fig. 3.16 displays the segmentation method output for 20 different frozen foods. A mathematical evaluation of accuracy is difficult since the ground truth itself is subjective in that it relies on the segmentation skills of a human. However, visual inspection of the images show that there are only a minimal number of pixels which belong to plate or background.

We also evaluated how well multiple foods were separated by counting the number of foods in an image and matching it with the number of food items counted by a human observer. In Fig. 3.17, X axis is the food image ID number and Y axis is the number of foods in the ID image. The bold line is for human counting and the dashed line is the total number of food items for the presented method. 8-connected labeling[64] was performed after the food segmentation processing. As a result, each food portion as well as the background have the different label numbers. Therefore, the total number of foods in the image is the number of the different label numbers without background label number (the number of the different labels - 1).

According to Fig. 3.17, the total number of food items identified by a human observer in all 80 food plates was 244, however the total number of food items distinguished by the method is 210. In 14 out of the 80 food plates at least one item was miscounted.

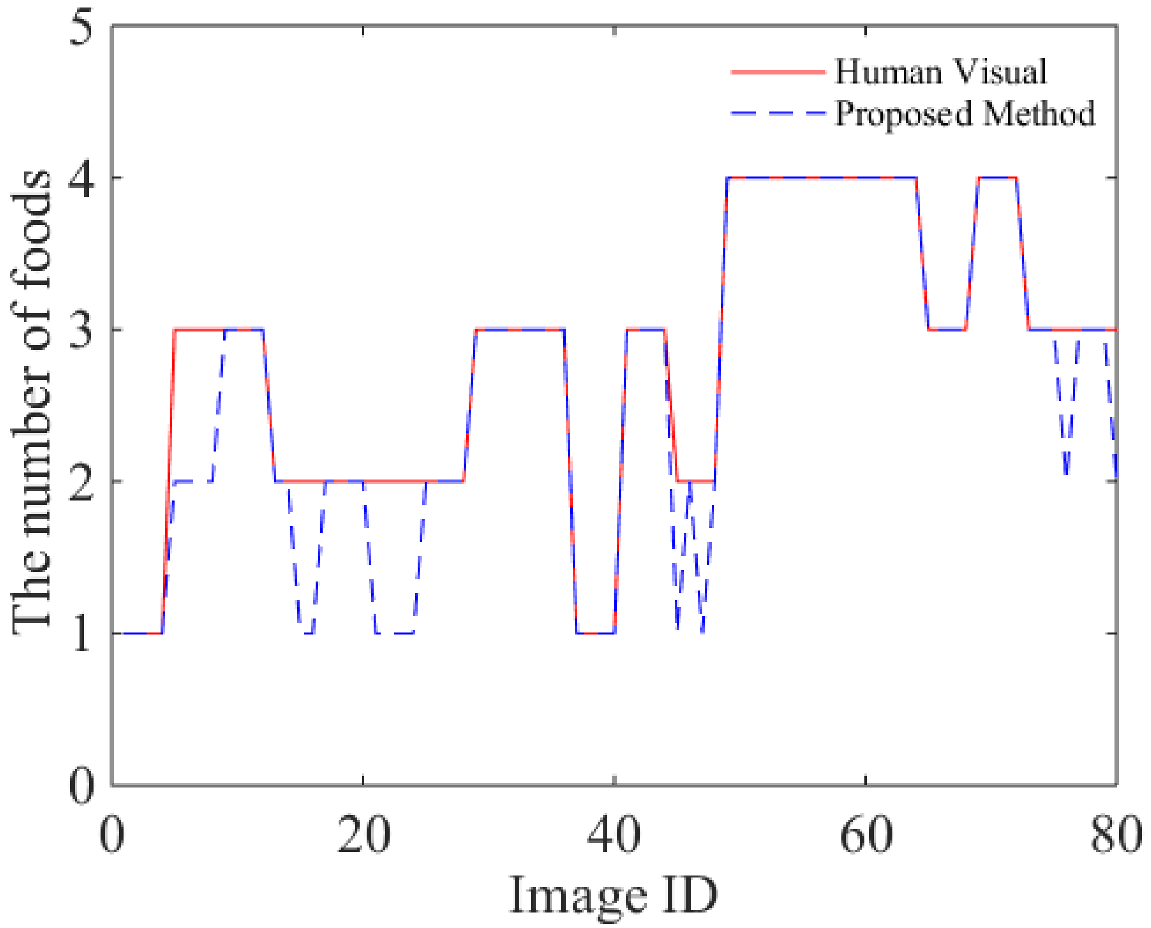


Figure 3.17: Compare The Number Of Foods Between Human Visual And Presented Method.

3.6.2 Food Identification

In the section, the food identification result will be discussed. In Table 3.3, we provide accuracy results using three features and three different ways to fuse them. The features used were: Gabor (texture), Histogram of Oriented Gradients (HOG), and RGB histogram. The first column (Feature) lists the type and the combination of the features used. The second column (Fusion Method) defines the method of fusion utilized. If only one feature was used, the fusion method is labeled as NTH since no fusion takes place. For methods that use multiple features, a separation is made on whether the features were concatenated or used separately. The next column (D

· R Method) defines the dimensionality reduction technique utilized which is either NTH, PCA or $KPCA$. The next four columns list the accuracies for the various SVM kernels. The next two columns list the statistical results of average and max accuracies. The execution times for all of the kernel types are in the next four columns followed by the average execution time. The last column (Feature Size) lists the total size of the features.

As seen from Table 3.3, the color feature (RGB) with KPCA was the best accuracy (88.11%) among the single features: HOG, Gabor, and RGB. And the accuracy of the two features fusion (88.93%) is better than the accuracy using single or all three feature fusion. Also, the *Separate* method is the best among the three fusion methods and $KPCA$ has higher accuracy than PCA. Therefore, the fusion of color and texture or the fusion of color and HOG with *Separate*, $KPCA$, and RBF kernel has the highest accuracy (88.93%) in Table 3.3.

3.6.3 Execution Time

In our food segmentation approach, there are four main tasks: DTT, HIS, ROF, and Grabcut. To analyze the execution time of these tasks, made a statistical execution time table, Table 3.4. According to the table, the execution time of HIS (83.08%) and Grabcut (12.78%) occupied 95.86% of the whole execution time. Also, since the Grabcut was performed to all segmented food images in a raw image, the *Grabcut* execution time was variable depending on the number of segmented image.

To evaluate the food identification execution time, there are two main tasks: Dimensionality reduction and SVM training depending on the kernel types. In Table 3.3, a sum of the execution time for these two tasks by the kernel types was displayed. According to the table, the average execution time of the RBF kernel is the highest and the average execution time for a linear SVM is the lowest.

Table 3.3: Accuracy And Execution Time Of Food Identification.

Feature	Fusion	D · R	Accuracy (%)						Execution Time (Second)					Feature
	Method	Method	Linear	Poly	RBF	Sigmoid	Ave	Max	Linear	Poly	RBF	Sigmoid	Avg	Size
Gabor	NTH	NTH	37.30	17.62	17.62	22.54	23.77	37.30	355.47	368.66	382.51	368.70	368.83	400000
		PCA	39.75	28.28	17.62	20.90	26.64	39.75	29.94	30.04	30.13	30.16	30.07	100
		KPCA	16.80	36.89	45.08	42.62	35.35	45.08	4.87	5.00	5.13	5.09	5.02	100
	Average		31.28	27.60	26.77	28.67	28.585	40.71	130.09	134.56	139.26	134.65	13.64	
	Best (KPCA, RBF)		39.75	36.89	45.08	42.62	35.35	45.08	4.87	5.00	5.13	5.09	5.02	
HOG	NTH	NTH	50.41	61.48	62.30	58.61	58.20	62.30	0.27	0.38	0.47	0.64	0.44	576
		PCA	49.18	60.66	63.11	59.84	58.20	63.11	0.10	0.28	0.35	0.31	0.26	100
		KPCA	32.79	57.38	56.97	55.33	50.62	57.38	0.08	0.26	0.33	0.31	0.24	100
	Average		44.13	59.84	60.79	57.93	55.67	60.93	0.15	0.31	0.38	0.42	0.32	
	Best (PCA, RBF)		50.41	61.48	63.11	59.84	58.2	63.11	0.08	0.26	0.33	0.31	0.24	
RGB	NTH	NTH	77.46	77.87	86.89	87.70	82.48	87.70	1.74	2.14	2.21	2.14	2.06	32768
		PCA	77.46	80.33	86.46	87.30	83.89	87.30	1.45	1.85	1.67	1.68	1.66	100
		KPCA	56.15	81.15	87.70	**88.11**	78.28	88.11	0.33	0.52	0.56	0.57	0.49	100
	Average		70.36	79.78	87.02	87.70	81.22	87.70	1.17	1.51	1.48	1.46	1.40	
	Best (KPCA, Sigmoid)		77.46	81.15	87.70	88.11	82.89	88.11	0.33	0.52	0.56	0.57	0.49	
HOG & Gabor	Concatenate	NTH	37.30	17.62	17.62	22.54	23.77	37.30	354.79	368.31	382.24	368.29	368.41	400576
		PCA	38.11	20.90	17.62	22.95	24.90	38.11	30.60	30.67	30.75	30.74	30.69	200
		KPCA	16.80	36.48	43.85	43.44	35.14	43.85	5.99	6.05	6.19	6.18	6.10	200
	Separate	PCA	39.75	28.28	17.62	20.90	26.64	39.75	30.02	30.14	30.21	30.26	30.16	200
		KPCA	33.61	59.02	59.43	58.20	52.57	59.43	4.91	5.10	5.13	5.16	5.07	200
	Average		33.11	32.46	31.23	33.61	32.60	43.69	85.26	88.05	90.90	88.13	88.09	
	Best (KPCA, RBF, Separate)		39.75	59.02	59.43	58.20	52.57	59.43	4.91	5.10	5.13	5.16	5.07	
HOG & RGB	Concatenate	NTH	83.20	83.61	84.02	84.02	83.71	84.02	2.10	2.26	2.66	2.26	2.32	33344
		PCA	84.02	83.20	84.43	84.02	83.92	84.43	1.49	1.64	1.71	1.73	1.65	200
		KPCA	35.25	75.82	79.92	80.33	67.83	80.33	0.51	0.64	0.74	0.74	0.66	200
	Separate	PCA	82.38	82.79	84.02	83.20	83.10	84.02	1.53	1.66	1.78	1.71	1.67	200
		KPCA	77.46	85.25	**88.93**	87.30	84.74	88.93	0.38	0.50	0.70	0.58	0.54	200
	Average		72.46	82.13	84.26	83.77	80.66	84.35	1.20	1.34	1.52	1.40	1.37	
	Best (KPCA, RBF, Separate)		84.02	85.25	88.93	87.30	84.74	88.93	0.38	0.50	0.70	0.58	0.54	
RGB & Gabor	Concatenate	NTH	37.30	17.62	17.62	22.54	23.77	37.30	357.22	370.56	384.53	370.62	370.73	432768
		PCA	38.11	20.90	17.62	22.95	24.90	38.11	32.35	32.48	32.62	32.51	32.49	200
		KPCA	16.80	36.48	43.85	43.44	35.14	43.85	6.23	6.36	6.38	6.40	6.34	200
	Separate	PCA	9.75	28.28	17.62	20.90	19.14	28.28	31.37	31.52	31.56	31.56	31.50	200
		KPCA	56.97	78.69	88.93	88.52	78.28	88.93	5.17	5.25	5.62	5.39	5.36	200
	Average		31.79	36.39	37.13	39.67	36.24	47.29	86.47	89.23	92.14	89.29	89.28	
	Best (KPCA, RBF, Separate)		56.97	78.69	88.93	88.52	78.28	88.93	5.17	5.25	5.62	5.39	5.36	
All	Concatenate	NTH	37.30	17.62	17.62	22.54	23.77	37.30	359.56	375.32	387.10	373.19	373.79	433344
		PCA	38.11	20.90	17.62	22.95	24.90	38.11	32.39	32.46	32.57	32.55	32.49	200
		KPCA	16.80	36.48	43.85	43.44	35.14	43.85	6.31	6.34	6.56	6.61	6.45	200
	Separate	PCA	39.75	28.28	17.62	20.90	26.64	39.75	31.45	31.55	31.59	31.59	31.54	300
		KPCA	78.28	85.25	87.70	87.30	84.63	87.70	5.22	5.30	5.43	5.37	5.33	300
	Average		42.05	37.71	36.88	39.43	39.02	49.34	86.99	90.19	92.65	89.86	89.92	
	Best (KPCA, RBF, Separate)		78.28	85.25	87.70	87.30	84.63	87.70	5.22	5.30	5.43	5.37	5.33	

Table 3.4: Execution Time Of Food Segmentation Based On 80 Frozen Foods.

	Min	Median	Max	Sum	Avg	STD
HIS	89.08 s	92.89 s	97.50 s	7403.90 s	92.55 s	1.66
DTT	2.83 s	3.13 s	3.90 s	256.12 s	3.20 s	0.24
ROF	0.39 s	0.78 s	1.01 s	62.77 s	0.78 s	0.09
Grabcut	2.40 s	14.24 s	41.15 s	1138.97 s	13.48 s	7.63
Other	0.41 s	0.55 s	1.44 s	50.03 s	0.63 s	0.22
Total	95.51 s	111.59 s	145.00 s	8911.79 s	111.64 s	7.71

Performing $KPCA$ was better than PCA with respect to the dimensionality reduction execution time. Also, the execution time of PCA and KPCA increases with respect to the feature vector size. For example, when all features were used as the input to the Dimensionality Reduction task, the execution time was the highest compared to the single features or two fusion features. Also, when HOG & RGB features were used, the execution time was lowest among two fusion features because the feature size of HOG & RGB feature was the smallest.

The dimensionality reduction, however, was not critical in whole food identification execution time. In the other words, the SVM execution time dominated the execution time for food identification. When we looked at any execution time in which Dimensionality Reduction is NTH compared to PCA and $KPCA$, the execution time was much more. Therefore, Dimensionality Reduction helps to not only improve accuracy but also to reduce the execution time.

The MT-Diet mobile application requires the user to take two pictures of the food plate: a) color and b) thermal. Upon clicking these two pictures, three operations are required to be performed: i) transferring of image data to the cloud server (which has Intel i7 processor), ii) computation of food segmentation, and iii) computation of food identification. The most computationally expensive operation in MT-Diet is the food segmentation operation taking almost $100s$. The data transfer takes $5s$ while the food identification method takes around $5s$ as shown in Table 3.3. Hence, the

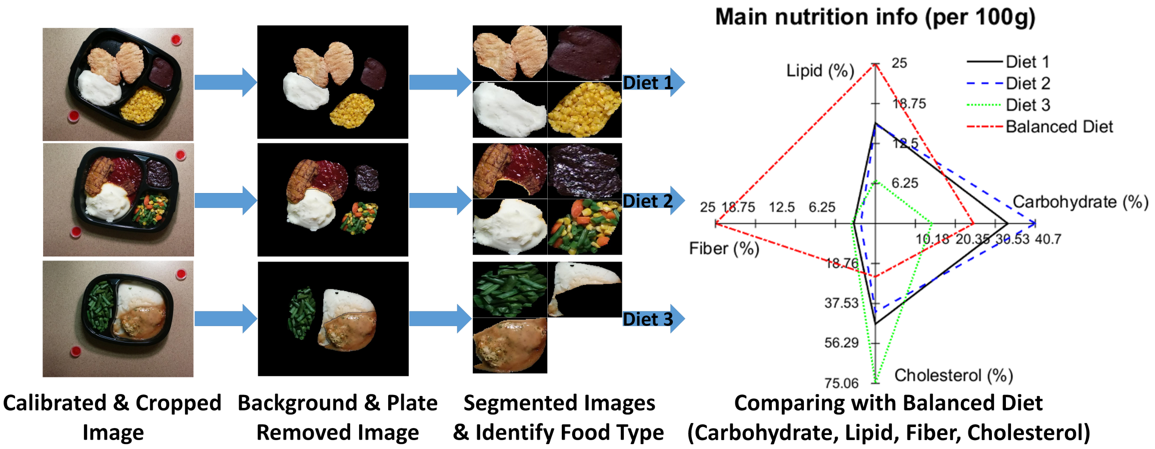


Figure 3.18: Segmentation, Identification, And Diet Recommendation.

overall response time of the MT-Diet application or the time between a user taking a picture and getting back the food type information is 110s.

3.7 Discussion and Future Works

3.7.1 Usability of MT-Diet for diet monitoring

MT-Diet requires images of food plate in the thermal and visual spectrum. Smart watches in the pervasive computing domain already interface cameras in wristbands. Thermal cameras can be interfaced easily through micro-usb ports as shown in Fig. 3.2 or even embedded in the hardware. Image capture can be invoked by recognizing sequence of hand gestures.

Once the images are captured, the smartphone can be used as a hub for data communication and computation. The smartphone may choose to implement food segmentation and identification or may choose to offload the implementation to the cloud server. To implement in the cloud server, the smartphone has to send both the thermal and visual image to the server. On the other hand, for offline implementation in the smartphone, it has to download a learned classification machine, trained using a sample database.

The result of the identification procedure is a set of labeled image areas each corresponding to a specific food item on the plate. The output can be used for several purposes including calorie intake estimation, balanced diet evaluation, or checking conformity to a specific type of diet. One of the several possible outputs is discussed next.

3.7.2 Balanced diet recommendation

The output of the food identification process is a set of image areas with identified food items. We assume that the plate has uniform depth and hence a ratio of the surface area multiplied by the density of the food items (obtained from USDA website) gives the ratio of weight of different food items. We then normalize the amount of each food item on the plate by considering the total food weight to be 100 grams. For each food item, we derive the amount of carbohydrates, lipids, fibers, and cholesterol content for the normalized weight using the statistics per 100 gram of each food item from USDA. For a balanced diet the carbohydrates, lipids, fibers, and cholesterol must have equal weights in the food plate. A balanced diet should have each component at 25%. Given a food plate MT-Diet shows how far it is from a balanced diet in the form of a spider chart (Fig. 3.18) and what component of the diet should be changed in order to make it a balanced diet. These assumption although intuitive need to be experimentally verified in the future.

3.7.3 Limitations

In this section the limitations will be described with respect to: a) usage and b) approach.

In practical usage, MT-Diet depends on the assumption that there is a discernible temperature difference between the food and non-food portions in the images captured. In most cases this is a reasonable assumption, however, if MT-Diet were to be used with non-hot food items then no performance guarantees can be provided. Another assumption made is that the plate is not overflowing with food. This assumption was necessary since overflowing food interferes with the ability to capture the surrounding portions of the plate which is required for automatic segmentation.

The approach limitation is regarding food segmentation which will be discussed in terms of three factors: i) color information, ii) thermal information, and iii) fusion.

Color: The contour detection method (gPb-owt-ucm [59]) may not identify the boundary between plate and background if the factors for deciding the boundary such as color and texture are very similar. Since the presented registration utilizes the plate location for X-axis translation, the successful contour detection between plate and background has to be guaranteed.

Thermal: MT-Diet improves the performance of cooked food item recognition using thermal information. The underlying assumption is that the food temperature is higher as compared to the temperature of plate and background. However, if the food temperature is the similar to the plate or background, it can result in low performance. Since only cooked foods were considered for our experiments, this possibility was not explored.

Fusion: We present a fusion method for thermal and color to improve the performance of food segmentation by exploring four possible cases for ROF. We assumed that each of these cases occurs independently. In other words, if two cases occur simultaneously, the successful performance for the food segmentation cannot be guaranteed.

Chapter 4

EATING ACTIVITY RECOGNITION

Dietary intervention in conjunction with physical activity and socio-economic environment changes is considered to be one of the significant factors in prevention of such chronic diseases, especially problems related to overweight and obesity [28]. Dietary interventions concentrate on two important aspects: a) the type and amount of food intake, and b) eating behavior, i.e., time, frequency, and speed of eating [28]. There have been significant efforts in designing interventions that consider various aspects of nutritional intake such as number and type of calories. However, such interventions often have financial impacts, especially on low-income populations [29, 30]. On the other hand, interventions that consider speed, time, duration and frequency of eating are behavioral in nature and are thus more affordable.

The presence of various chronic diseases such as type 2 diabetes, eating disorders such as binge eating or night eating syndrome, or other metabolic syndromes has been shown to be associated with the speed, duration, frequency and times of meals [69]. Faster speed of eating by itself has been identified as a cardiovascular risk factor. Individuals with type 2 diabetes who engaged in faster eating were found to have higher glycated hemoglobin levels (HbA1c) [70] and weight gain [71]. Moreover, interventions that attempted to reduce speed of eating resulted in improvement of postprandial hormonal responses in adults and adolescents [72].

The administration of such interventions can be facilitated by continuous monitoring of eating actions. Eating is a complex action consisting of three different operations: a) picking food using a utensil or free hand, b) movement from picking to eating, and c) eating that is frequently interleaved by other unrelated hand actions

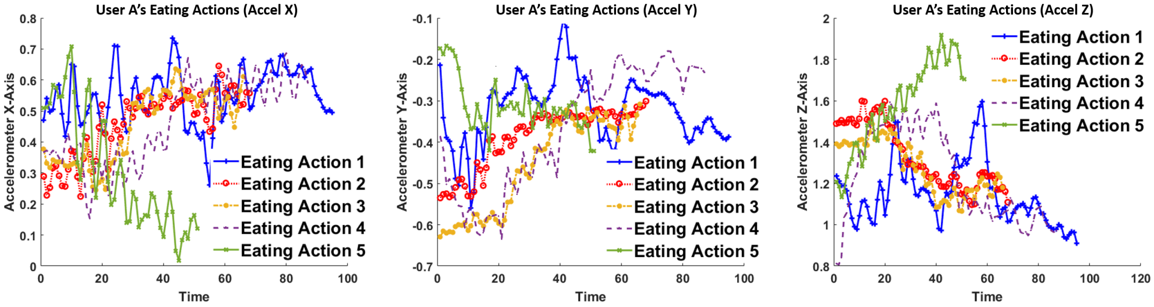


Figure 4.1: Accelerometer Data From A User For Multiple Eating Actions Show No Trend To Identify Pauses Or Arm Movement From Plate To Mouth.

such as gestures while speaking, resting the arm while chewing and other random movements. This makes recognition of eating actions very challenging.

Recently, researchers have used various devices such as wristbands, smartwatches, finger movement sensors, ear-based sensors, glasses or cameras to automatically detect eating action [73, 74, 75, 76, 77]. Due to privacy or environmental constraints, the usage of cameras is often not feasible in many scenarios thus we focus on the challenges associated with using non-visual sensors which typically include Inertial Movement Unit (IMU) and Electromyography (EMG).

There have been several recent works that use non-visual sensors to detect eating actions. The following characteristics of such works make them unusable in a practical scenario: i) **usage of customized sensors** such as finger motion detectors, wearable cameras, data gloves or multiple accelerometers typically **reduces the usability and adherence**, ii) to the best of our knowledge, all current techniques use some form of machine learning that requires an **initial training phase** where the user has to manually provide labeled data during an eating episode. In addition, the data collection, manual labeling and training may have to be repeated to account for variations in the utensils, food items and/or plates. This requirement of manual input from every user makes the system **not User-Independent** and thus cannot be used in a plug-and-play manner. In addition, providing manual input is typically annoying to the user and causes low adherence, iii) Instant feedback is critical to bring about behavioral changes to eating patterns, however, most research works **do not perform instant detection of eating action**. Most recent works with accelerometer data propose detection after complete data collection [74, 73, 75], and the works which do propose instant eating action detection use other sensors such as video, data gloves, or finger movement sensors [78, 79] which are not easy to utilize.

Even though accelerometers were found to be the most widely used non-visual sensor for eating action identification they do not show very high performance. To understand the challenges that exist for the usage of accelerometers for eating action identification, we performed a randomized controlled study (IRB 1: STUDY00004571). For this study, we collected wristband IMU data from 10 individuals (2 females and 8 males between 20 and 35 years of age) during a controlled eating episode. The data was collected from food that was served in a circular plate with three sections from Panda Express. Ground truths were determined by referring to video-recordings of the entire eating episodes. Individuals were asked to use either a spoon or a fork to pick food from each section on the plate. Each session included at least 30 (15 with spoon and 15 with fork) different eating actions for every individual. Fig. 4.1, shows the accelerometer data for all eating actions using a spoon from the same section in the plate. The experimental results point to the following challenges for using accelerometer for eating action identification:

1. Accelerometer signatures are not unique for eating actions: Accelerometer sensors in a wristband give movement data of the arm. However, it only gives acceleration and does not give any information regarding the direction of the arm movement. For example, during an eating episode, when an individual puts food in their mouth, the arm is stationary but there can still be acceleration because the person may have already started to move the arm away from the mouth. Further,

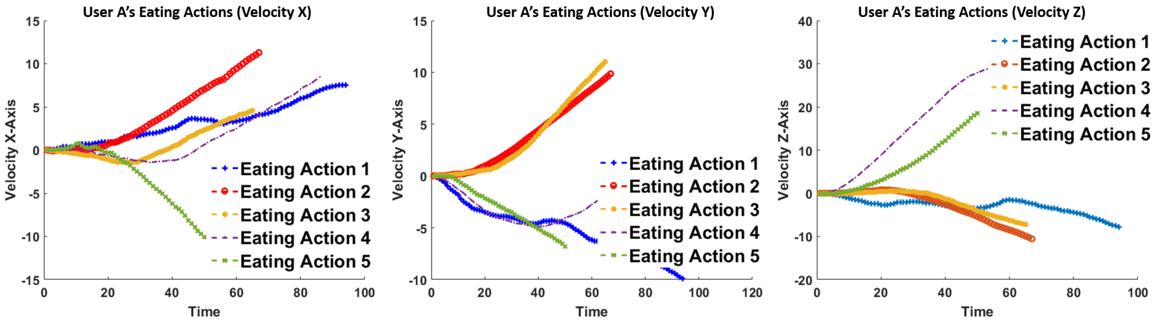


Figure 4.2: Velocity derived from accelerometer with pick points as zero velocity shows no consistent pattern.

when an arm is moving towards the mouth, the acceleration may not be in the positive z direction. Rather, a smooth arm movement from the food plate to the mouth will result in an acceleration pattern that is first positive in the z direction and then negative in the z direction. However, such a signature is common for other actions such as pointing towards the ceiling. Further, such signatures are often different for individuals as evidenced in Fig. 4.1, and 4.2. Thus, any technique that relies solely on accelerometer signatures requires User-Dependent training data.

2. Absence of initial conditions makes it impossible to derive velocity and locations: The start and end position of the arm also cannot be extracted from the wristband accelerometer data. This is because to derive positions the velocity and subsequently distance has to be computed by integrating the accelerometer data. However, such computations are not feasible because there is no way to know when the arm is stationary. As an evidence to this claim, in our data, we have seen there can be acceleration in Z-axis even if the arm is stationary.

3. Lack of consistency in User-Independent tests: Problem of eating action identification is a specific example of larger problem that is an action understanding. Recent works on action understanding can be classified into two broad categories: **i) User-Dependent**, where training data is collected from every individual, and **ii) User-Independent**, where training data is only collected from a subset of users, called **donors**, that does not include the test users. Accelerometer signals lack visually discernible patterns and hence identifying types of features that are fundamental to understanding of eating actions is difficult. Based on our performance metrics, we observed that User-Dependent accelerometer models are far more consistent w.r.t accuracy than User-Independent ones. However, the User-Dependent model needs calibration which requires significant input from the user.

4.1 Overview

In this dissertation, we present **Instant Detection of Eating Action (IDEA)** [1] [**80**] that can operate with a commodity wristband sensor to instantly identify eating actions without any manual input from the user. During testing, IDEA uses a novel unsupervised data processing technique that automatically identifies the very discernible actions within the test data and adds them to training data. These actions, called 'Definite segments', are the ones that can be correctly recognized with very high confidence. Thus, IDEA generates a **User-Specific** model where training data comes from two different sources: i) User-Independent train data and ii) 'Definite segments' which serve as User-Dependent data but are collected during real-time testing. Therefore, IDEA builds a potential User-Dependent model in a plug-n-play manner and provides eating speed feedback instantly as seen in Fig. 4.3.

The core hypothesis of IDEA is that despite variations in arm movements, locations of the mouth, food plate, type of food, and utensils used, an eating action, which consists of the arm movement to lift food up from table to mouth, is assumed to be fairly common across all individuals. In addition, the eating action of an individual test user will be more similar to a subset of other users than to the entire population. To exploit this, IDEA uses a Two-tier Hierarchical Action Detection (**THAD**): **a)**

[1] Accepted paper as UMAP2018 Extended Abstract.

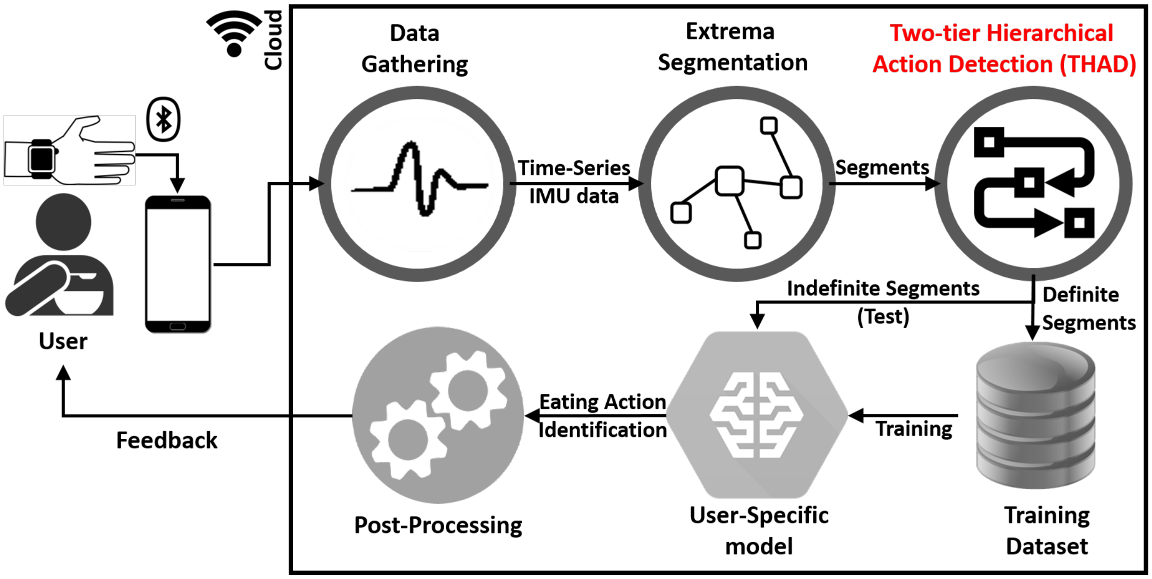


Figure 4.3: IDEA overview.

TIER-1: Generalized Model based Detection where data from an individual A is compared with the data from the set of donors to derive strong and weak candidates for eating action of A. This is done by Deep Neural Networks (DNN) that use Dynamic Time Warping (DTW) [81] based feature matrices. The output of this stage is twofold: i) a set of confirmed eating actions and a set of unconfirmed actions, and ii) a set of users who have "similar eating pattern" to the given user′s eating pattern. **b) TIER-2: Personalized Model based Identification** where the eating actions obtained from the set of users who have "similar eating pattern" given by TIER 1 is used as training set to classify the unconfirmed actions sets as either 'Indefinite' or 'Definite' segments. Then the 'Definite segments' are added to the training set, after which IDEA is able to automatically builds a User-Specific machine learning model without any user-intervention.

THAD is inspired from collaborative filtering techniques applied in recommendation systems. The central assumption in user-based collaborative filtering is that if two users A and B agree on the ratings of a given set of items P, then for an item

Y, unrated by A, A is more likely to agree with the rating of B. This assumption can then be used to decide the recommendation of the item Y to A. For this decision, k-nearest neighbor algorithm (k-NN) is commonly used to identify the users who have similar patterns as those of a test user. The input data is a 2-Dimensional rating data matrix whose X-axis is the item name and Y-axis is the user ID. However, we cannot apply k-NN or other existing techniques directly to our system because the nature of input data is different. For instance, most wristband dataset would have to be represented by a 4-Dimensional matrix consisting of user ID, sensor ID, time, and action types. Although THAD is conceptually the same with user-based collaborative filtering, the differences in nature of data makes it difficult to utilize existing techniques. Therefore, we present a signal processing based novel technique (THAD) which maintains collaborative filtering concepts but works on wearable sensor dataset. To the best of our knowledge, this is the first time this concept has been tried for eating action identification.

Summary of results: We tested the performance of IDEA on 36 subjects. Each subject participated as a volunteer for an eating episode that lasted for at least 15 mins with an average of 30 eating actions. With a training set of eight donors and a test set of 28, the precision of eating action identification was 0.93 while the recall was 0.89. For the worst-case users on an average, IDEA improves precision by 0.11 and recall by 0.15 with when compared to the best state-of-the-art technique as seen in Fig. 4.32. In addition, THAD can also be used for automated labeling of eating action since the mislabeling rate for THAD is 11 out of nearly 10,000 eating or non-eating actions while that for humans is 18.

4.1.1 Intuition

Conventional machine learning models when used for eating detection have significant variance as described in Sec. 4.6.3 and 4.6.4. This happens due to the presence of variables during an eating episode that are not related to the core eating action. To better understand the issues caused by these variables and how IDEA tackles them, we categorize them as follows:

(a) external environment: The pattern of eating changes when external environmental variables such as the food type, utensil used, or the portion of the plate varies, and it is impossible to control these conditions in the real world. For example, the eating activity for soup using the spoon is dissimilar to the eating activity for a steak using a fork and a knife. Thus, changes in external environment causes many different types of eating activities. Despite these differences, we found that all eating activities have three common components: 1) Picking up food 2) Carrying food, and 3) putting the food in the mouth as seen in Fig. 4.4. However, we used only the middle component (carrying food to mouth) as the key eating action in this dissertation. This was done to overcome the issue of inconsistent external environment because the middle component is usually more regular as compared to other two components.

(b) discrete time-series sensor data purity: After defining the key components of eating action, conventional machine learning models was used to identify the middle component as seen in Fig. 4.18. Most models showed reasonable average F1-scores but still had high variance. Ideally, using only the middle component, carrying food to mouth, should help mitigate some of these irregularities, however in practice, we found that this is not the case. We hypothesize that imperfections in the segmentation process of the time-series sensor data and differences in speeds of eating for the various eating actions causes the high variance.

Automatic segmentation leaves snippets of data before and after a desired segment which results in imperfect segments as seen in Fig. 4.6. The data contained in these segments may also contain noise due to the discrete nature of sensors. Moreover, the sensors used were configured for a fixed sampling rate across eating episodes and across users. This can lead to differences in the resulting segmented data, as the speed of eating varies greatly between users and even between various eating actions of the same user which can cause under-sampled or noisy segments. To solve these problems that are specific to eating action detection, we applied scale-space theory to the input data as well as extracted Difference of Gaussian (DoG) features. The issue of noise present in the segmented data is solved to some extent because the scale-space representations are smoother than the raw time series data and can thus contribute to better segments. Using scale-space representations also helps mitigate issues caused by differences in eating speeds since the various octaves represent different sampling rates. In addition to this, DoG provides latent features which are beneficial for eating action detection. However, most models applied scale-space & DoG did not show better performance compared to models with raw data as seen in Fig. 4.19 and 4.20 because scale-space & DoG greatly inflates the feature size which can lead to a potential curse of dimensionality. We were able to avoid this issue by using a Deep Neural Network (DNN). DNN models extract optimal features automatically, so the overall performance is not as adversely affected by these extra features when compared to other machine learning models.

(c) personal innate patterns: For an eating action detection technique to work out-of-the-box and in real-world scenarios, it should have reasonable User-Independent performance. We found that using conventional machine learning models for User-Independent eating action detection did not give us reasonable performance as seen in Fig. 4.21. Even the usage of a DNN with scale-space & DoG features could not give acceptable performance in User-Independent scenarios. We hypothesized that this is because of the huge variations in eating patterns across various users. To test this, we did an experiment to see how number of donors affects the worst-case performances for user independent scenarios. We found that, the performance of the model does not increase by increasing the number of donors and counter-intuitively sometimes actually decreases as seen in Fig. 4.24.

Thus, for User-Independent scenarios, it becomes much more important to consider similarities between a test user and the donors. Tier 1 of THAD clusters similar users using DTW based features as a similarity measure. This ensures acceptable out-of-the-box performance. However, we found that (1) any particular user‘s eating actions are most similar to their own eating actions and (2) they are even more similar during the duration of the same meal. Tier 2 of THAD considers this and includes previous eating actions detected with high confidence into the model to create a User-Specific model. This helps to fine-tune the model and decreases both variance and bias with continued usage as seen in Fig. 4.25 and 4.32. It can be seen that the THAD architecture significantly improves performance for the worst-cases while still increasing the overall performance.

4.1.2 Integration with Nutrition Estimation Systems

Recent works [14, 37, 16, 82, 79, 83] that provide nutrition information from a meal generally focus on the entire served food and not on the amount and type of food that was actually consumed. To be able to access nutrition information from the actual food consumed, before and after images of the food plate have been used. However, these techniques require more manual user input which can lead to low adherence. My previous work [84] solves this problem by estimating nutrition information only from the food that was actually consumed by tracking eating actions and inferring portion

Table 4.1: Existing Works For Eating Identification. UD Is User-Dependent, UI Is User-Independent, And US Is User-Specific. Performance Metric Are F1-score Or Accuracy. Accuracy Unit Is %.

Works	Wearable	Sensor			Model		Performance		
	Location	Origin	Type	Channel	Method	Class	Subject	UD	UI
Smart-Table [85]	N/A	Customized	FSR	130	AdaBoost	8	5	0.92	0.76
Sensing-Fork [86, 87]			IMU	6	Threshold	4	6	95.4%	N/A
Cadavid et al. [88]		Commercial	Camera	1	SVM	5	37	N/A	91.8%
eButton [83]	Chest	Customized	Multi	11	SVM	11	N/A		
GlasSense [76]	Eye		Load-Cell	4	SVM	6	10	0.89	N/A
EarBit [77]	Ear		IMU	9	RF	6	10	N/A	0.8
Liu et al. [79]			Audio	1	ELM	4	6	0.72	N/A
BodyBeat [89]	Neck		Audio	1	LDC	9	14	0.74	0.57
Slowee [78]			EMG, Piezo	2	Threshold	2	10	93.8%	N/A
Olubanjo et al. [90]		Commercial	Audio	1	Threshold	9	6	0.76	N/A
Thomaz et al. [91]	Wrist		IMU	3	RF	10	27	N/A	0.76
Sen et al. [92, 93]			IMU	6	RF	2	21	N/A	98.2%
IDEA			IMU, EMG	18	Proposed	2	36	US: 0.93	

of the plate for each eating action in real-time. However, the eating action detection utilized by this work was not user-independent and thus, required training for each user. IDEA can be directly integrated with such works to provide fine-grained eating action information with high usability. Thus, it will be very beneficial to incorporate IDEA into existing nutrition estimation systems to increase their performance in terms of actual intake in a User-Independent plug-n-play manner.

4.2 Related Work

a) Eating action detection with a single wearable wristband - In the state-of-the-art works, researchers used two different sensors to detect eating actions: wearable and external. The external sensor-based works such as Smart-Table [85] embedded 130 force sensitive resistors under the table, Sensing Fork [86] embedded IMU sensor in the utensil, and Cadavid et al. [88] used front facial video for chewing detection. In these works, it is difficult to keep the eating action detection pervasive, which results in low adherence. To solve this issue, many researchers proposed works based on wearable sensors. These works can be categorized by the wearable locations: chest, eyes, ear, neck, or wrist. For chest, Sun et al. [83] proposed the badge based wearable sensor, eButton involving camera, accelerometer, gyroscope, audio, and proximity sensors. For eyes, GlasSense [76] proposed the glass embedded with load-cell sensors to measure the temporalis muscle for the chewing action detection. For ear, Liu et al. [79] and EarBit [77] proposed the use of headphones embedded with microphone or IMU sensor. BodyBeat [89], Slowee [78], and Olubanjo et al. [90] proposed the neck-wearable based eating action detection system. These works need extra hardware dedicated for eating action monitoring and have limitations such as inability to distinguish between gulping saliva, talking, and eating, or poor performance in noisy environments. The Increase in popularity of wristbands such as Fitbit, however, makes a wristband-based diet monitoring solution desirable. According to our survey (IRB 1) discussed in Sec. 4.4, the most favorable wearable location is wrist. IDEA does not need any dedicated hardware and can provide reasonable performance using any commercially available wristband or smartwatch.

b) IDEA can accurately detect distracted eating patterns - As discussed in Sec. 4.1.1, identification of eating actions within an eating episode which may contain other unrelated actions is a much more difficult problem than the identification of eating episodes during daily life. Users can be involved in other activities like talking, swallowing saliva, shifting in their seats, picking multiple times before eating, or taking multiple bites of a food item that was picked during any eating episode which comprises distracted eating. Thomaz et al. [91] proposed the smartwatch-based eating activity recognition, but they did not consider the distracted eating situations. Also, most works [83, 76, 89, 78, 90] focused on the eating detection among the daily activities but could not account for distracted eating situations. However, since IDEA identifies the three basic actions: a) picking up food, b) carrying food, and c) putting in mouth, separately, it can correctly detect eating actions in the presence other unrelated actions.

c) Plug-n-play operation for User-Specific modeling - According to Tab. 4.1, User-Dependent models are usually more accurate than User-Independent models because User-Dependent models can account for the personal eating pattern of a particular user. Building a User-Dependent model requires an initial training phase where the test user must provide labeled data related to an eating action. This initiation task is time consuming and often annoying. Moreover, such training must be redone if the food item, plate, and utensil changes. On the other hands, although the User-Independent model does not need to collect the labeled data from test user, the reliability in terms of the accuracy has not been guaranteed, which means that the accuracy gap between test users is huge such as in Smart-Table [85] and BodyBeat [89]. Also, Sen et al. [92, 93] showed almost perfect accuracy with User-Independent model, but their segmentation is manually done, which requires extremely high user intervention for preprocessing. IDEA is plug-n-play and automates the segmentation and training process by first using THAD to detect some high-confidence actions and using these as training data to automatically create a User-Specific model with a better performance than a User-Independent model.

4.3 Problem Statement

In this section, we define the eating action detection problem. The input of IDEA is a set of time-series signals, $q_i(t)$ from user i. IDEA predicts j number of tuples corresponding to j actions, $m_j = (t_s, t_f, a)$ where t_s is the start time , t_f is the finish

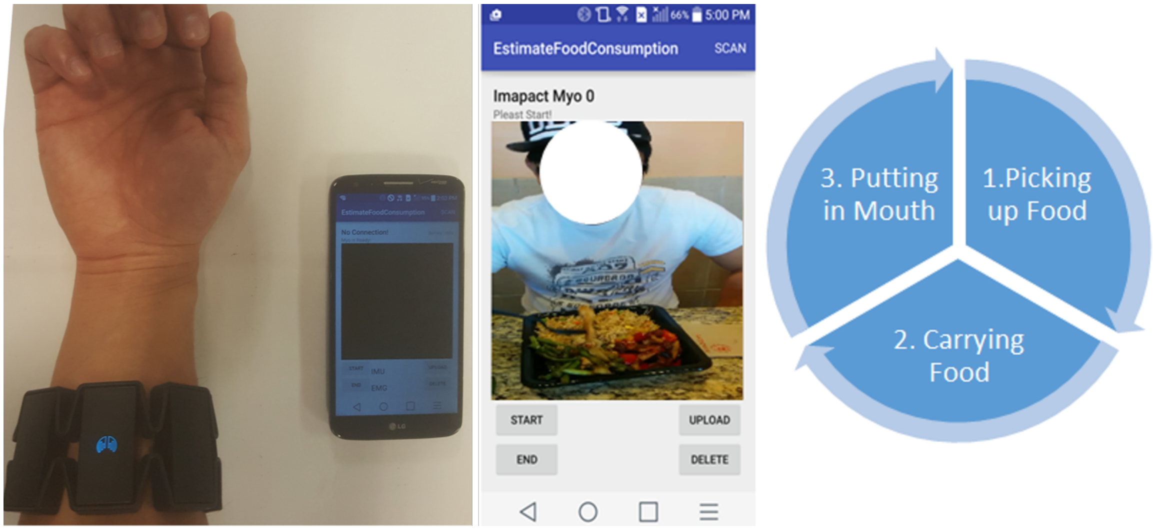


Figure 4.4: Left Is Our System Prototype (LG G2 And Myo Wristband). Center Is Our Android Application For The Data Collection. Right Is Three Components Of Eating Action.

time, and a is the label for the action. The preliminary knowledge are: 1) given a set U_n of n users we know the tuples, $m_j(t_s, t_f, a) \forall j \in U_n$ and 2) user $i \notin U_n$.

4.4 System Architecture

User Study: We take a user-driven approach towards developing IDEA and performed an initial user study to guide our design choices. A diet monitoring system can potentially collect eating activity data from various modalities including wristbands, smartwatches, smartphones, or custom wearables. To determine the most usable alternative for designing the system architecture, we conducted a user study.

In this study after receiving required IRB approval (IRB 1), a total of 102 participants were enrolled with 52 females and 50 males with a median age of 33.3. The participants were initially asked whether they would want to use any mobile application to track their diet. While it was found that the majority of participants (nearly 80%) are favorable towards usage of mobile technology for diet monitoring nearly 19% said they are not at all likely to use a mobile app. Some of the common concerns for not using a mobile app were cumbersome operations, forgetting to use, inaccurate outputs, as well as a general apathy towards diet monitoring.

They were then asked how likely they are to wear a single wearable device that measures their eating behavior. 83% of participants responded favorably to wearing a single wearable device that monitors their eating behavior. When asked whether they would like to wear any item near their head or face or neck, almost 95% of participants responded negatively. A wristband was the choice for nearly 70% of the participants. Amongst rest of the participants, a significantly common suggestion was that if the functionalities realized using a wristband can be done using a smartwatch instead, then they will also use the system even if the smartwatch has to be worn in their dominant hand. When a smartwatch usage is suggested, nearly 91% of participants responded that they see themselves using the system for nearly a month at a stretch.

The conclusions from the user study were: a) customized hardware that are worn on the head or near the face are not preferred, b) a smartwatch is the most preferred wearable for diet monitoring followed by the wristband, and c) no manual intervention is preferable.

System Setup: Fig. 4.4 displays our system architecture. The user wears the Myo wristband which collects accelerometer, orientation, gyroscope data at a rate of 50 Hz. It also measures Electromyogram (EMG) data at 200 Hz. LG G2 (smartphone) is connected to the Myo wristband [94] through Bluetooth to receive the orientation, gyroscope, accelerometer, and EMG data. For each user, we also recorded video data simultaneously using LG G2 camera. The video data is used to build the ground truth.

Myo device provides 18 data streams from four sensors including three data streams for accelerometer, four data streams for orientation, three data streams for gyroscope sensors, and the rest eight data streams from the EMG sensors. The reason behind orientation having four streams is the well-known problem of Gimbal locking which can happen if two axes align in such a way that we lose the degree of freedom in one direction for that moment [95]. The ground truth videos are recorded at 30 frames per second (fps).

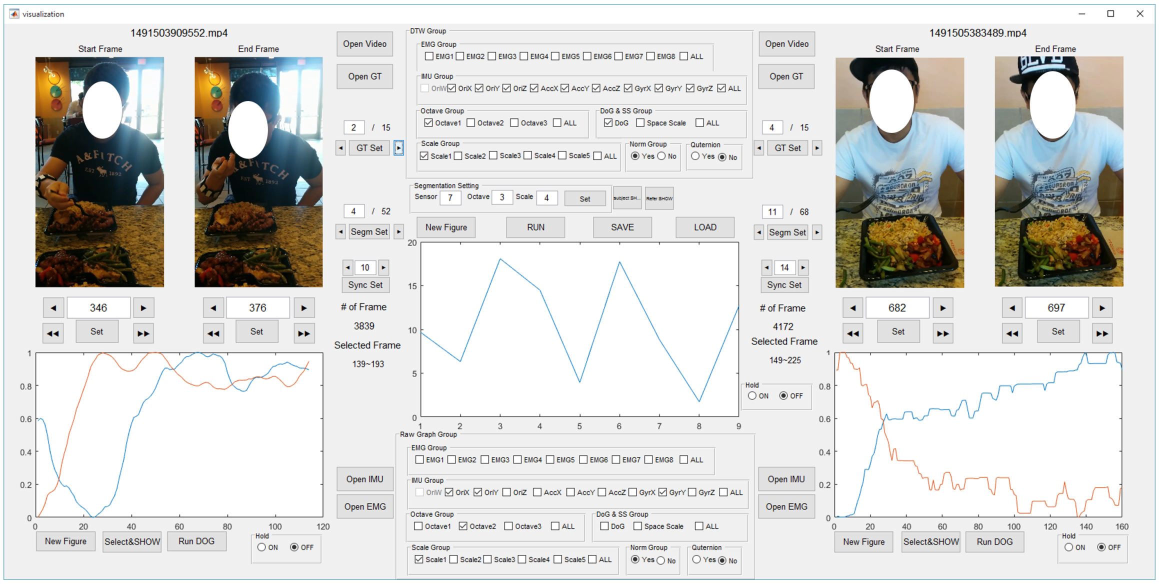
4.5 Methodology

For this work, we assume a prior knowledge about the start times for eating episodes. This is a reasonable assumption since identifying eating among other coarse activities such as running, walking, eating, and sitting is feasible as demonstrated by Thomaz et al [91]. The overall signatures of these activities were found to be clustered in the feature space even if there existed individual variations for each activity type. However, gestures related to eating action, are poly-componential in nature, where multiple gestures are combined in a definite sequence [52]. Thus, in the dissertation, we focus on detecting eating actions from eating episodes in a User-Independent manner, which is a much more difficult problem than determining the presence of an eating episode.

To achieve this, IDEA was designed with five processes as seen in Fig. 4.3: 1) data gathering, 2) extrema segmentation, 3) Two-tier Hierarchical Action Detection (THAD), 4) User-Specific modeling, and 5) interval calculation and feedback. The data gathering is discussed in Sec. 4.5.1 and in this section we will describe other four processes in detail.

4.5.1 Data Collection

Thirty-six subjects were recruited following IRB approvals (IRB 2: STUDY00004155) for data collection. Subjects were between the ages of 20 and 39, 15 of them were female and 21 of them were male. Paired T-test is a method that is applied to deter-



2 / 15

4 / 52

10

# of Frame
3839
Selected Frame
139~193

DTW Group
EMG Group
□ EMG1 □ EMG2 □ EMG3 □ EMG4 □ EMG5 □ EMG6 □ EMG7 □ EMG8 □ ALL
IMU Group
□ OriW ☑ OriX ☑ OriY ☑ OriZ ☑ AccX ☑ AccY ☑ AccZ ☑ GyrX ☑ GyrY ☑ GyrZ ☑ ALL
Octave Group
☑ Octave1 □ Octave2 □ Octave3 □ ALL
DoG & SS Group
☑ DoG □ Space Scale □ ALL
Scale Group
☑ Scale1 □ Scale2 □ Scale3 □ Scale4 □ Scale5 □ ALL
Norm Group
● Yes ○ No
Quternion
○ Yes ● No

Segmentation Setting
Sensor 7 Octave 3 Scale 4

Raw Graph Group
EMG Group
□ EMG1 □ EMG2 □ EMG3 □ EMG4 □ EMG5 □ EMG6 □ EMG7 □ EMG8 □ ALL
IMU Group
□ OriW ☑ OriX ☑ OriY □ OriZ □ AccX □ AccY □ AccZ □ GyrX ☑ GyrY □ GyrZ □ ALL
Octave Group
□ Octave1 ☑ Octave2 □ Octave3 □ ALL
DoG & SS Group
□ DoG □ Space Scale □ ALL
Scale Group
☑ Scale1 □ Scale2 □ Scale3 □ Scale4 □ Scale5 □ ALL
Norm Group
● Yes ○ No
Quternion
○ Yes ● No

4 / 15

11 / 68

14

# of Frame
4172
Selected Frame
149~225

Hold
○ ON ● OFF

Hold
○ ON ● OFF

Hold
○ ON ● OFF

Figure 4.5: Graphic User Interface-based Video Annotation Application For The Labeling Task With Sensor Data Visualization.

mine the number of subjects requited to validate medical studies. To obtain an effect size of 0.5, α of 0.5, and power of 0.80, which is considered as minimum acceptable parameters thirty-four subjects were required. Our study had thirty-six subjects, which meets this expectation.

Subjects were instructed to wear a Myo wristband and eat a meal as they would normally do and process was video-recorded for ground truth determination. They were instructed to eat either with a spoon or a fork and have at least 20 eating actions during the meal. There were no other restrictions on type or origin of food (homemade or restaurant), type of plate, duration of meal, or actions they could perform during the meal. At the end of the experiments, there were a total of 1246 eating actions with an average of almost 35 eating actions per meal per subject.

Labeling: Although the same Android application was used to collect all the data, it was observed that the video and Myo timestamps were sometimes not perfectly synchronized due to underlying Operating System lags. To mitigate this, every subject was asked to make a sharp jerk movement with the arm where the Myo was worn. This movement is easily discernible in video as well as accelerometer graphs, so it could be used for synchronization. Note that the jerk is only needed for the purpose of synchronization and is not required in a practical deployment. Also, the default frequency of data collected using the Myo device varies slightly from the mean frequency and such variation is different for different sensors. Inertial Measurement Unit (IMU) sensors such as accelerometer, gyroscope, and orientation have a default frequency of 50Hz while that of the EMG sensor was 200Hz. The EMG data from the Myo is re-sampled to the default frequency of IMU sensor (50Hz). After completing these two tasks, we annotated the meal video manually through visual inspection.

The ground-truth annotations were performed across four sessions spread over a month. The ground truth labels determined in the first session was verified and corrected in the subsequent sessions. In the second session 18 number of corrections were made. However, there were no further changes that were made during the fourth annotation session. The annotated videos are labeled as one of the following: (1) picking up food, (2) carrying food, and (3) putting in mouth. Fig. 4.4 displays three components for one eating action. Then, the sensor data streams are labeled based on the annotated videos. For the labeling task, we created a MATLAB graphic user interface-based application as seen in Fig. 4.5, which significantly decreases the time spent in labeling.

4.5.2 Extrema Segmentation

Although, food consumption in humans is considered to be one continuous activity by Thomaz et al [91], it can be split into specific and meaningful sub-activities. This step is crucial for the functioning of IDEA and it is one of the reasons that IDEA is able to achieve high performance by identifying relevant sub-activities directly. For segmentation, a sliding-window-based algorithm was utilized which resulted in four

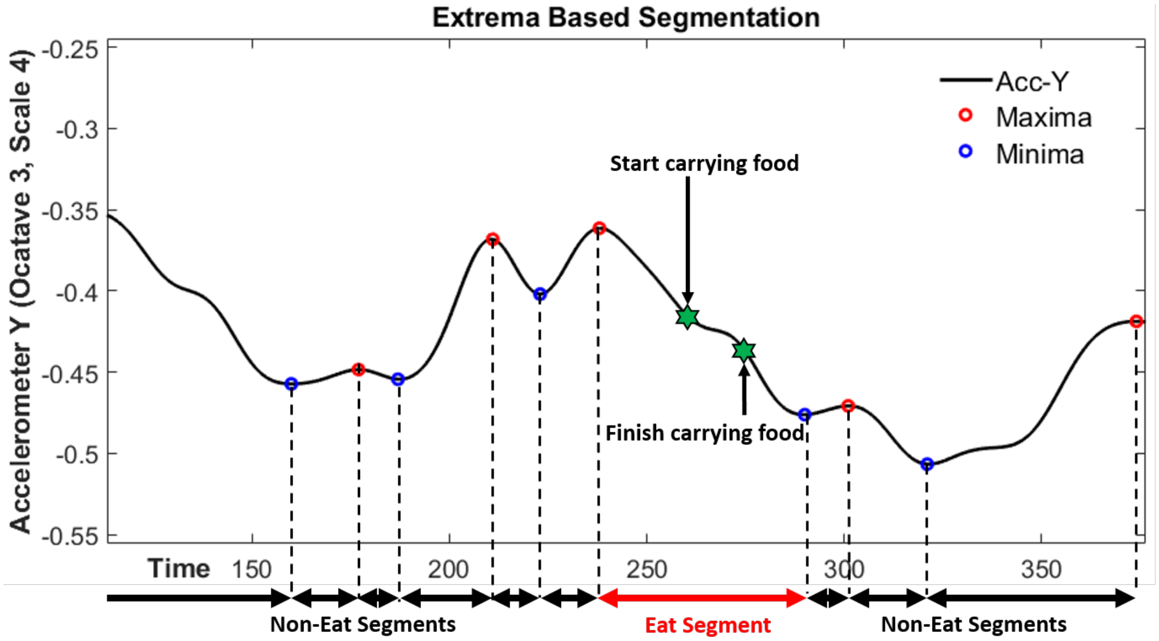


Figure 4.6: Extrema Segmentation.

types of segments: i) segments with only onset of an eating action, ii) segments that fall in the middle of an eating action, iii) segments with the culmination of an eating action, or iv) segments partially spanning multiple eating actions. A valid segment for the purpose of this application is required to have the following properties: **P1:** A segment should have continuous data from the start to the end of only one eating action. **P2:** An eating action should be contained in only one segment (i.e. no overlapping segments). To achieve such properties in the segments, it is required to specify exact time duration of eating actions in the sliding-window-based algorithm. This is not possible in a User-Independent setting, because the average duration of eating action varies significantly from person to person. Hence, an alternative segmentation method based on extrema points was performed.

From our observation of over 1200 eating actions from 36 different users, we conclude that when the user starts and finishes any eating action component, their hand is paused momentarily or there is a sharp change in the orientation of their wrist.

Table 4.2: Extrema Segmentation Average Accuracy By IMU-sensors And Scale-space For All Users. O-# Is Octave Number. S-# Is Scale-space Number. (Unit:%)

Scale-Space		Ori W	Ori X	Ori Y	Ori Z	Acc X	Acc Y	Acc Z	Gyr X	Gyr Y	Gyr Z
O-1	S-1	44.83	67.84	72.16	45.75	0.06	2.6	0	9.31	0	1.12
	S-2	47.73	70.89	73.32	47.91	3.41	31.83	6.42	27.44	0.66	6.7
	S-3	48.65	71.86	73.53	48.95	4.89	41.07	10.09	30.66	1.38	8.69
	S-4	49.56	73.35	74.44	50.48	9.29	50.26	15.64	34.93	2.85	10.42
	S-5	51.4	73.96	75.25	52.44	15.31	56.18	22.16	38.93	3.99	11.84
O-2	S-1	53.36	74.53	75.14	55.78	17.28	57.71	25.42	42.03	6.38	14.31
	S-2	63.14	82.16	82.01	66.03	50.95	76.73	49.28	58.65	26.98	27.76
	S-3	65.08	82.71	84.25	67.83	56	79.45	53.83	60.95	31.79	31.84
	S-4	67.85	84.53	86.28	69.68	61.05	82.98	57.92	63.27	36.54	37.29
	S-5	71.33	85.94	88.1	70.89	66.71	85.62	61.9	65.46	41.88	43.71
O-3	S-1	75.59	87.21	88.89	75.68	67.92	85.79	68.45	72.13	47.11	51.79
	S-2	86.16	92.63	96.05	87.85	91.45	96.34	84.32	73.86	77.87	72.9
	S-3	86.78	94.21	96.64	89.84	94.1	97.55	86.15	74.4	81.76	74.97
	S-4	88.76	94.63	96.21	91.15	95.9	**98.26**	88.73	77.49	85.53	79.63
	S-5	90.33	95.35	97.98	91.53	97.34	98.06	89.91	81.87	87.75	81.91

The most significant arm movement is related to carrying of food item from the plate to mouth, i.e., the second eating action component. Based on this observation, we utilized the extrema to segment the continuous movement of hand gesture into two types of segments: a) second component of eating action and b) all other eating action components or unrelated gestures.

Each extrema point becomes the start and end moment for the second eating action component as seen in Fig. 4.6. However, in order to obtain accurate segmentation, two factors should be considered: a) selection of the appropriate sensor as an input for segmentation and b) data smoothing. In our experiments, we studied the

Table 4.3: Extrema Segmentation Accuracy By EMG-sensors And Scale-space (Unit:%). O-1 Is Octave1, O-2 Is Octave2, And O-3 Is Octave3.

Scale-Space		EMG 1	EMG 2	EMG 3	EMG 4	EMG 5	EMG 6	EMG 7	EMG 8
O-1	Scale 1	0	0	0	0	0	0	0	0
	Scale 2	0	0	0	0	0	0	0	0.07
	Scale 3	0	0	0	0	0	0	0	0.07
	Scale 4	0	0	0	0.07	0	0	0	0.07
	Scale 5	0.06	0	0	0.07	0	0	0	0.07
O-2	Scale 1	0.26	0.06	0.06	0.2	0.45	0.06	0.33	0.2
	Scale 2	5.41	5.58	7.19	7.13	6.44	7.36	6.81	5.94
	Scale 3	10.31	8.47	11.16	9.92	9.94	11.06	10.45	10.55
	Scale 4	14.46	14.02	15.25	14.73	15.25	15.09	16.72	15.49
	Scale 5	21.94	20.3	21.54	21.5	22.18	22.21	24.31	21.81
O-3	Scale 1	27.18	25.67	26.67	26.31	27.31	25.71	27.09	28.79
	Scale 2	74.28	71.22	71.85	74.2	72.05	73.59	72.41	75.15
	Scale 3	79.6	77.47	76.84	80.3	76.96	78.07	77.68	78.09
	Scale 4	83.52	82.36	81.11	83.47	82.73	83.82	82.78	82.53
	Scale 5	86.07	85.61	84.52	86.61	86.79	87.84	87.42	86.39

entire set of 18 sensors to derive the sensor data that best segments eating action on an average for all users (Tab. 4.2, 4.3). Further, the raw data obtained at a high sampling frequency using Myo device has significant noise resulting in large number of potentially insignificant extrema. Hence, the extrema segmentation will result in large number of segments that are small and hence an eating action can span over multiple segments. Smoothing can remove extrema from the raw data and can result in a reduction of number of segments and increase in the average length of the segments. However, too much smoothing will result in segments that are too large and incorporate multiple eating actions. Thus, finding the optimal smoothing parameter is a crucial task for accurate segmentation. Smoothing can be parameterized using the well-known scale-space theory [96], where raw data is convolved with Gaussian

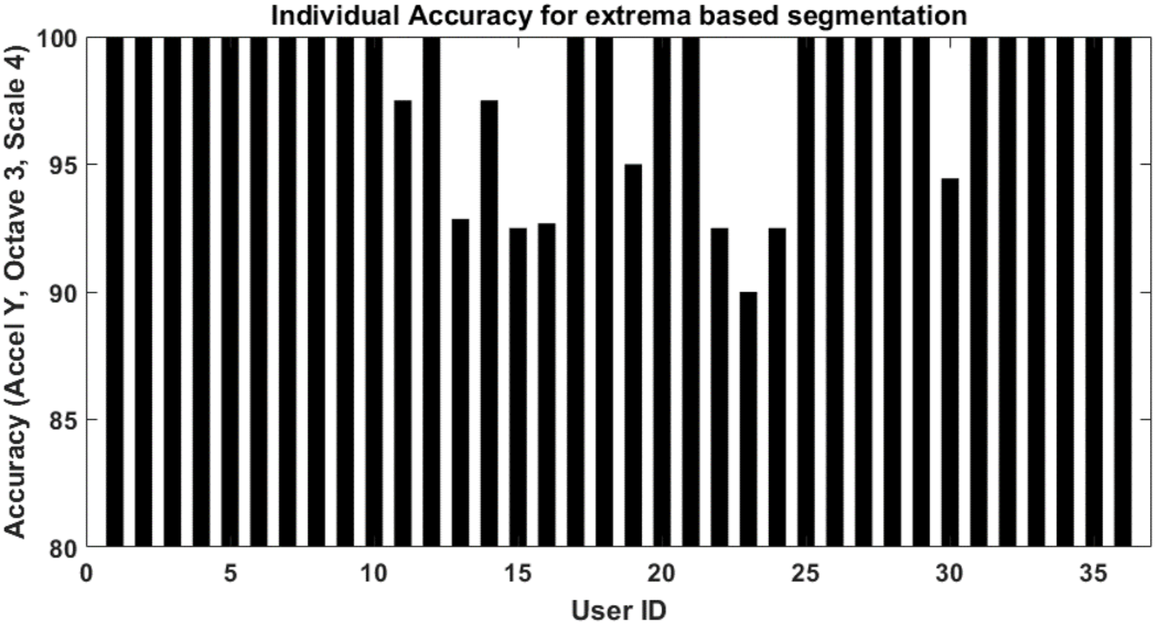


Figure 4.7: Extrema Segmentation Accuracy For Individual User Using Accelerometer Y.

filters with varying standard deviation (known as scale-spaces) and subsequently subsampling the convolution results (also known as octaves). The Difference of Gaussian (DoG) features was utilized to perform segmentation. The usage of scale-space theory is discussed in Sec. 4.5.3 in detail. When the second eating action component i.e., carrying food from plate to mouth, appears in a segment generated from extrema points as seen in Fig. 4.6, the segment was labeled as 'Eat'. If not, the segment was labeled as 'Non-Eat'.

Tab. 4.2, 4.3 show the accuracy of segmentation by sensor and scale-space. For each sensor data, we evaluated three octaves and five scale-spaces. Since the aim of eating activity detection is to identify only 'Eat' segments, we measured the accuracy of our segmentation by computing the percentage of the second eating action component in the ground truth that falls in one segment.

If an 'Eat' segment identified by human eye is not contained in the one segment, we consider it as wrong segment. According to the table, the accelerometer Y axis with third octave and fourth scale has the best average accuracy (98.06%) for all users. Therefore, the accelerometer Y axis with third octave and fourth scale was selected as an input for extrema segmentation. Fig. 4.7 displays the individual accuracy of extrema segmentation. The maximum number of wrong segment in the worst-case individual was 4 out of 40.

The main reason for such a result is that during an eating action, there is a sharp reorientation of the arm to align the food item with the mouth. This is why we see that orientation Y is also a good indicator of segments for eating action. Orientation Y was seen to have a positive correlation with accelerometer Y, hence we could potentially use either one. Intuitively though accelerometer Z should have been the better identifier of eating action segments, given that there is a sharp movement of the hand from the plate to the mouth. But from our results we see that it can only achieve an accuracy of 89%. Often, this is because the users while moving their hand towards their mouth, also move their head towards the food item. This dampens the change in accelerometer Z during an eating action.

4.5.3 Two Hierarchical Action Detection (THAD)

According to the evaluation of User-Independent modeling discussed in Sec. 4.6.4, a possible reason why the User-Independent performance is much poorer than the performance of user dependent is that the DNN is potentially over-fitting to the donors. This may be due to the fact that the amount of training data segments for the DNN model is relatively smaller than what is expected for a machine with the complexity of DNN. To increase DNN accuracy, we not only have to increase the duration of monitoring of the donors or the number of donor but also potentially include labeled training data from the user itself, which we are trying to avoid.

To solve this issue, IDEA uses THAD which differs from the traditional usage of

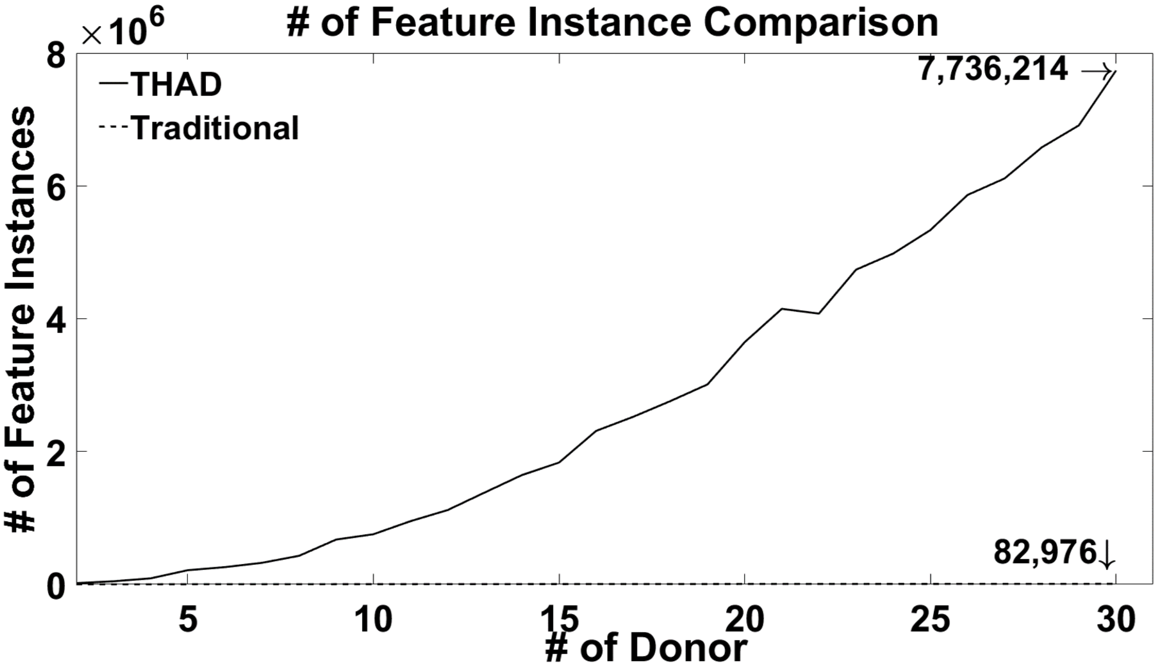


Figure 4.8: The Number Of Feature Instances Comparison Between THAD And Traditional Model.

machine learning systems in two key factors:

K1: Instead of using raw data or scale-space & DoG segment as training samples, THAD uses the comparison between eating actions and non-eating actions of different users as training samples. THAD in fact uses Dynamic Time Warping (DTW) based comparison, which considers similarity between two eating actions of different individuals at various time and magnitude scales. This serves three purposes: a) it increases the number of feature instances by orders of magnitude as shown in Fig. 4.8 and hence reduces the potential for over-fitting by the DNN, b) it allows fine-tune training with the small number of donors, and c) it guides the DNN to derive a reduced set of donors that are most similar to the given test user.

K2: Utilizing the observation that User-Dependent models are far more consistent the accuracy than User-Independent models, THAD uses two-tier hierarchical steps. In the first step, the THAD provides a set of most similar users though clustering, which are then used in the second step to derive a personalized model for the test user.

THAD consists of one step of preprocessing and two steps of a two-tier hierarchical modeling: a) Scale-space & DoG creation (preprocessing), b) generalized model encoding and personalized model extraction, and c) personalized model-based eating action classification. The aim is to create the training set for User-Specific model by defining 'Definite' segments amongst test user's segments generated by extrema segmentation.

Scale space & DoG creation

In this phase, IDEA requires labeled data collection from a training population. Each member of the population is called a donor. Note that this step is one time and does not involve data collection from the user. The data streams from the different sensor modalities are analyzed through space scaling using Gaussian filters and their differences. As seen in Fig. 4.9, a sensor stream $q_i(t)$ is first convolved with a Gaussian signal with zero mean and standard deviation $\sigma = 1$, $G(0, \sigma)$ to form the first scale-space. The second scale-space is formed by convolving, $q_i(t)$ with a zero mean Gaussian signal having standard deviation $\sigma = 2^{\frac{1}{4}}$. The n^{th} scale-space is thus given by, $ss_1(q_i(t), n) = q_i(t) * G(0, 2^{\frac{n-1}{4}})$. The first five scale-spaces for each sensor stream is denoted as the *first octave* also denoted as ss_1.

The fifth scale-space of the first octave is then under-sampled at half the rate, also represented by the function $sub(ss_1, \frac{1}{2})$. Five subsequent scale-spaces of this sub-sampled signal are then computed to derive the second octave. Hence the n^{th} scale-space of the second octave can be represented as, $ss_2(q_i(t), n) = sub(ss_1(q_i(t), 5), 2) * G(0, 2^{\frac{n-1}{4}})$. In this manner five octaves are generated from each sensor stream. Hence, the n^{th} scale-space of the j^{th} octave can be represented as, $ss_j(q_i(t), n) = (sub(ss_{j-1}(q_i(t), 5), 2) * G(0, 2^{\frac{n-1}{4}})$.

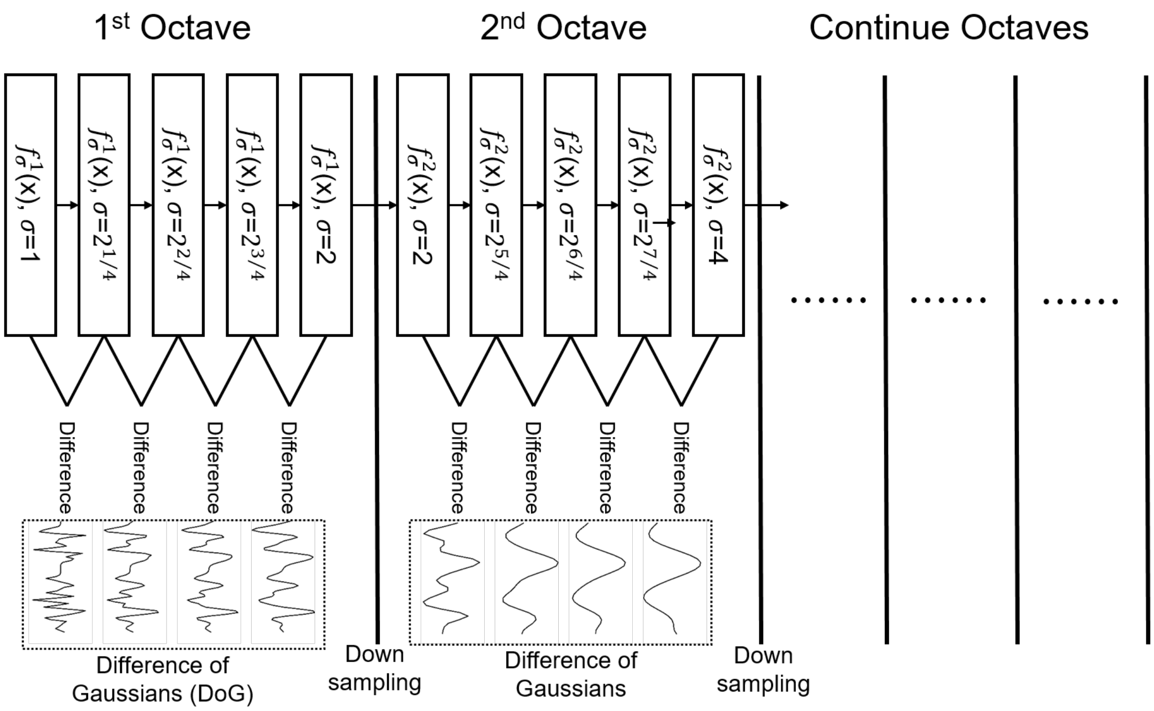


Figure 4.9: Scale-space & DoG Creation.

This operation generates different scale-spaces of the signal, such that each scale-space can capture some hidden patterns in the signal. We then compute the difference in successive scale-spaces also called Difference of Gaussians or DoG signal. For example, the first DoG is computed as, $ss_1(q_i(t), 1) - ss_1(q_i(t), 2)$. This is done for every sensor stream and for every donor. Thus, for each donor we obtain a *feature set*, F_k, that consists of 15 scale-spaces (3 octaves × 5 scale-spaces) of each sensor data stream and 12 DoGs (3 octaves × 4 DoGs), since 3 octaves was considered.

For each sensor data stream, the extrema segmentation algorithm then was applied discussed in Sec. 4.5.2. The output of this step is a set of scale-space & DoG for each sensor data stream labeled as 'Eat' and 'Non-Eat' for the set of donors. Hence, corresponding to each eating or non-eating action, there are 486 data streams (18 sensors × 3 octaves × (5 scale-spaces + 4 DoGs)) and one label.

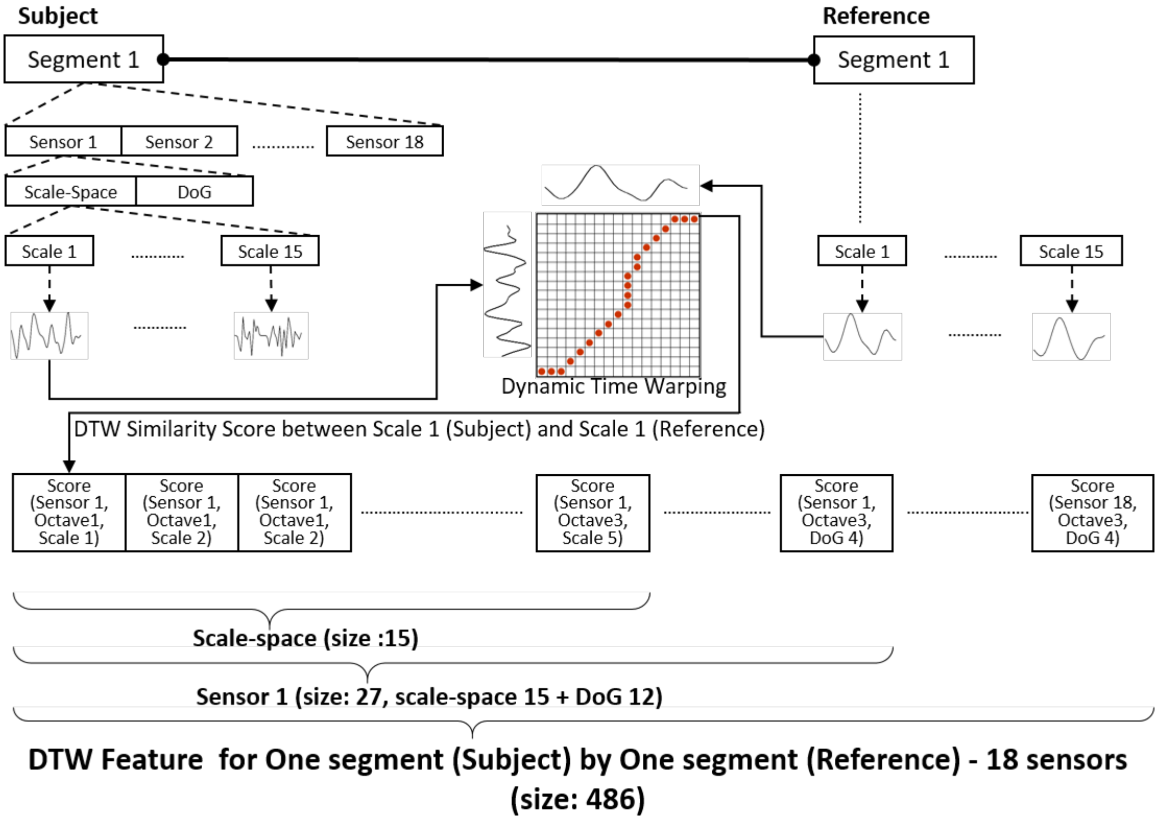

Subject
Segment 1
Reference
Segment 1
Sensor 1
Sensor 2
Sensor 18
Scale-Space
DoG
Scale 1
Scale 15
Scale 1
Scale 15
Dynamic Time Warping
DTW Similarity Score between Scale 1 (Subject) and Scale 1 (Reference)
Score (Sensor 1, Octave1, Scale 1)
Score (Sensor 1, Octave1, Scale 2)
Score (Sensor 1, Octave1, Scale 2)
Score (Sensor 1, Octave3, Scale 5)
Score (Sensor 1, Octave3, DoG 4)
Score (Sensor 18, Octave3, DoG 4)
Scale-space (size :15)
Sensor 1 (size: 27, scale-space 15 + DoG 12)
DTW Feature for One segment (Subject) by One segment (Reference) - 18 sensors (size: 486)

Figure 4.10: DTW Feature Extraction Methodology.

Generalized Model Encoding

As shown in Fig. 4.11, the entire donor population set is first divided into two groups: a) a subject member T, and b) reference subgroup with all other members. For a subject member, the segments corresponding to the second component of eating action labeled as 'Eat' are compared with all segments of the reference subgroups using a distance metric. The matrix obtained as a result of such comparison is termed as *DTW feature matrix* due to the distance metric used in this dissertation. Fig. 4.10 shows the DTW feature matrix extraction methodology. Given two segments (one 'Eat' signal from subject and one 'Eat' signal from a member of reference subgroup), there are 18 sensors signals that represent each segment. For each sensor signal segment, there are 15 scale-spaces and 12 DoGs. The methodology chooses a specific

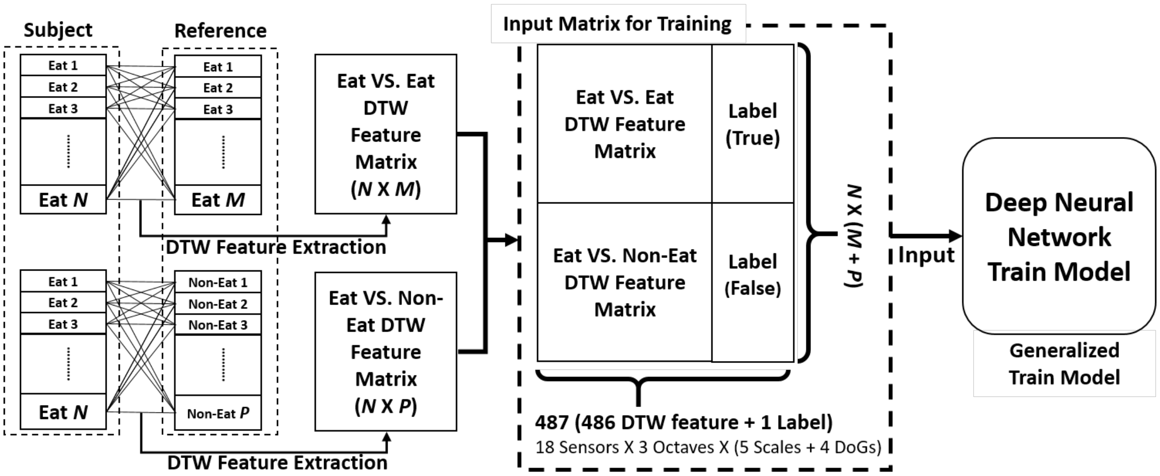

Subject
Reference
Eat 1
Eat 2
Eat 3
Eat N
Eat 1
Eat 2
Eat 3
Eat M
Eat VS. Eat DTW Feature Matrix (N X M)
DTW Feature Extraction
Eat 1
Eat 2
Eat 3
Eat N
Non-Eat 1
Non-Eat 2
Non-Eat 3
Non-Eat P
Eat VS. Non-Eat DTW Feature Matrix (N X P)
DTW Feature Extraction
Input Matrix for Training
Eat VS. Eat DTW Feature Matrix
Label (True)
Eat VS. Non-Eat DTW Feature Matrix
Label (False)
N X (M + P)
487 (486 DTW feature + 1 Label)
18 Sensors X 3 Octaves X (5 Scales + 4 DoGs)
Input
Deep Neural Network Train Model
Generalized Train Model

Figure 4.11: Generalized Model Encoding.

scale-space or DoG of a given sensor for subject segment and compares it with the corresponding scale-space or DoG of the same sensor for the member of the reference subgroup using DTW distance metric. Given two-signal snippets, DTW provides a distance matrix by using all possible alignments of the two snippets. The last value of warping path is called the *DTW score*. The DTW score between the two segments with the specific scale-space or DoG of the chosen sensor becomes an element of the DTW feature matrix. Hence, for each combination of two segments obtained by taking one segment from subject and another from a user in the reference subgroup, there can be a total of 486 DTW scores. This forms one row of the DTW feature matrix. Thus, the DTW feature matrix for a given subject has *number of segments of subject (N)* × *number of segments of all other users in reference subgroup (M + P)* rows and 486 columns as seen in Fig. 4.11.

Since every segment is either 'Eat' or 'Non-Eat', there are two kinds of comparisons between, a) subject member 'Eat' and reference member 'Eat', and b) subject member gesture 'Eat' and reference member 'Non-Eat'. Corresponding to the DTW matrix there is a label matrix M_T for a subject member. The matrix entries of the first type are given the label $True$ while the other types are given the label $False$. This

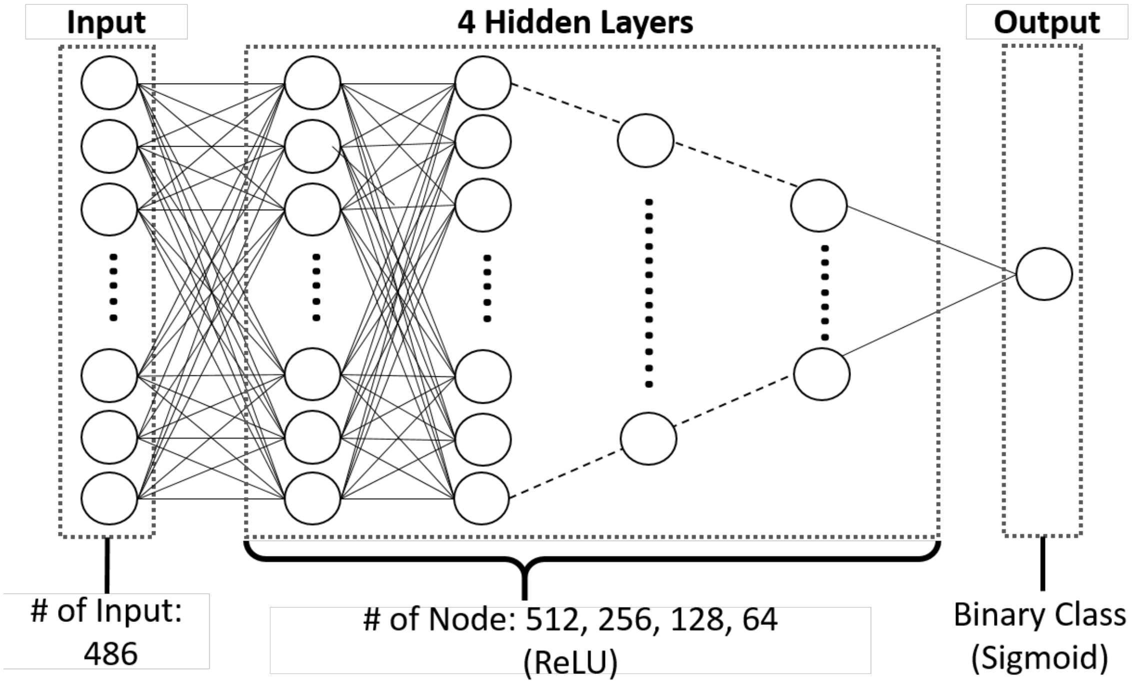


Figure 4.12: DNN Structure For Generalized Model.

operation is done for every possible subject member selected from the group of donors. The resulting set $U_h = \bigcup_{\forall T} D_T$ and the set of matrix labels $M_h = \bigcup_{\forall T} M_T$ are used to train a deep learning network $N(U_h, M_h, h)$ to recognize the second component of eating action. This deep learning model is the generalized model to recognize eating action.

The DNN model used for this step is shown in Fig. 4.12, it has four hidden layers with nodes starting from 512 and exponentially reducing to 64. The activation function is ReLU, the output layer is sigmoid for binary classification, and the gradient descent optimization is ADAptive Moment estimation (ADAM) [97]. Note that the general model will not be used to identify all the instances of eating action. It will only be used to extract example gesture instances so that they can used as training examples for the specific user.

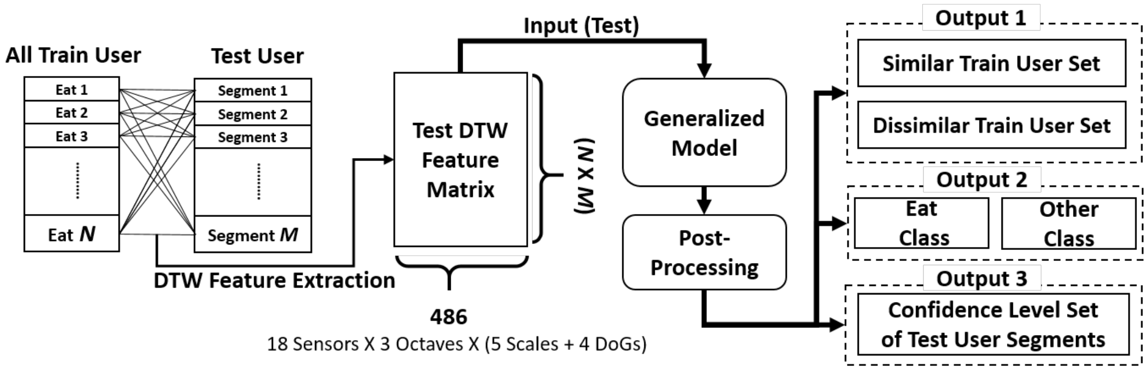


Figure 4.13: Generalized Model Prediction And Outputs.

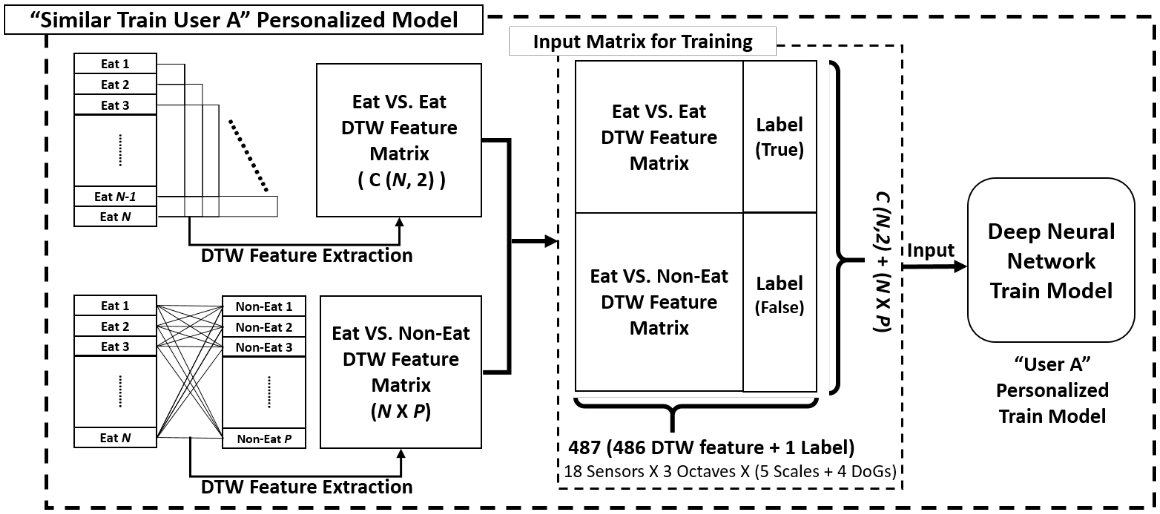

"Similar Train User A" Personalized Model
Eat 1
Eat 2
Eat 3
Eat N-1
Eat N
Eat VS. Eat DTW Feature Matrix (C (N, 2))
DTW Feature Extraction
Eat 1
Eat 2
Eat 3
Eat N
Non-Eat 1
Non-Eat 2
Non-Eat 3
Non-Eat P
Eat VS. Non-Eat DTW Feature Matrix (N X P)
DTW Feature Extraction
Input Matrix for Training
Eat VS. Eat DTW Feature Matrix
Label (True)
Eat VS. Non-Eat DTW Feature Matrix
Label (False)
C (N,2) + (N X P)
487 (486 DTW feature + 1 Label)
18 Sensors X 3 Octaves X (5 Scales + 4 DoGs)
Input
Deep Neural Network Train Model
"User A" Personalized Train Model

Figure 4.14: Personalized Model Encoding For 'User A'.

Personalized model extraction

Given a test user, IDEA first extracts the scale-space & DoG features from all 18 sensor data streams. It then applies the extrema segmentation method discussed in Sec. 4.5.2 to derive potential eating action segments. As shown in Fig. 4.13, IDEA then uses the DTW feature matrix extraction algorithm discussed in Fig. 4.10, to compare every segment of the test user with 'Eat' segments of all donors. The resultant DTW feature matrix is used as test data in the generalized DNN model of Fig. 4.12.

The test execution of the generalized model results in class assignment *True* or

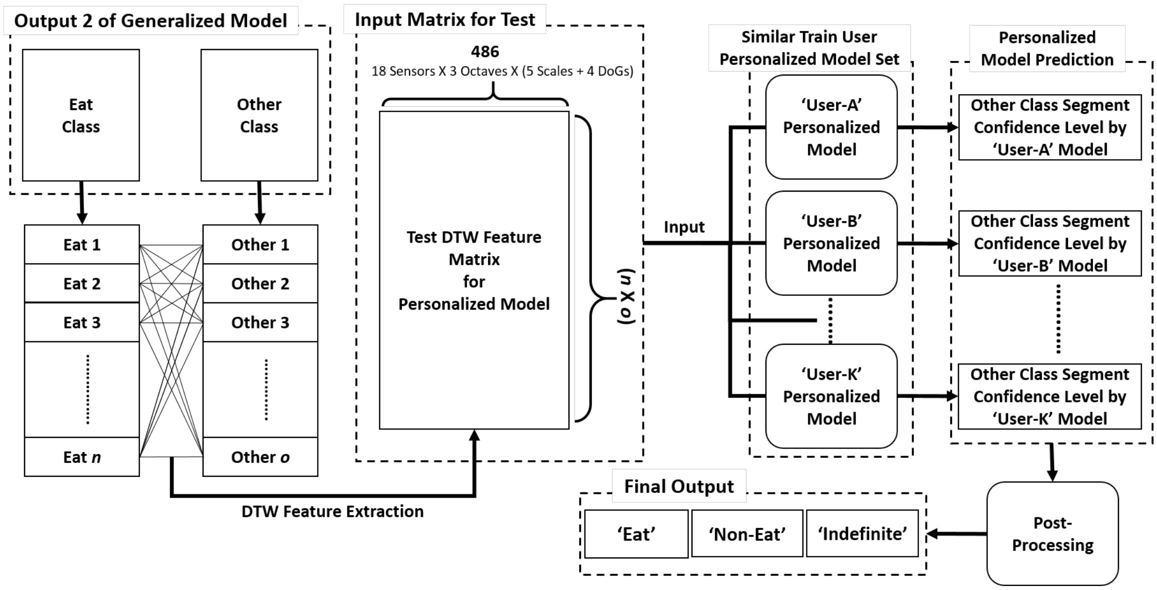

Output 2 of Generalized Model
Eat Class
Other Class
Eat 1
Eat 2
Eat 3
Eat n
Other 1
Other 2
Other 3
Other o
DTW Feature Extraction
Input Matrix for Test
486
18 Sensors X 3 Octaves X (5 Scales + 4 DoGs)
Test DTW Feature Matrix for Personalized Model
(n X o)
Input
Similar Train User Personalized Model Set
'User-A' Personalized Model
'User-B' Personalized Model
'User-K' Personalized Model
Personalized Model Prediction
Other Class Segment Confidence Level by 'User-A' Model
Other Class Segment Confidence Level by 'User-B' Model
Other Class Segment Confidence Level by 'User-K' Model
Post-Processing
Final Output
'Eat'
'Non-Eat'
'Indefinite'

Figure 4.15: Personalized Model Prediction And Outputs.

False for each row in the DTW feature matrix. If a row is labeled as *True*, it means that the corresponding segment of the test user matched with a segment of the corresponding member of the reference subgroup. Hence analyzing the output of the generalized model, **we can identify how many segments of a given member of the reference subgroup matches with a segment of the test user**, in the post processing phase. we can use this information in three ways: a) determine a set of users in the reference subgroup that are **most similar** (users who have the most number of matching 'Eat' segments with the test user′s segments) to the given test user, b) use a threshold such that if a segment has high number of matches with 'Eat' action segments of reference subgroup it can be classified as 'Eat' segment, and c) use the number of matches as a weight feature of the segments to be used in subsequent personalized model-based classification.

In the next step for each user in the *most similar* set, a DNN based personalized model is developed as shown in Fig. 4.14. For each similar user in the donor set, there are two sets of segments: 'Eat' or 'Non-Eat'. IDEA computes the DTW feature matrix for segments that are labeled 'Eat'. The rows of this 'Eat' vs. 'Eat' DTW feature matrix is labeled as $True$. IDEA also computes the DTW feature matrix between a segment labeled 'Eat' and all other segments labeled 'Non-Eat'. The rows of this 'Eat' vs. 'Non-Eat' matrix is labeled as $False$. The combined DTW feature matrix is used to train a User-Specific DNN model for each user in the similar user set. The configuration of this DNN is similar to Fig. 4.12 but each hidden layer has the same number of nodes (256). If a given test user has three similar users in the train set, then it has three personalized DNN models. Note that the personalized DNN model does not use any data generated from the test user for train, it is rather developed based on donors that are similar to the test user.

Personalized Model based Classification

The aim of this step is to classify the 'Other' class as **'Definite'** or **'Indefinite'**. As shown in Fig. 4.15, IDEA first computes the DTW feature matrix between segments of test user labeled as 'Eat' and those labeled in the 'Other' class following the methodology in Fig. 4.10. This feature matrix is then used as test data for each of the personalized models for the similar user set. The output of each User-Dependent DNN model is again a segment weight, which is the number of eating actions of user K that are similar to one segment labeled as 'Other'. Hence for each segment in the 'Other' set, this step provides a vector with the segment weight corresponding to every member in the similar user set. In the post processing step IDEA sets a threshold such that of the similarity score exceeds this threshold for all similar users then the segment is classified as 'Eat', if the segment weight is 0 for all user then it is classified as 'Non-Eat', and if the threshold is not exceeded for at least one similar user it is classified as 'Indefinite'. The segments classified as 'Eat' and 'Non-Eat' will be henceforth referred to as 'Definite'.

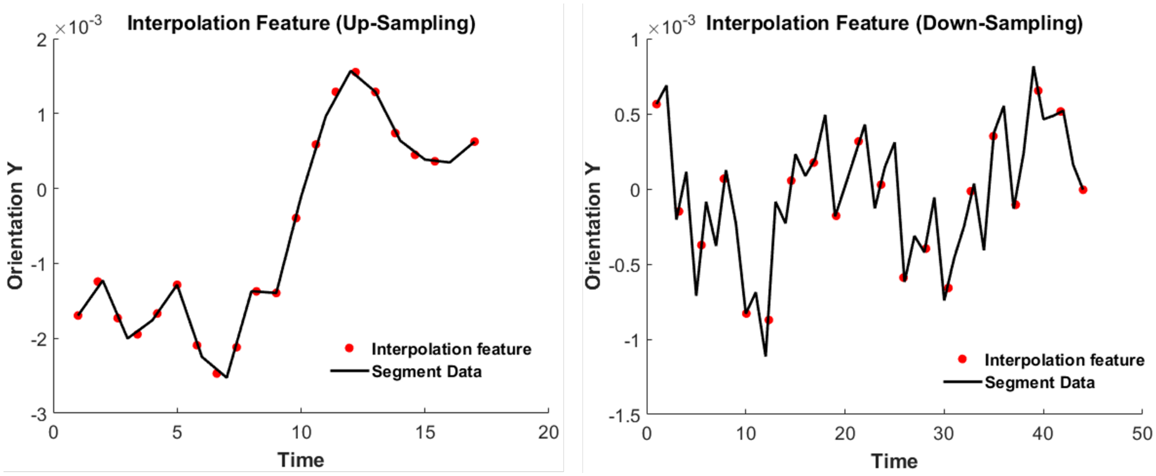
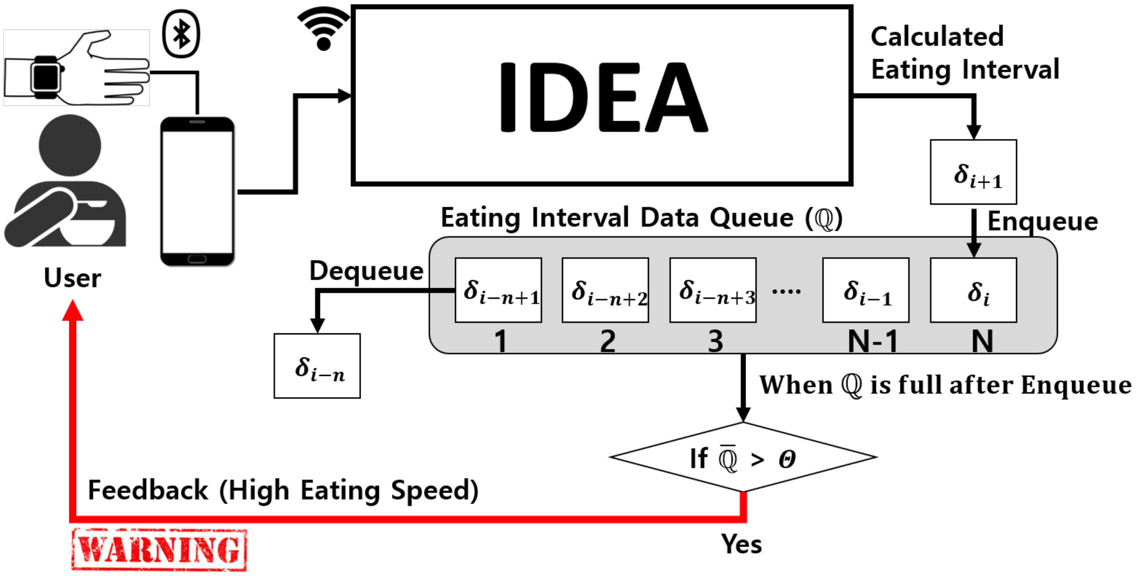


Figure 4.16: Examples For Interpolation Feature Extraction. Left Is Up-sampling And Right Is Down-sampling.

4.5.4 User-Specific Modeling

The aim of User-Specific model is to classify the 'Indefinite' segments as 'Eat' or 'Non-Eat'. To build the User-Specific model, there are three steps: i) training data renewal, ii) feature extraction, and iii) machine learning training. For the training data renewal step, we simply added the 'Definite' segments to the training set of User-Specific model. The training data renewal makes the User-Specific model update from the User-Independent to User-Dependent manner without any user intervention.

For the feature extraction, we used the combination of scale-space & DoG and interpolation. The scale-space & DoG set of each segment is already obtained for THAD discussed in Sec. 4.5.3. We then applied the interpolation to all 486 scale-space & DoG set (18 sensors × 3 octaves × (5 scale-spaces + 4 DoGs)). The aim of interpolation technique is to make the uniform size for each segment. The segment size should be uniform to use it as an input for supervised-learning. However, the extrema segmentation method generates irregular size segments, so we used cubic interpolation to obtain uniform size segments. The segment size for scale-space & DoG set corresponding of the first octave was fixed to 20 samples. Among twenty



IDEA
Calculated Eating Interval
δ_{i+1}
Enqueue
Eating Interval Data Queue ($\mathbb{Q}$)
Dequeue
δ_{i-n+1}
δ_{i-n+2}
δ_{i-n+3}
....
δ_{i-1}
δ_i
1
2
3
N-1
N
δ_{i-n}
User
When $\mathbb{Q}$ is full after Enqueue
If $\overline{\mathbb{Q}} > \theta$
Feedback (High Eating Speed)
WARNING
Yes

Figure 4.17: Queue Based Warning System.δ Is One Eating Interval And θ Is The Interval Threshold Value.

samples, the first and last sample always set as the first and last raw data. Then, eighteen features are obtained by down-sampling or up-sampling through the cubic interpolation method as seen in Fig. 4.16. For second octave set, the size was fixed to 10 samples and for third octave set, it was fixed to 5 samples with the same manner.

We selected Deep Neural Network (DNN) model for the machine learning training. The DNN model has four hidden layers with 256 nodes for each layer. The activation function is ReLU, the output layer is sigmoid for binary classification, and the gradient descent optimization is ADAM [97]. The feature extraction method and machine learning algorithm are selected based on our User-Independent experiment evaluation discussed in Sec. 4.6.4.

4.5.5 Interval Calculation and Warning System

In the phase, we obtained the eating interval by simply calculating the time-stamp difference between previous ‘Eat’ and current ’Eat’. The eating interval provides eating speed feedback using queue-based eating speed feedback [84] [2] as seen in Fig. 4.17. First, the eating interval (δ) is continuously stored in Q which is N size queue data structure until the user completes the meal. When the Q is full, the average of the interval data in Q called $\bar{Q}$ is computed. Then, it is compared with a predefined threshold time θ. If the $\bar{Q}$ is greater than θ, the user receives a warning related to her high eating speed and the next eating interval is stored in Q after the front item in Q is removed.

4.6 Experimental Results

In this section, we discuss the performance results of IDEA on the data collected from 36 users. First, we talk about the performance metrics, then we discuss the feature extraction techniques necessary for pre-processing data before the application of the various modeling techniques. Then, we show the performances of various state-of-the-art machine learning techniques for both User-Dependent and User-Independent scenarios. We considered Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM) with linear, polynomial, Radial Basis Function (RBF), and Sigmoid (Tanh) kernels, and Deep Neural Network (DNN) for comparison.

4.6.1 Metrics

For our experiment, F1-Score ($2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$) was utilized as a performance indicator because both precision ($\frac{TP}{TP+FP}$) and recall ($\frac{TP}{TP+FN}$) in terms of identifying 'Eat' segments are crucial evaluation factors. Here, TP is True Positive, FP is False Positive, FN is False Negative, and TN is True Negative. When recall is low, the eating action detection system would miss many eating actions. When precision is low, the system will consider non-eating actions as eating actions. Both results

[2] Published in ICMLA 2018

in unreliable output of the system. Note that the accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$) is an improper performance indicator because of the data imbalance for 'Eat' and 'Non-Eat' segments in any eating episode. An example from our dataset shows that one eating episode consists of 359 'Non-Eat' as compared to only 41 'Eat' segments. Due to the data imbalance, if the system were to predict all segments as 'Non-Eat', the accuracy would be 89.75%. However, this would not be a good performance since the system did not detect any 'Eat' so it couldn't provide the number of correct eating actions or feedback based on eating speed.

4.6.2 Feature Extraction

Two different feature set are considered as an input to the models. The first feature set is **interpolation features** discussed in Sec. 4.5.4. The second feature set is **statistical features**. The statistical feature set is benchmarked features from Thomaz et al. [91] experiment. Although they used only five features: mean, variable, skew-

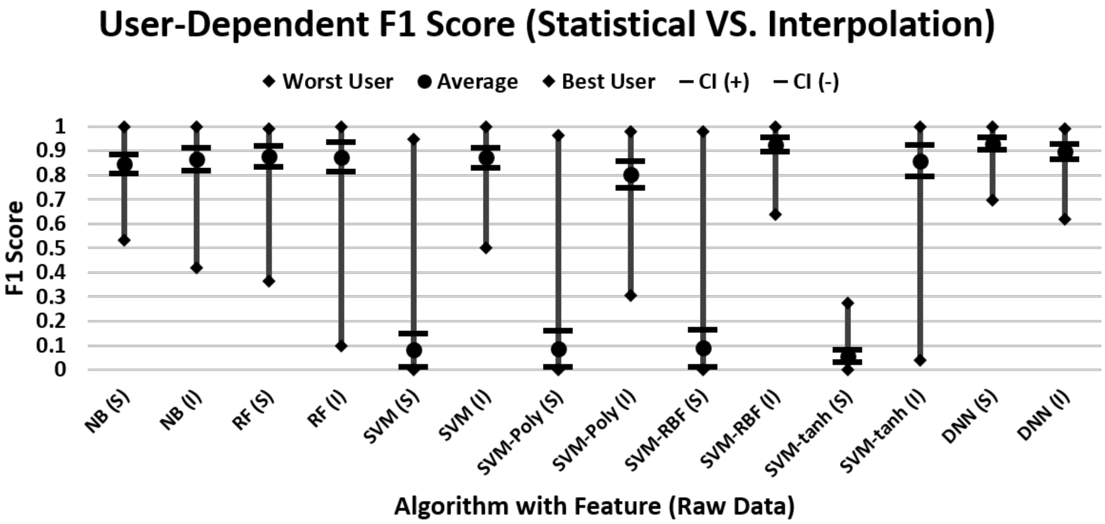


Figure 4.18: Raw Data-based User-Dependent Model Comparison Between Statistical Features And Interpolation Features. (S) Is The Statistical Feature And (I) Is The Interpolation Features. CI Is Confidence Interval (95%).

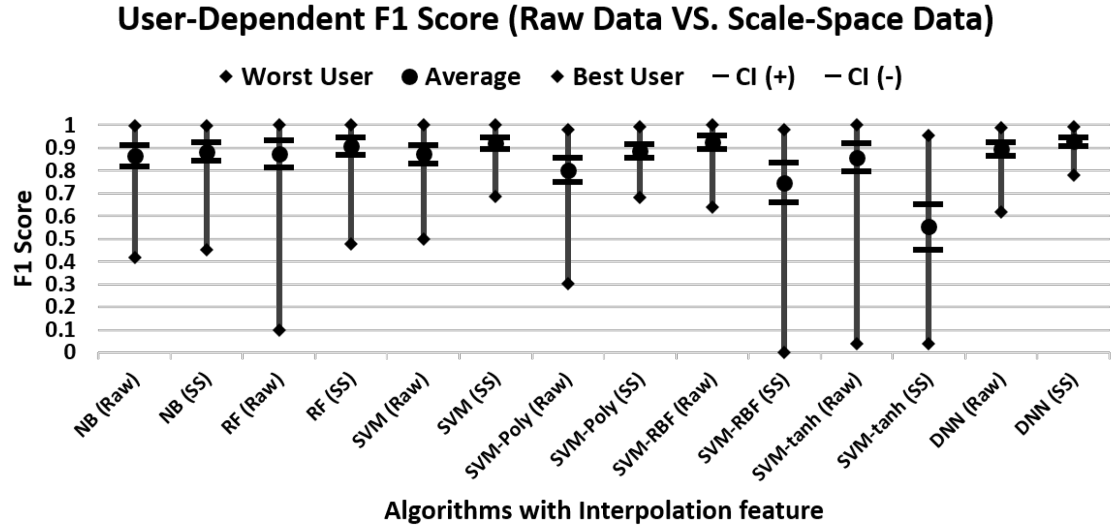


Figure 4.19: Interpolation Feature-based User-Dependent Model Comparison Between Raw Data And Scale-space & DoG Data. (SS) Is The Scale-space & DoG Data. CI Is Confidence Interval (95%).

ness ($\frac{\sum_{n=1}^{N}(x_n-x)^3}{(N-1)s^3}$), kurtosis ($\frac{\sum_{n=1}^{N}(x_n-x)^4}{(N-1)s^4}$), and Root-Mean-Square ($\frac{1}{N}\sum_{n=0}^{N-1}|X_n|^2$), three more features were added: min, max, median. Therefore, the statistical feature set consists of eight features: min, median, mean, max, variable, skewness, kurtosis, and Root-Mean-Square.

In our experiments, there are four different input combinations with: (1) statistical features and raw sensor data, (2) statistical features and scale-space & DoG data, (3) interpolation features and & raw sensor data, and (4) interpolation features and scale-space & DoG data. The feature size per segment for (1) is 144 (8 feature × 18 sensor), for (2) is 3888 (8 feature × 18 sensor × 9 scale-space & DoG × 3 octaves), and for (3) is 360 (20 feature × 18 sensor). When interpolation features from scale-space & DoG data were extracted, each octave has different sampling ratio and hence segment sizes vary. To have uniform feature size, the interpolation feature size of the three octaves was fixed to 20, 10, and 5. Hence, for (4), the feature size is 5670 (35 feature for all octaves × 18 sensor × 9 scale-space & DoG).

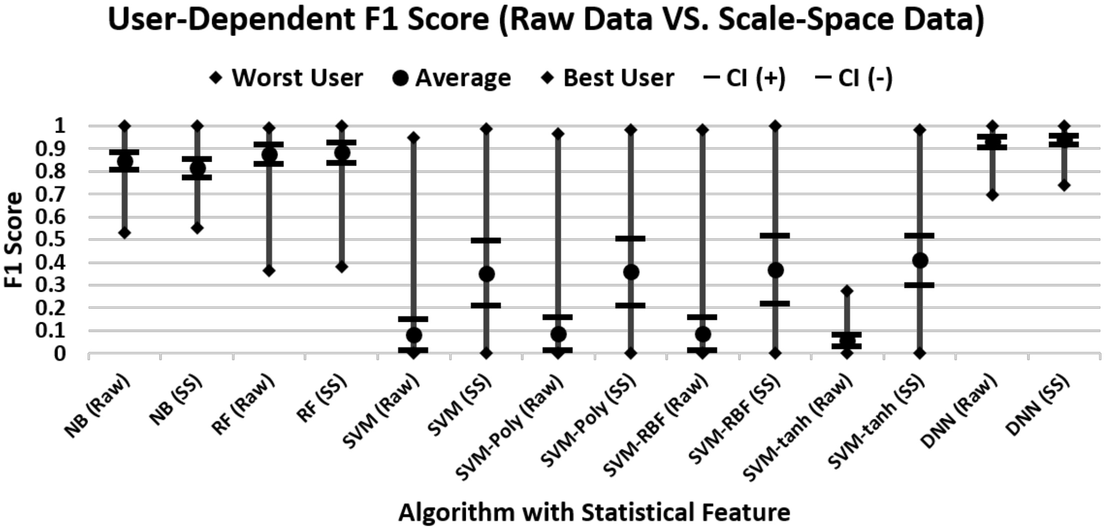


Figure 4.20: Statistical Feature-based User-Dependent Model Comparison Between Raw Data And Scale-space & DoG Data. CI Is Confidence Interval (95%).

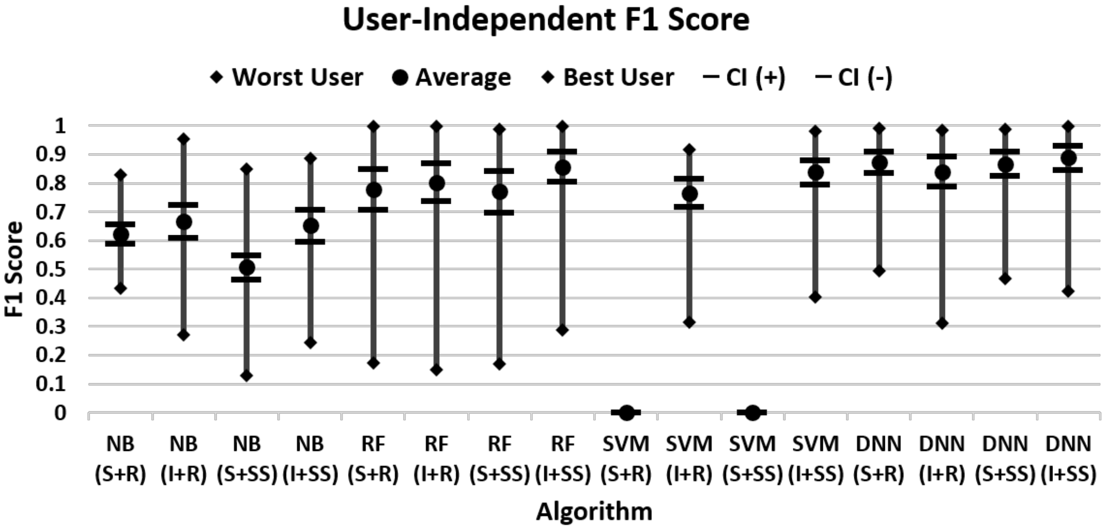


Figure 4.21: User-Independent Scenario Comparison. (S+R) Is Statistical Features And Raw Data, (I+R) Is Interpolation Features And Raw Data, (S+SS) Is Statistical Features And Scale-space & DoG Data, (I+SS) Is Interpolation Features And Scale-space & DoG Data. CI Is Confidence Interval (95%).

4.6.3 User-Dependent

To evaluate the user dependent model, K-Cross (where K=5) validation method was used with five repetitions. The Fig. 4.18 shows that the DNN model with statistical features is the most reliable model. It can be seen that the F1-scores of (i) the worst user (0.7) and (ii) average for all users (0.93) in this model are the highest.

Fig. 4.19, 4.20 show the F1-score comparison between raw data and scale-space & DoG data for interpolation features and statistical features respectively. It can be seen that in general, scale-space & DoG data-based models have better F1-scores than raw data-based models. Also, DNN models have the least variation in F1-score between the worst and average case user as compared to other machine learning models. Overall for the User-Dependent scenarios, the DNN model with interpolation features and scale-space & DoG data is the most reliable (the worst F1 is 0.78 and average F1 is 0.93).

4.6.4 User-Independent

In this scenario, we use labeled data from a set **donors** that does not include the test user to train the machine learning systems. For evaluation, we built different machine learning models by varying the number of donors from 2 to 25. The donors are selected randomly and all experiments were repeated five times. As seen in Fig. 4.21, the average performance of the DNN models is better than that of other machine learning models.

Fig. 4.22 displays DNN model F1-score with respect to number of donors. It can be observed that increasing the number of donors increases the overall F1-score. The DNN model with interpolation features and scale-space & DoG data shows the best F1-score among all DNN models (0.93 F1-score with 21 donors).

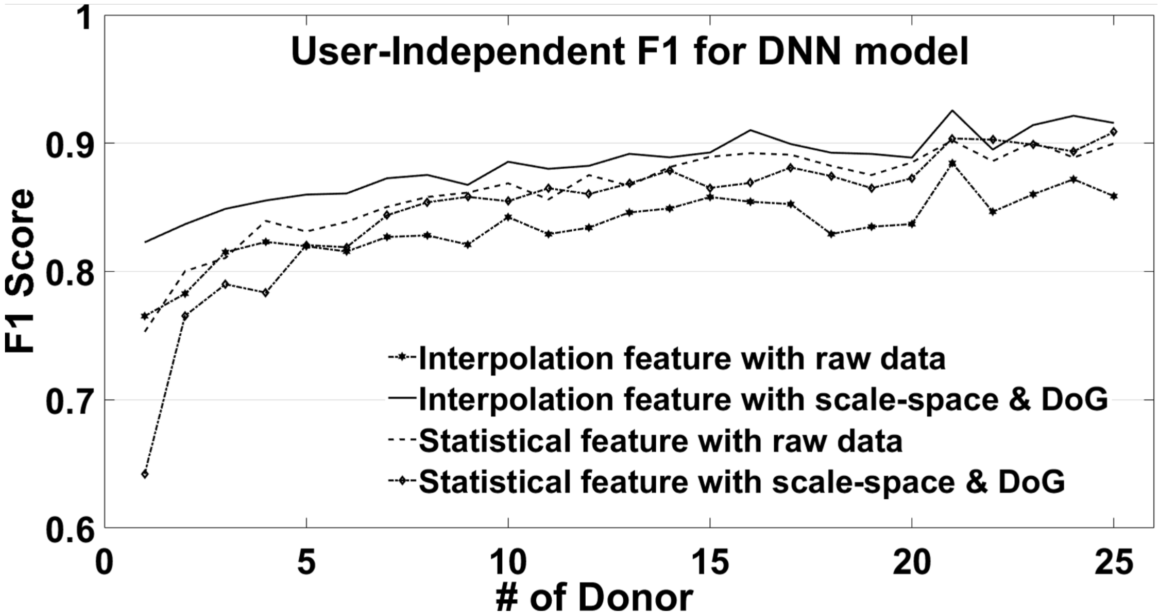


Figure 4.22: Performance Of DNN Model With User-Independent Scenario.

For both User-Dependent and User-Independent evaluations, the DNN model with interpolation features and scale-space & DoG data is the most reliable. Thus, we compared the User-Dependent and User-Independent with 21 donors for this setting. **As seen in Fig. 4.23, average F1 in both of them are similar, but the User-Independent scenario has a much higher range between the average and the worst user. For the worst user group comparison, the average F1 for User-Dependent is 0.87 and the average F1 for User-Independent is 0.58.**

Fig. 4.24 shows the User-Independent F1-score using DNN model with interpolation features and scale-space & DoG data for the worst three users. **Unlike Fig. 4.22, the performance of the model for the worst users does not increase as the number of donors increases.**

By comparing the performances of User-Dependent and User-Independent scenarios, we derived three significant facts:

a) The best model in this study for both User-Independent and User-Dependent scenarios is the DNN model with interpolation features and scale-space & DoG data.

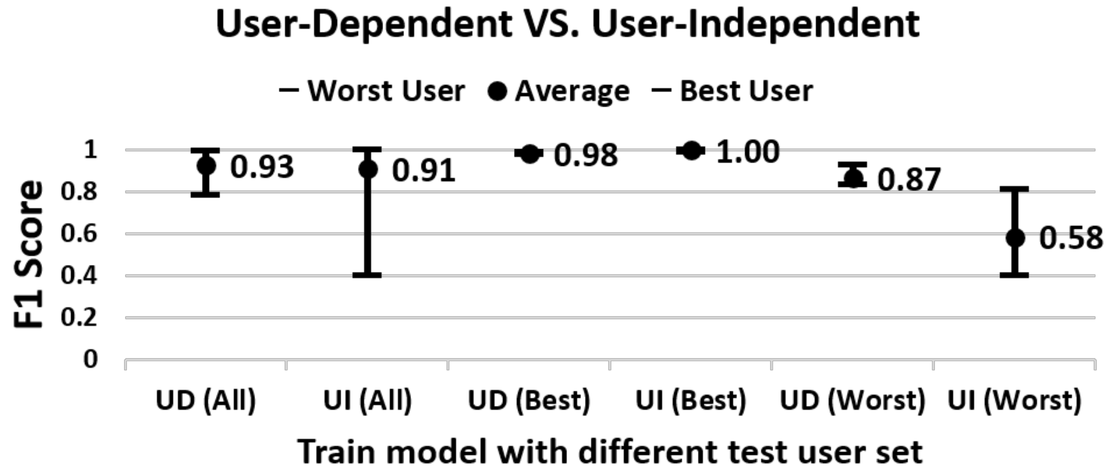


Figure 4.23: F1-score Comparison Between User-Independent And User-Dependent By Different Test User Set Based On DNN Model With Interpolation Features And Scale-space & DoG Data. US Is The User-Dependent And UI Is User-Independent. (All) Is All Test User Set, (Best) Is The Best 10% Test User Set, And (Worst) Is The Worst 10% Test User Set.

b) The User-Independent is less reliable than the User-Dependent scenario. This is especially true for the worst case users. **c)** The average performance in the User-Independent scenario increases with an increase in the number of donors. However, the random fluctuation in F-1 scores for the worst case users shows that the performance is more dependent on the set of donors included rather than the total number of donors.

4.6.5 IDEA

In this section, we first illustrate the overall eating action identification performance of IDEA for: a) all users and b) worst case users. Then we analyze the effectiveness of the two tier approach by individually discussing generalized and personalized model results.

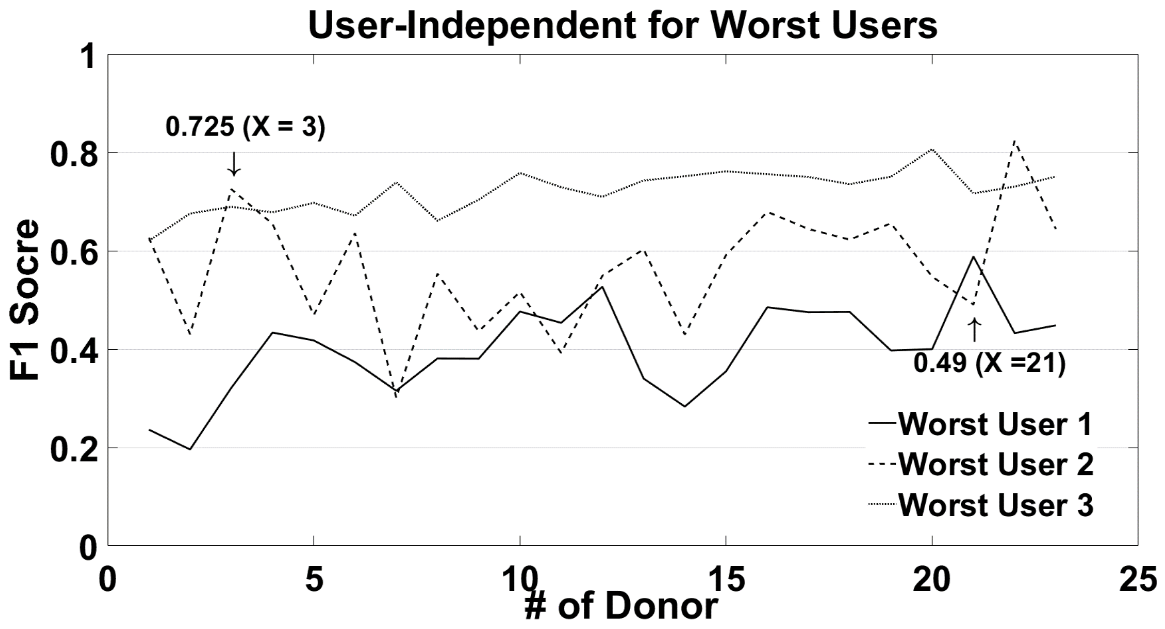


Figure 4.24: Worst User F1-score, User-Independent Scenario.

Overall Performance

Fig. 4.25 shows that on an average IDEA improves the F1-score by 0.05 for 8 donors with respect to the DNN model with interpolation features and scale-space & DoG data which showed the best result in User-Independent discussed in Sec. 4.6.4. When compared with the random-forest model with statistical features and raw data which is a benchmarked from Thomaz et al. [91], IDEA has 0.18 F1-score improvement. IDEA has an F1-score of more than 0.9 if at least 7 users are included in the training set. The figure also shows the performance for the worst-case user set (determined in Sec. 4.6.4). For the worst-case users, on an average IDEA improves F1-score by 0.15 with respect to the best DNN User-Independent model and by 0.12 with respect to the state-of-the-art benchmark.

Fig. 4.32 shows the performance of IDEA for the worst-case users. The best improvement in F1-score achieved by IDEA over the best User-Independent DNN model is 0.27. When we look at the precision and recall for the worst user, we

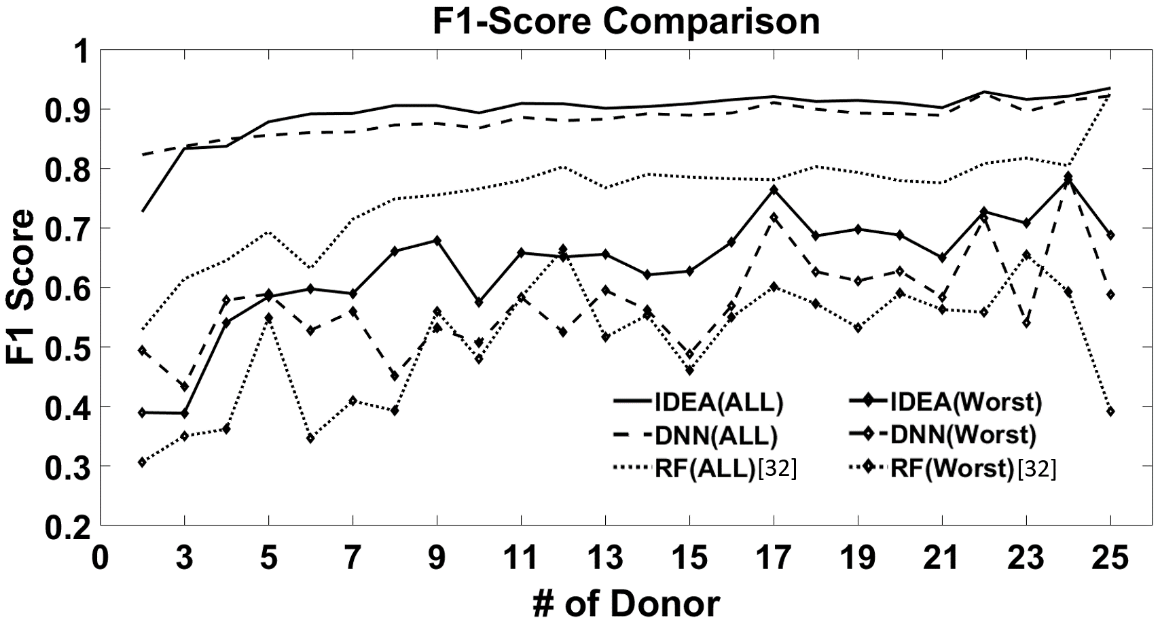


Figure 4.25: F1-score Comparison Between IDEA (User-Specific), User-Independent DNN (interpolation Features And Scale-space & DoG Data), And User-Independent Random Forest (statistical With Raw). (ALL) Is The Average F1 For All Users And (Worst) Is The Average F1 For Worst User Groups

observed that IDEA has nearly the same precision as the User-Independent DNN model but it improves the recall by 0.25. This means that IDEA detects nearly 25% more eating actions. This can result in huge improvements in eating speed accuracy or other applications such as calorie estimation.

THAD-Generalized Model

For the generalized model, Fig. 4.26 shows the area under the curve (AUC) for the receiver operation characteristic (ROC) curve. The ROC curve plots the true positive rate ($\frac{TP}{TP+FN}$) with respect to the false positive rate ($\frac{FP}{FP+TN}$) with varying thresholds on the confidence level of eating actions, which is the output of the generalized model. A value of AUC close to 1 indicates that the classifier maximizes true positive rate while minimizing false positives. From the figure, we establish that with only 8 donors are needed for the worst-case AUC to be above 0.9, and the average AUC to be 0.98.

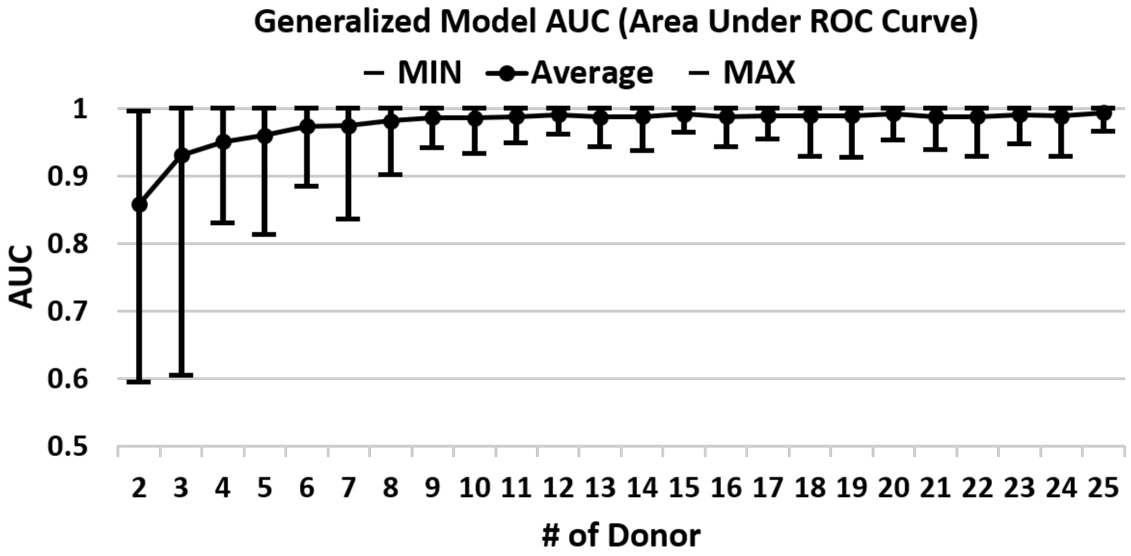


Figure 4.26: Average AUC (Area Under The ROC Curves) For Generalized Model.

The output of the generalized model is 'Similar Users' and 'Dissimilar Users'. IDEA utilizes this information in the personalized model to improve the final performance. However, in this stage there is no ground truth for which of the donors are indeed similar to a given test user. Thus, to evaluate how well THAD identifies similar users, we evaluated the performance of THAD-Personalized Model when trained separately on a) similar users b) dissimilar users and c) all users. It can be seen from Fig. 4.27 that the F-1 Scores for Similar User Model is higher than both of the other models when the number of donors is 3 or higher. This shows that the usage of 'Similar Users' for modeling in fact improves the performance of the personalized model.

THAD-Personalized Model

The final aim of Tier-2 of THAD is the correct identification of 'Definite' segments which are the subset of segments from the test user that can be identified as "eating" or "non-eating" with high confidence levels. It can be seen from Fig. 4.28 that if the

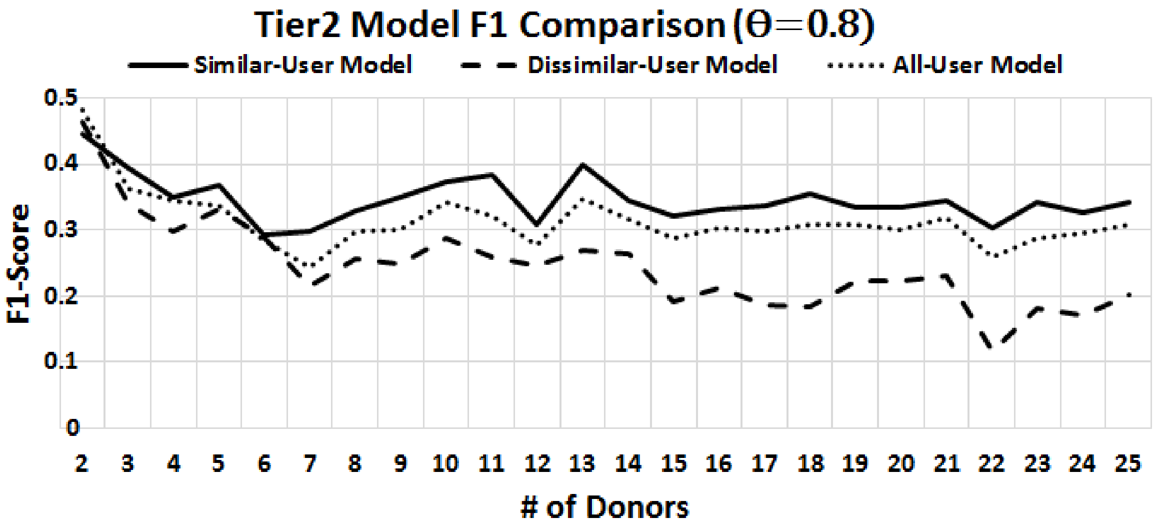


Figure 4.27: Performance Comparison Between Similar, Dissimilar And All Users For Personalized Model.

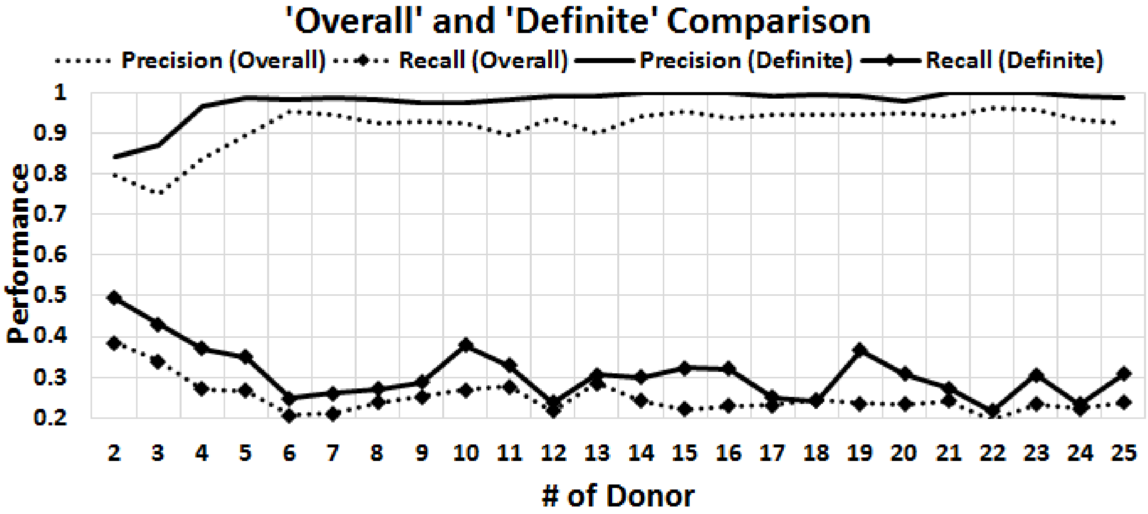


Figure 4.28: Performance Comparison Between Definite Class And Overall.

number of donors is five or higher than we can guarantee a precision of 0.99 or higher for 'Definite' segments. However, this is not true for the 'Overall' segments which consist of both 'Definite' and 'Indefinite' segments. This is the reasoning behind usage of only the 'Definite' segments from the test user for constructing the User-Specific model. It can also be seen that the recall of 'Definite' classes is higher than overall which further supports the usage of 'Definite' classes.

Fig. 4.31 Left shows the variation of relative size of the 'Definite' class with respect to overall test data as the number of donors increase and Right illustrates the misclassification error for segments corresponding to eating or non-eating actions that are identified as ‘Definite’. It can be observed that although the number of segments identified as ‘Definite’ goes down as the number of donors increase, the misclassification rate decreases. The personalized model with eight donors has a misclassification error of 11 out of 10,000, while manual labeling (by human) has 18 errors. The human error is computed by a three-fold cross validation by asking random users to browse through one eating action video once and annotate the eating actions start and end times. This shows that THAD can also be utilized for auto-labeling or assistance for manual labeling.

The above mentioned results show that usage of ‘Definite’ class in personalized model increases accuracy. However, presence of any false negatives or false positives in ‘Definite’ class can significantly affect overall performance. The initial hypothesis of THAD was that usage of similar users can improve the quality of ‘Definite’ class i.e. have reduced false positive and negative examples. In Fig. 4.29 by using similar users instead of all-users the size of the definite class increases without sacrificing accuracy. On the other hand, usage of dissimilar users also increases the size of the definite class, but the results in Fig. 4.30 show that it significantly reduces accuracy.

4.6.6 IDEA Time Complexity

The time complexity with respect to number of multiplications of the steps involved in IDEA is as follows:

a) Scale-space & DoG creation: This involves a convolution operation with Gaussian filters which is quadratic with respect to sample size. The complexity is $O(cn(\tau f)^2 + k\tau f)$, where c is the number of scale-space, n is the number of sensors, τ is the total duration of monitoring, f is the sampling frequency, and k is the total number of

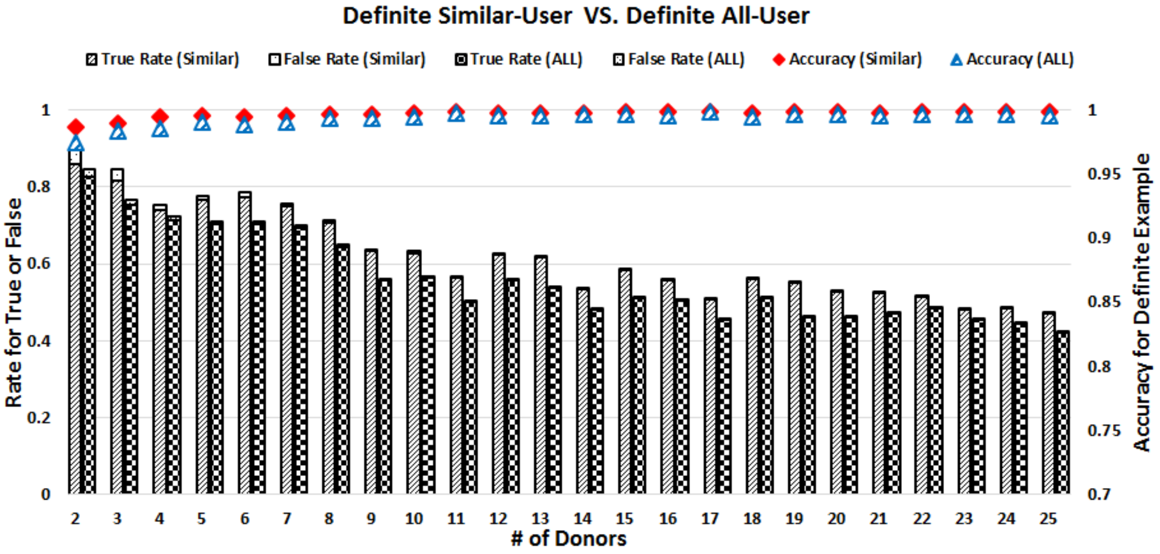


Figure 4.29: Performance Comparison Between Similar-users And All-users In The Definite Class

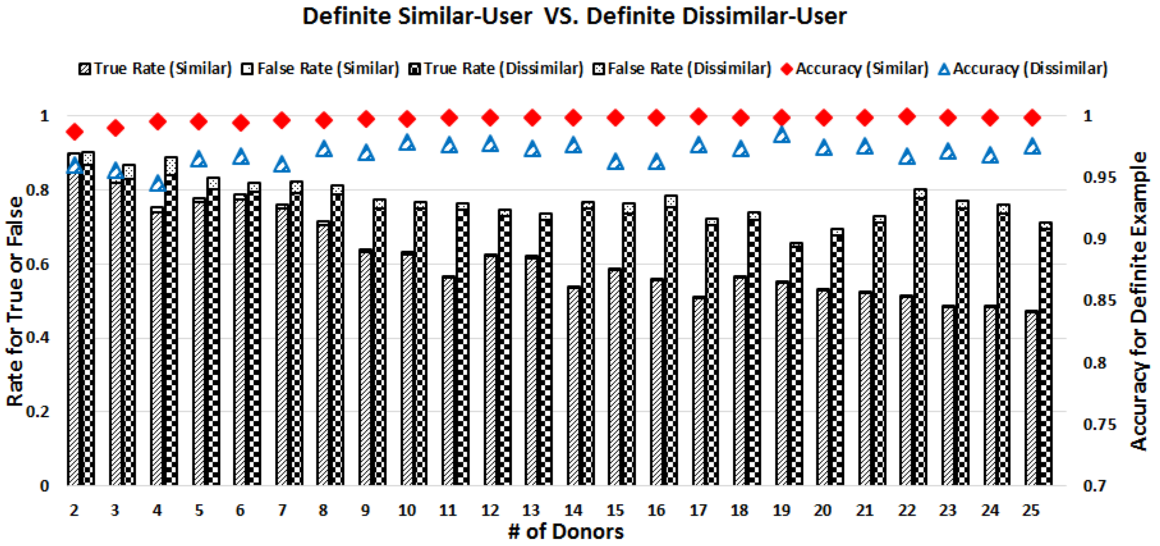


Figure 4.30: Performance Comparison Between Similar-users And Dissimilar-users In The Definite Class

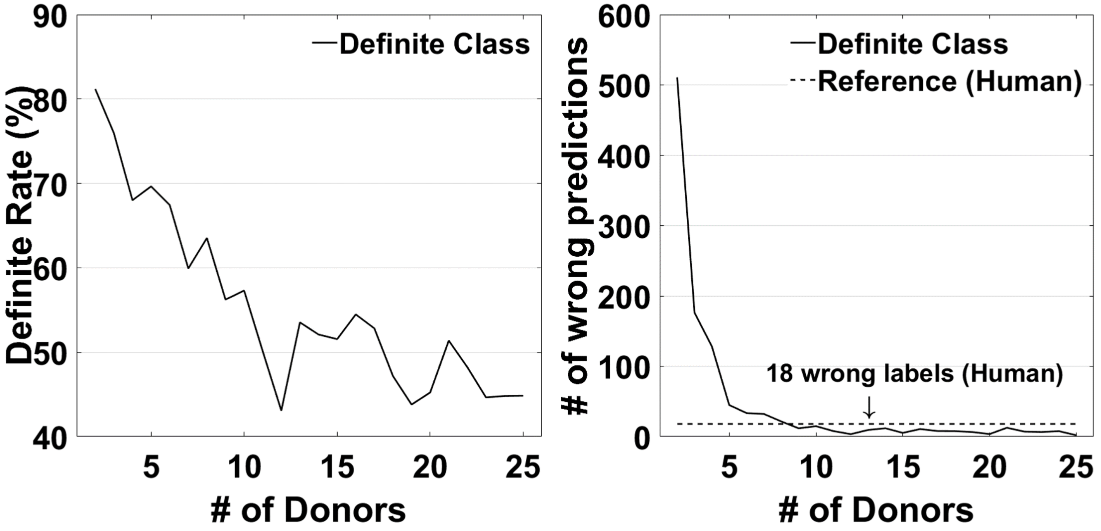


Figure 4.31: THAD Performance For Personalized Model. Left Is Definite Segment Rate And Right Is Definite Segment Accuracy.

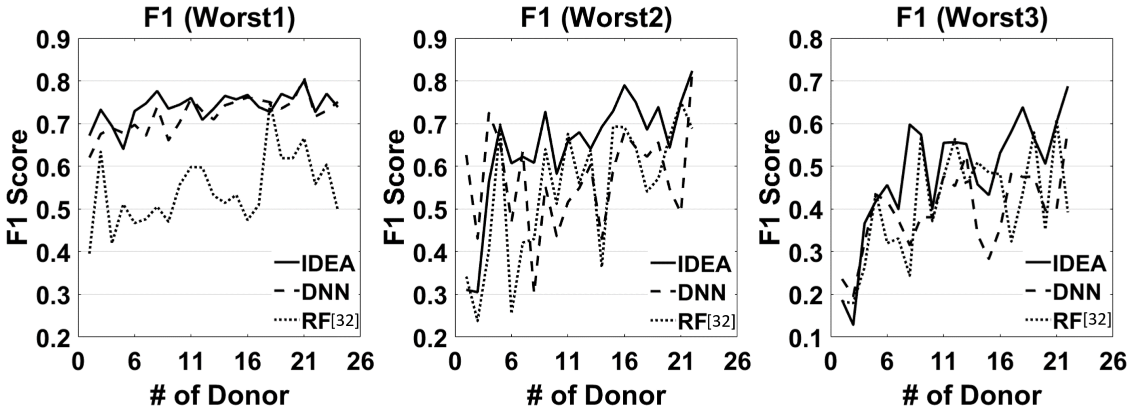


Figure 4.32: F1-score Comparison Between IDEA, User-Independent DNN (interpolation Features And Scale-space & DoG Data), And User-Independent Random Forest (statistical Features And Raw Data) For Worst Users.

DoGs.

b) Segmentation and Cubic spline interpolation to create uniform sample segments: It is linear with respect to sample size, $O((cn+1)\tau f)$.

c) DTW feature matrix creation: It is quadratic with respect to segment size, and it is required to compute a significantly large number of DTW computations. The complexity is $O(sl_{avg}^2 xdenc)$, where sl_{avg} is the average segment length, x is the total number of segments of the test user, d is the total number of donors, and e is the total number of segments labeled 'Eat' for all the donors.

d) The generalized model DNN execution time is only dependent on the segment size and number of segments.

e) Personalized model DTW feature matrix generation: The complexity is $O(K(x-K)sl_{avg}^2cn)$, where K is the total number of segments labeled 'Eat' by the generalized model.

f) Personalized model execution time.

We implemented the backend on Intel(R) Xeon(R) CPU E5-2660 with 8 cores, 16 threads, 64 GB RAM. The DTW computations were offloaded to NVidia GeForce GPU. Overall for a snippet of 2 mins of eating episode, the execution times for a set of 7 donors are {a = 0.043s, b = 6.65s, c = 25.06, d = 10.18s, e = 9.58s and f = 0.22s}. Overall it takes 51.76 seconds to provide eating speed feedback for a 2 min eating episode. Hence, 3 min after the start of an eating episode the user gets feedback on eating speed every 2 mins. To the best of our knowledge, IDEA is the fastest in providing eating speed feedback without any manual input.

4.7 Discussion

4.7.1 *Size of donor set*

The number of donors utilized by IDEA has a big impact on its performance. As seen in the Fig 4.25, when the number of donors was small, the performance of IDEA was found to be worse than the performance of existing techniques (DNN). This is because a small number of donors might not have made a strong 'Generalized model'. Although one of aims of the 'Generalized model' is to extract 'similar donors' as seen in Fig. 4.13, a weak 'Generalized model' cannot match correct 'similar donors' with

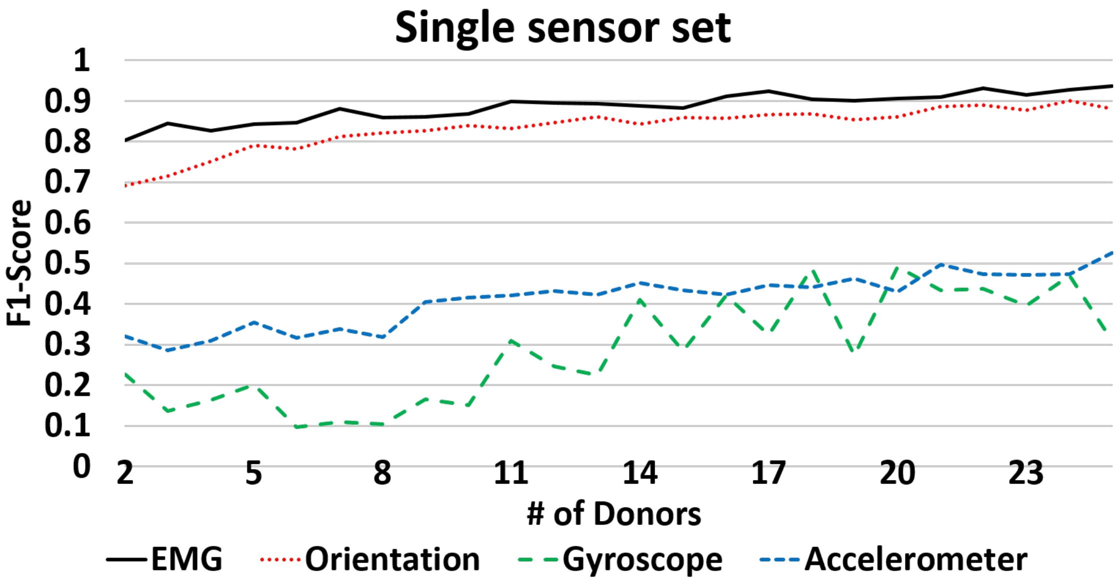


Figure 4.33: F1 Comparison For Single Sensor-sets Using User-Independent DNN Model (interpolation Features And Scale-scale & DoG Data).

a user, which impacts the personalization model negatively. Hence, to build a strong 'Generalized model', enough donors are required. On the other hand, if the number of donors becomes too large, IDEA cannot guarantee real-time performance. Thus, determining the optimal number of donors for the system is of prime importance but it is very challenging. One way to solve this problem would be to start with a high number of donors at first and then gradually make the system more personalized. After IDEA collects data on a user for many meals, data from other donors will become less meaningful because we expect that the 'Generalized model' will pick the same person as a similar donor since the 'definite segments' are reused for next meal′s eating action identification. This can easily be achieved with IDEA, since THAD continuously extracts 'definite segments' and stacks it into the donor set. However, this is left for future work.

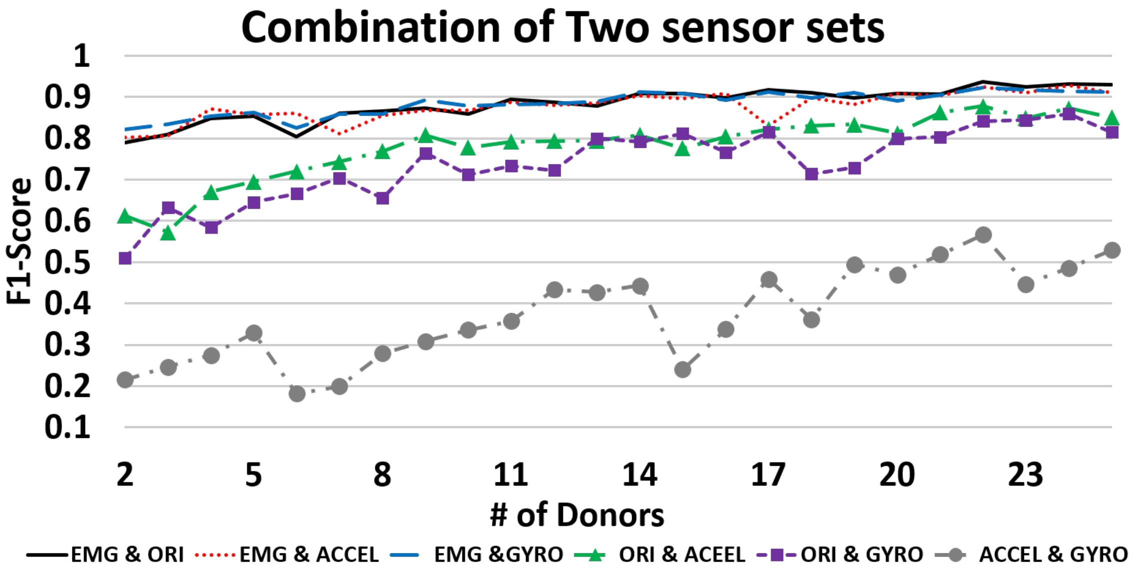


Figure 4.34: F1 Comparison For Combination Of Two Sensor-sets Using User-Independent DNN Model (interpolation Features And Scale-scale & DoG Data). ORI Is Orientation, ACCEL Is Accelerometer, And GYRO Is Gyroscope.

4.7.2 Sensor Selection

The IDEA prototype uses two devices: a smartphone and a wristband as seen in Fig. 4.4. IDEA utilizes Myo as a wristband, which includes four sensor-sets: (1) accelerometer (X, Y, and Z axis), (2) gyroscope (X, Y, and Z axis), (3) orientation (W, X, Y, and Z axis), and (4) electromyogram (eight pods). However, not all commercial wristbands include these four sensor-sets. To verify the feasibility of IDEA using other commercial wristbands, we performed DNN for all combinations of available sensors. Since we have four sensor-sets in our collected dataset, fifteen combinations can be applied. When we apply the various combinations, we fixed the same experimental conditions such as DNN architecture, segmentation, feature extraction, etc. we hypothesize that the lack of good final performance using only accelerometer or gyroscope sensors as seen in Fig. 4.33 is strongly related to the challenges of using accelerometers that are discussed in above. As seen in Fig. 4.33,

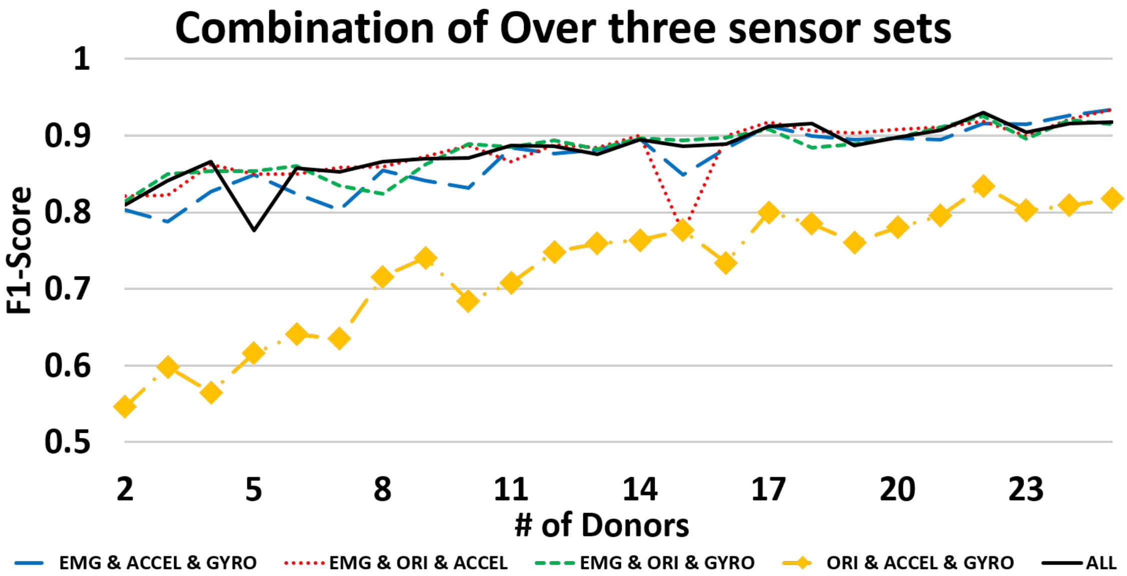


Figure 4.35: F1 Comparison For Combination Of Over Three Sensor-sets Using User-Independent DNN Model (interpolation Features And Scale-scale & DoG Data). ORI Is Orientation, ACCEL Is Accelerometer, And GYRO Is Gyroscope.

4.34, and 4.35, the most significant sensor-set is the EMG set because the combination involving EMG sensors had a better performance compared to other combinations. However, usage of devices with only EMG sensors would not be feasible due to the poor segmentation performance of the EMG sensors as seen in Tab. 4.3. Although, most of the commercially available smartwatches or wristbands do not yet include the EMG sensors, they do include accelerometer, gyroscope or orientation sensors. Fig. 4.33, 4.34 show that the orientation sensor is the second most significant sensor and has reasonable performance even in the absence of EMG sensors. Moreover, orientation values can be computed for devices that do natively include them by using a combination of accelerometer and gyroscope sensors thus the usage of IDEA with commercially available sensors is feasible.

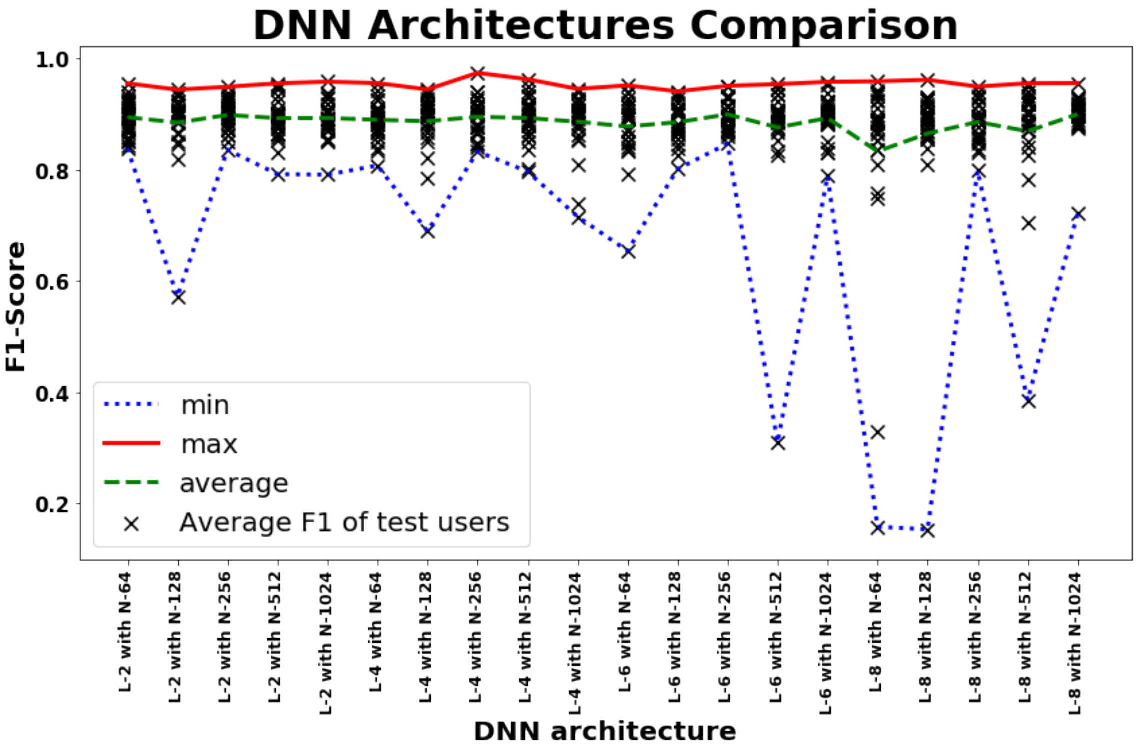


Figure 4.36: Different DNN Architecture Comparison. L-# Is The Number Of Layers And N-# Is The Number Of Nodes For Each Layer.

4.7.3 DNN Architecture Selection

Theoretically, performance of a deep neural network (DNN) should improve as the number of layers and nodes increases. However, in many practical scenarios, the performance of a DNN architecture with a higher number of layers and/or nodes can be poorer compared to the one with a smaller number of layers and/or nodes. This is because having higher number of layers and/or nodes can result in over-fitting. Therefore, determining the optimal number of layers and nodes is an important issue for DNN architectures to be robust in real-world scenarios.

As discussed in Sec. 4.6.3 and 4.6.4, a DNN model, which has four layers and 256 nodes for each layer, showed the best performance. In these sections, we showed that this DNN model overwhelms other machine learning models, but only one DNN architecture was experimented with. However, there may exist many other DNN architectures that might have similar or comparable performances. Hence, in the section, we explore other DNN architectures. There can be infinite possibilities for combining the two parameters: (1) number of layer and (2) number of nodes for each layer. Experimenting over all these possibilities would be impossible, so we considered a few convenient numbers for these two parameters to obtain 20 different DNN architectures. The number of layers were fixed to be either two, four, six, or eight and the number of nodes for each layer were fixed to be one of 64, 128, 256, 512, or 1024. The results for using these 20 different architectures for the same dataset is seen in Fig. 4.36. A different number of donors (from 4 to 30) was used for training each architecture resulting in 27 different trained models for each architecture. Donors were selected randomly from a pool of 36 users and all tests were User-Independent.

As seen in Fig. 4.36, each of the 20 architectures has 27 '×' markers. Each marker is the average performance corresponding to training with a different number of donors. Also, there are three line-graphs which display the minimum, average, and the maximum F-1 scores. The highest F1-score of 0.97 was achieved by the architecture with four layers and 256 nodes when trained with 27 donors. The lowest F1-score of 0.83 trained with six donors for this architecture was similar to the best lowest score across all architectures.

We found that architectures that had more than four layers often showed overfitting. Although loss values during training converged close zero, the performance testing suffered. To solve this issue, we applied Batch Normalization after each hidden layer. Batch Normalization [98] is a popular regularization technique to prevent overfitting.

Most architectures with more than six layers had high variance. It can be seen in the Fig. 4.36 that a few architectures between 'L-2 with N-64' and 'L-6 with N-

256' have similar performances with respect to average F-1 score as well as overall variance. Any of these architectures could be selected and it would make intuitive sense to select the one with the least parameters which is 'L-2 with N-64'. However, when we performed the same experiments without strong regularization we found that the variance for 'L-4 with N-256' was lower than that for 'L-2 with N-64', thus we selected the former architecture.

4.7.4 Distracted Actions

In the section, we experiment the eating action detection with five additional users using IDEA (Instant Detection of Eating Actions) and Deep Neural Network (DNN). To measure the IDEA's performance in noisy environment, these five users have made the distracted activities with eating action during meal: 1) play with a smartphone, 2) touch their head and hair, 3) read a magazine, 4) free talking, and 5) play a laptop game. As seen in Fig. 4.37, the average of F1 score for IDEA showed 0.79 when the number of donors is eight. Also, IDEA showed the better performance as compared to DNN based approach for all different number of donors. We performed an experiment to further evaluate IDEA (Instant Detection of Eating Actions) with five additional users in the presence of distracted eating conditions during the meals. The distracted eating conditions were: 1) play with a smartphone, 2) touch their head and hair, 3) read a magazine, 4) free talking, and 5) play a laptop game. The purpose for choosing these distracted conditions was to replicate noisy conditions that may occur during extreme cases in real world usage. For the experiment, new five subjects were recruited and followed the same data collection approach as discussed in Sec. 4.5.1. Additional, each subject was instructed to replicate a particular distracted pattern randomly throughout the meal. Although, most of the eating episodes in the real world will have far less distracted eating conditions than the ones in this experiment,

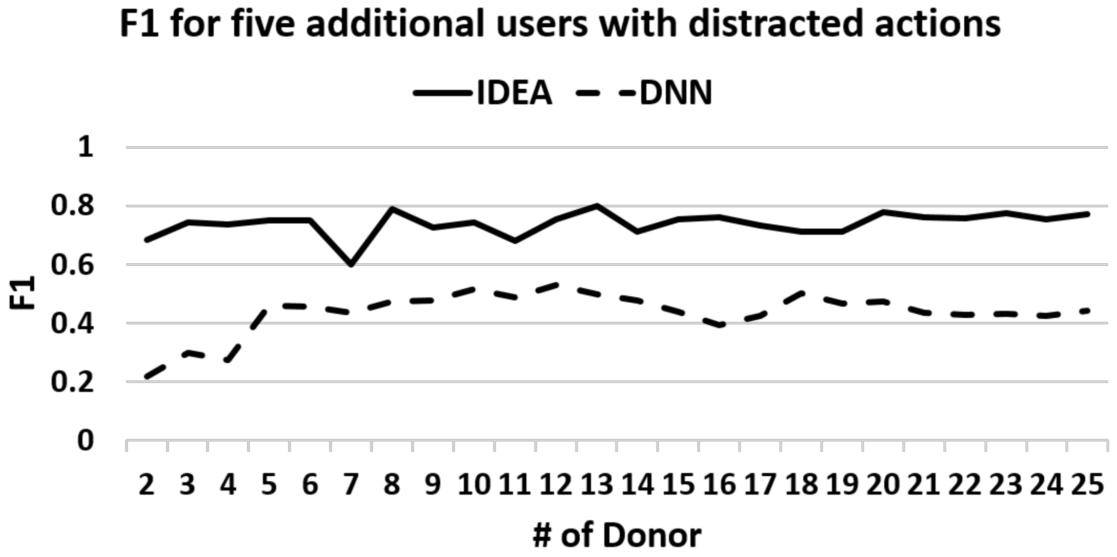


Figure 4.37: F1 Comparison For Five Addition Users With Distracted Actions

we hypothesize that by demonstrating the performance of IDEA in these conditions, we can effectively demonstrate its robustness. The experimental results as seen in Fig. 4.37, show a F1 score comparison between IDEA and a Deep Neural Network (DNN). The DNN architecture that was selected for comparison was the one with the best performance in our experiments as discussed in Sec. 4.6.4. IDEA has overall better performance than the DNN, and achieves an F1 score of 0.79 with only eight donors.

4.7.5 Limitations

According to our usability study discussed in Sec 4.4, we found that in general, participants preferred the usage of smartwatch for monitoring food intake. However, when it comes to other specialized wearables such as wristbands, subjects have their own diverse preferences. For instance, any head mounted sensor was rejected outright by the users. In summary, users prefer a system that can operate on wearables that are already in use. A dedicated system only for food intake estimation may not be acceptable.

IDEA primarily operates with a wristband that can measure accelerometer, gyroscope, orientation, and EMG activity of the wrist. If the user does not want to use a wristband, IDEA can also operate with a smartwatch. Although current commercial smartwatches do not support EMG sensing, IDEA has reasonable performance in the absence of EMG sensors as seen in Fig. 4.33, 4.34. An argument against the usage of smartwatch is that users tend to wear smartwatches their left arm but eat with their right arm. In such cases, eating action monitoring becomes infeasible. Hence, actual users in real-world should be instructed to wear the smartwatch on their dominant-hand.

Chapter 5

CONCLUSIONS

In this dissertation, a pervasive diet monitoring system using multiple sensors was presented to monitor: i) the intake food using the thermal map which provides crucial information for fully automatic and accurate food recognition and ii) eating actions using only a wristband sensor and without the need for collecting training data from the user. For i) monitoring food, the contributions are three novel techniques for a) image registration between thermal and color images, b) food segmentation, and c) food identification.

Existing image registration techniques could not be used because of the differences in the nature of images taken from a thermal camera and color camera. Thus a novel technique had to be developed to facilitate the image registration between the thermal and color images. This enabled the fusion of knowledges obtained from the visible spectrum and infrared spectrum which greatly enhanced the performance of the food segmentation process. Critically, with this methodology, the food segmentation could be performed without any user intervention which is a big step towards high adherence. The high performance of food segmentation also directly contributes to improve the food identification accuracy. Specifically, a combination of these techniques helped me improve the state-of-art for food item recognition to 88.93% (an improvement of nearly 25%). This was achieved using dimensionality reduction, features fusion, and automatic and high quality food segmentation.

Then, for ii) automatic monitoring of eating actions, the results for various experiments and comparisons with traditional machine learning techniques was presented. It was demonstrated that automatic monitoring can be achieved in a plug-n-play manner without the need for any initialization from the user. We also showed detailed experimental results demonstrating the efficacy of such a techniques for User-Independent Scenarios which is very important for user acceptance. This was achieved by using a novel automatic segmentation method for continuous activity recognition. Although the segmentation method for monitoring of eating activity was utilized, it can be used for automatic segmentation of any activity that uses wristbands. This will greatly benefit future researches in this domain. Finally, summary is that the results of three surveys to ascertain user needs and evaluate the usability of my approach which can also provide insight to future researchers.

REFERENCES

[1] R. C. Baker and D. S. Kirschenbaum, “Self-monitoring may be necessary for successful weight control,” *Behavior Therapy*, vol. 24, no. 3, pp. 377–394, 1993.

[2] ——, “Weight control during the holidays: highly consistent self-monitoring as a potentially useful coping mechanism.” *Health Psychology*, vol. 17, no. 4, p. 367, 1998.

[3] X. Wang, Y. Li, H. Wei, and F. Liu, “An asift-based local registration method for satellite imagery,” *Remote Sensing*, vol. 7, no. 6, pp. 7044–7061, 2015.

[4] G. Yu and J.-M. Morel, “Asift: An algorithm for fully affine invariant comparison,” *Image Processing On Line*, vol. 1, 2011.

[5] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[6] T. L. Burrows, R. J. Martin, and C. E. Collins, “A systematic review of the validity of dietary assessment methods in children when compared with the method of doubly labeled water,” *Journal of the American Dietetic Association*, vol. 110, no. 10, pp. 1501–1510, 2010.

[7] C. M. Champagne, G. A. Bray, A. A. Kurtz, J. B. R. Monteiro, E. Tucker, J. Volaufova, and J. P. Delany, “Energy intake and energy expenditure: a controlled study comparing dietitians and non-dietitians,” *Journal of the American Dietetic Association*, vol. 102, no. 10, pp. 1428–1432, 2002.

[8] K. Poslusna, J. Ruprich, J. H. de Vries, M. Jakubikova, and P. van’t Veer, “Misreporting of energy and micronutrient intake estimated by food records and 24 hour recalls, control and adjustment methods in practice,” *British Jnl. of Nutrition*, vol. 101, no. S2, pp. S73–S85, 2009.

[9] K. Wakai, “A review of food frequency questionnaires developed and validated in japan,” *Jnl. of epidemiology*, vol. 19, no. 1, pp. 1–11, 2009.

[10] J. R. Hebert, C. B. Ebbeling, C. E. Matthews, T. G. Hurley, M. Yunsheng, S. Druker, and L. Clemow, “Systematic errors in middle-aged women’s estimates of energy intake: comparing three self-report measures to total energy expenditure from doubly labeled water,” *Annals of epidemiology*, vol. 12, no. 8, pp. 577–586, 2002.

[11] T. Joutou and K. Yanai, “A food image recognition system with multiple kernel learning,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on.* IEEE, 2009, pp. 285–288.

[12] H. Hoashi, T. Joutou, and K. Yanai, “Image recognition of 85 food categories by feature fusion,” in *Multimedia (ISM), 2010 IEEE International Symposium on.* IEEE, 2010, pp. 296–301.

[13] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Multimedia and Expo (ICME),.* IEEE, 2012, pp. 25–30.

[14] Y. Kawano and K. Yanai, "Real-time mobile food recognition system," in *Computer Vision and Pattern Recognition Workshops (CVPRW),.* IEEE, 2013, pp. 1–7.

[15] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, "Measuring calorie and nutrition from food image," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 8, pp. 1947–1956, 2014.

[16] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, "Food image analysis: Segmentation, identification and weight estimation," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on.* IEEE, 2013, pp. 1–6.

[17] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Computer Vision and Pattern Recognition (CVPR),.* IEEE, 2010, pp. 2249–2256.

[18] B. M. Chaudhry, C. Schaefbauer, B. Jelen, K. A. Siek, and K. Connelly, "Evaluation of a food portion size estimation interface for a varying literacy population," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* ACM, 2016, pp. 5645–5657.

[19] G. J. Hollands, T. M. Marteau, and P. C. Fletcher, "Non-conscious processes in changing health-related behaviour: a conceptual analysis and framework," *Health psychology review*, pp. 1–14, 2016.

[20] J. Oh and U. Lee, "Exploring ux issues in quantified self technologies," in *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU).* IEEE, 2015, pp. 53–59.

[21] J.-S. Shim, K. Oh, and H. C. Kim, "Dietary assessment methods in epidemiologic studies," *Epidemiology and health*, vol. 36, 2014.

[22] J. R. Pleis, J. W. Lucas, and B. W. Ward, "Summary health statistics for us adults: National health interview survey, 2008." *Vital and health statistics. Series 10, Data from the National Health Survey*, no. 242, pp. 1–157, 2009.

[23] T. A. Wadden, K. D. Brownell, and G. D. Foster, "Obesity: responding to the global epidemic." *Journal of consulting and clinical psychology*, vol. 70, no. 3, p. 510, 2002.

[24] F. M. Sacks and M. Katan, "Randomized clinical trials on the effects of dietary fat and carbohydrate on plasma lipoproteins and cardiovascular disease," *The American journal of medicine*, vol. 113, no. 9, pp. 13–24, 2002.

[25] C. for Disease Control *et al.*, "Maps of trends in diagnosed diabetes and obesity," *Center for Disease Control and Prevention Division of Diabetes Translation*, 2015.

[26] S. Sharma, S. Vik, M. Pakseresht, L. Shen, and L. N. Kolonel, "Diet impacts mortality from cancer: results from the multiethnic cohort study," *Cancer Causes & Control*, vol. 24, no. 4, pp. 685–693, 2013.

[27] B. A. Swinburn, I. Caterson, J. C. Seidell, and W. James, "Diet, nutrition and the prevention of excess weight gain and obesity," *Public health nutrition*, vol. 7, no. 1a, pp. 123–146, 2004.

[28] "Maps of trends in diagnosed diabetes and obesity," September 2016, accessed: 2016-10-1. [Online]. Available: https://www.downtoearth.org/news/2016-09/9171/diet-related-illnesses-cost-us-economy-1-trillion-annually

[29] M. Rao, A. Afshin, G. Singh, and D. Mozaffarian, "Do healthier foods and diet patterns cost more than less healthy options? a systematic review and meta-analysis," *BMJ open*, vol. 3, no. 12, p. e004277, 2013.

[30] A. Carlson and E. Frazão, "Food costs, diet quality and energy balance in the united states," *Physiology & behavior*, vol. 134, pp. 20–31, 2014.

[31] K. Aizawa, K. Maeda, M. Ogawa, Y. Sato, M. Kasamatsu, K. Waki, and H. Takimoto, "Comparative study of the routine daily usability of foodlog: A smartphone-based food recording tool assisted by image retrieval," *Journal of diabetes science and technology*, vol. 8, no. 2, pp. 203–208, 2014.

[32] N. Hongu, B. T. Pope, P. Bilgiç, B. J. Orr, A. Suzuki, A. S. Kim, N. C. Merchant, and D. J. Roe, "Usability of a smartphone food picture app for assisting 24-hour dietary recall: a pilot study," *Nutrition research and practice*, vol. 9, no. 2, pp. 207–212, 2015.

[33] L. M. Y. Chung, J. W. Y. Chung, and T. K. S. Wong, "Usability test of an interactive dietary recording." *International Electronic Journal of Health Education*, vol. 12, pp. 123–134, 2009.

[34] S. De Francisco, F. Freijser, I. van der Lee, M. van Sinderen, S. Verburg, and J. Yao, "Myfitnesspal iphone app usability test."

[35] P. J. Huth, V. L. Fulgoni, D. R. Keast, K. Park, and N. Auestad, "Major food sources of calories, added sugars, and saturated fat and their contribution to essential nutrient intakes in the us diet: data from the national health and nutrition examination survey (2003–2006)," *Nutrition journal*, vol. 12, no. 1, p. 116, 2013.

[36] P. Pouladzadeh, P. Kuhad, S. V. B. Peddi, A. Yassine, and S. Shirmohammadi, "Mobile cloud based food calorie measurement," in *Multimedia and Expo Workshops (ICMEW)*. IEEE, 2014, pp. 1–6.

[37] P. Pouladzadeh, S. Shirmohammadi, A. Bakirov, A. Bulut, and A. Yassine, "Cloud-based svm for food categorization," *Multimedia Tools and Applications*, pp. 1–18, 2014.

[38] P. Pouladzadeh, S. Shirmohammadi, and A. Yassine, "Using graph cut segmentation for food calorie measurement," in *Medical Measurements and Applications (MeMeA).* IEEE, 2014, pp. 1–6.

[39] M.-Y. Chen, Y.-H. Yang, C.-J. Ho, S.-H. Wang, S.-M. Liu, E. Chang, C.-H. Yeh, and M. Ouhyoung, "Automatic chinese food identification and quantity estimation," in *SIGGRAPH Asia.* ACM, 2012, p. 29.

[40] C. K. Martin, T. Nicklas, B. Gunturk, J. B. Correa, H. R. Allen, and C. Champagne, "Measuring food intake with digital photography," *Journal of Human Nutrition and Dietetics*, vol. 27, no. s1, pp. 72–81, 2014.

[41] J. Lee, A. Banerjee, and S. K. S. Gupta, "Mt-diet: Automated smartphone based diet assessment with infrared images," in *PerCom.* IEEE, 2016.

[42] A. Jarc, J. Perš, P. Rogelj, M. Perše, and S. Kovačič, *Texture features for affine registration of thermal (FLIR) and visible images.* Citeseer, 2007.

[43] R. Istenic, D. Heric, S. Ribaric, and D. Zazula, "Thermal and visual image registration in hough parameter space," in *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on.* IEEE, 2007, pp. 106–109.

[44] S. Cao, J. Jiang, G. Zhang, and Y. Yuan, "An edge-based scale-and affine-invariant algorithm for remote sensing image registration," *International journal of remote sensing*, vol. 34, no. 7, pp. 2301–2326, 2013.

[45] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[46] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[47] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition. CVPR.*, vol. 1. IEEE, 2005, pp. 886–893.

[48] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, 1996.

[49] R. L. Duyff, *American dietetic association complete food and nutrition guide.* Houghton Mifflin Harcourt, 2012.

[50] T. Maruyama, Y. Kawano, and K. Yanai, "Real-time mobile recipe recommendation system using food ingredient recognition," in *Proceedings of the 2nd ACM international workshop on Interactive multimedia on mobile and portable devices.* ACM, 2012, pp. 27–34.

[51] P. Pouladzadeh, P. Kuhad, S. V. B. Peddi, A. Yassine, and S. Shirmohammadi, "Food calorie measurement using deep learning neural network," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings.* IEEE, 2016, pp. 1–6.

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

[53] L. Bally, J. Dehais, C. T. Nakas, M. Anthimopoulos, M. Laimer, D. Rhyner, G. Rosenberg, T. Zueger, P. Diem, S. Mougiakakou *et al.*, "Carbohydrate estimation supported by the gocarb system in individuals with type 1 diabetes: A randomized prospective pilot study," *Diabetes care*, vol. 40, no. 2, pp. e6–e7, 2017.

[54] M. Bolanos and P. Radeva, "Simultaneous food localization and recognition," in *2016 23rd International Conference on Pattern Recognition (ICPR).* IEEE, 2016, pp. 3140–3145.

[55] Accessed: 2015-07-10. [Online]. Available: http://www.thermal.com/.

[56] A. Torabi, G. Massé, and G.-A. Bilodeau, "An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 210–221, 2012.

[57] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[58] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding," *Image and vision computing*, vol. 8, no. 1, pp. 71–77, 1990.

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

[60] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision.* Springer, 2006, pp. 430–443.

[61] T. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[62] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[63] J. Serra, *Image analysis and mathematical morphology.* Academic Press, Inc., 1983.

[64] R. M. Haralock and L. G. Shapiro, *Computer and robot vision.* Addison-Wesley Longman Publishing Co., Inc., 1991.

[65] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Identification using encrypted biometrics," in *Computer Analysis of Images and Patterns.* Springer, 2013, pp. 440–448.

[66] I. Jolliffe, *Principal component analysis.* Wiley Online Library, 2002.

[67] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," *arXiv preprint arXiv:1207.3538*, 2012.

[68] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[69] R. A. Mekary, E. Giovannucci, W. C. Willett, R. M. van Dam, and F. B. Hu, "Eating patterns and type 2 diabetes risk in men: breakfast omission, eating frequency, and snacking," *The American journal of clinical nutrition*, vol. 95, no. 5, pp. 1182–1189, 2012.

[70] T. Ohkuma, H. Fujii, M. Iwase, Y. Kikuchi, S. Ogata, Y. Idewaki, H. Ide, Y. Doi, Y. Hirakawa, N. Mukai *et al.*, "Impact of eating rate on obesity and cardiovascular risk factors according to glucose tolerance status: the fukuoka diabetes registry and the hisayama study," *Diabetologia*, vol. 56, no. 1, pp. 70–77, 2013.

[71] S. Tanihara, T. Imatoh, M. Miyazaki, A. Babazono, Y. Momose, M. Baba, Y. Uryu, and H. Une, "Retrospective longitudinal study on the relationship between 8-year weight change and current eating speed," *Appetite*, vol. 57, no. 1, pp. 179–183, 2011.

[72] A. Kokkinos, C. W. le Roux, K. Alexiadou, N. Tentolouris, R. P. Vincent, D. Kyriaki, D. Perrea, M. A. Ghatei, S. R. Bloom, and N. Katsilambros, "Eating slowly increases the postprandial response of the anorexigenic gut hormones, peptide yy and glucagon-like peptide-1," *The Journal of Clinical Endocrinology & Metabolism*, vol. 95, no. 1, pp. 333–337, 2010.

[73] D. Fan, J. Gong, and J. Lach, "Eating gestures detection by tracking finger motion." in *Wireless Health*, 2016, pp. 1–6.

[74] S. Whitehouse, K. Yordanova, A. Paiement, and M. Mirmehdi, "Recognition of unscripted kitchen activities and eating behaviour for health monitoring," 2016.

[75] S. Zhang, M. H. Ang, W. Xiao, and C. K. Tham, "Detection of activities by wireless sensors for daily life surveillance: eating and drinking," *Sensors*, vol. 9, no. 3, pp. 1499–1517, 2009.

[76] J. Chung, J. Chung, W. Oh, Y. Yoo, W. G. Lee, and H. Bang, "A glasses-type wearable device for monitoring the patterns of food intake and facial activity," *Scientific Reports*, vol. 7, p. 41690, 2017.

[77] A. Bedri, R. Li, M. Haynes, R. P. Kosaraju, I. Grover, T. Prioleau, M. Y. Beh, M. Goel, T. Starner, and G. Abowd, "Earbit: Using wearable sensors to detect eating episodes in unconstrained environments," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 37, 2017.

[78] J. Kim, K.-J. Lee, M. Lee, N. Lee, B.-C. Bae, G. Lee, J. Cho, Y. M. Shim, and J.-D. Cho, "Slowee: A smart eating-speed guide system with light and vibration feedback," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems.* ACM, 2016, pp. 2563–2569.

[79] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, and G.-Z. Yang, "An intelligent food-intake monitoring system using wearable sensors," in *Wearable and Implantable Body Sensor Networks (BSN), 2012 Ninth International Conference on.* IEEE, 2012, pp. 154–160.

[80] J. Lee, P. Paudyal, A. Banerjee, and S. K. Gupta, "Idea: Instant detection of eating action usingwrist-worn sensors in absence of user-specific model," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization.* ACM, 2018.

[81] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[82] J. Lee, A. Banerjee, and S. K. Gupta, "Mt-diet: Automated smartphone based diet assessment with infrared images," in *PerCom.* IEEE, 2016, pp. 1–6.

[83] 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," in *Proceedings of the 51st Annual Design Automation Conference.* ACM, 2014, pp. 1–6.

[84] J. Lee, P. Paudyal, A. Banerjee, and S. K. Gupta, "Fit-eve&adam: Estimation of velocity & energy for automated diet activity monitoring," in *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on.* IEEE, 2017, pp. 1071–1074.

[85] B. Zhou, J. Cheng, M. Sundholm, A. Reiss, W. Huang, O. Amft, and P. Lukowicz, "Smart table surface: A novel approach to pervasive dining monitoring," in *PerCom.* IEEE, 2015, pp. 155–162.

[86] A. Kadomura, C.-Y. Li, Y.-C. Chen, K. Tsukada, I. Siio, and H.-h. Chu, "Sensing fork: eating behavior detection utensil and mobile persuasive game," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems.* ACM, 2013, pp. 1551–1556.

[87] A. Kadomura, C.-Y. Li, K. Tsukada, H.-H. Chu, and I. Siio, "Persuasive technology to improve eating behavior using a sensor-embedded fork," in *Proceedings of the 2014 acm international joint conference on pervasive and ubiquitous computing.* ACM, 2014, pp. 319–329.

[88] S. Cadavid, M. Abdel-Mottaleb, and A. Helal, "Exploiting visual quasi-periodicity for real-time chewing event detection using active appearance models and support vector machines," *Personal and Ubiquitous Computing*, vol. 16, no. 6, pp. 729–739, 2012.

[89] T. Rahman, A. T. Adams, M. Zhang, E. Cherry, B. Zhou, H. Peng, and T. Choudhury, "Bodybeat: a mobile system for sensing non-speech body sounds." in *MobiSys*, vol. 14, 2014, pp. 2–13.

[90] T. Olubanjo and M. Ghovanloo, "Real-time swallowing detection based on tracheal acoustics," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4384–4388.

[91] E. Thomaz, I. Essa, and G. D. Abowd, "A practical approach for recognizing eating moments with wrist-mounted inertial sensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 1029–1040.

[92] S. Sen, V. Subbaraju, A. Misra, R. K. Balan, and Y. Lee, "The case for smartwatch-based diet monitoring," in *Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2015, pp. 585–590.

[93] ——, "Experiences in building a real-world eating recogniser," in *Proceedings of the 4th International on Workshop on Physical Analytics*. ACM, 2017, pp. 7–12.

[94] https://www.myo.com/., "Accessed: 2015-07-10."

[95] M. Šenk and L. Cheze, "Rotation sequence as an important factor in shoulder kinematics," *Clinical biomechanics*, vol. 21, pp. S3–S8, 2006.

[96] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[97] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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