Personal Dietary Assessment Using Mobile Devices

Anand Mariappan[a], **Marc Bosch**[a], **Fengqing Zhu**[a], **Carol J. Boushey**[b], **Deborah A. Kerr**[c], **David S. Ebert**[a], and **Edward J. Delp**[a]

[a]School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana USA

[b]Department of Foods and Nutrition, Purdue University, West Lafayette, Indiana USA

[c]Curtin Institute of Technology, Perth, Western Australia

Abstract

Dietary intake provides valuable insights for mounting intervention programs for prevention of disease. With growing concern for adolescent obesity, the need to accurately measure diet becomes imperative. Assessment among adolescents is problematic as this group has irregular eating patterns and have less enthusiasm for recording food intake. Preliminary studies among adolescents suggest that innovative use of technology may improve the accuracy of diet information from young people. In this paper we describe further development of a novel dietary assessment system using mobile devices. This system will generate an accurate account of daily food and nutrient intake among adolescents. The mobile computing device provides a unique vehicle for collecting dietary information that reduces burden on records that are obtained using more classical approaches. Images before and after foods are eaten can be used to estimate the amount of food consumed.

Keywords

dietary assessment; diet record method; mobile telephone; user interface; mobile device; classification; pattern recognition; image texture; feature extraction

1. Introduction

Dietary intake provides valuable insights for mounting intervention programs for prevention of disease [1,2]. However, accurate assessment of diet is problematic. Immerging technology in mobile telephones (cell phones) with higher resolution images, improved memory capacity, and faster processors, allow these devices to process information not previously possible. To adequately address these challenges, we further describe enhancements to a novel dietary assessment system we are developing using a mobile device (e.g. a mobile telephone or pda-like device) to provide an accurate account of daily food and nutrient intake [3].

Our goal is to use a mobile device with a built-in camera, network connectivity, integrated image analysis and visualization tools with a nutrient database, to allow a user to discretely record foods eaten. Mobile devices, such as PDAs and mobile telephones with cameras, are general purpose computing devices that have a great deal of computational power that can be exploited for solutions to this problem. Mobile computing devices provide an unique vehicle for collecting dietary information [3]. Images of food can be marked with a variety of input methods that link the item for image analysis to estimate the amount of food.

Images acquired before and after foods are eaten can estimate the amount of food consumed [3].

The user will capture an image of his/her meal or snack both before and after eating. This reduces the burden of many aspects of recording for the users and reduces analysis for the researchers. However, in the event that a image of a food cannot be obtained, the system needs an alternative way of determining the food consumed.

In this paper we describe the status of our system development since last year [3]. We also designed and implemented an user interface for a mobile telephone food record which allows the user to record new food(s) or drink(s) by capturing the images before and after consumption. The user interface also allows a user to label each food item in the meal.

2. System Overview

In this section we review our two different configurations for deploying our dietary assessment system on a mobile device: a client-server configuration and a standalone configuration. We will also describe some of the approaches we are using for image analysis.

Client-Server Configuration—The idea here is to use the mobile device as a client where the user acquires an image of the food and sends the image along with metadata (e.g. location information, date, and time) to the server where the image is analyzed. This analysis includes food recognition and volume estimation. Once this process is completed, the client receives an image segmented, the name of the food items in the image and a volume estimate of the food consumed, a confirmation by the user is required at this point, in order to obtain more accuracy in the nutrient information consumed at the server side.

Currently, we are using a Nokia N800/N810 tablet as the mobile device and using 802.11g for the network connectivity in ad-hoc mode to the server. The overall flow is described In Figure 1. The first step is to send the image and metadata to the server for food identification and volume estimation (step 2 and 3), these results are sent back to the client where he/she confirms and/or adjusts this information (step 4). In step 5, the server receives the user confirmation and extracts the nutrient information using the FNDDS database (step 6), finally these results can be sent to the research community for further analysis (step 7). At this point we have implemented a first version of our system where we are sending the image that we want to analyze to the server, and then, the name of the food item and the segmented image is received, Figure 1. At the server side the nutrient information of the identified food is displayed.

Standalone Configuration—This approach consists of performing all the image analysis on the mobile device. This can be particular benefitial in those areas with poor network connectivity. However, one of the main disadvantages of this approach is the battery consumption in the mobile device. Currently, we are exploring strategies to perform parts of the image analysis in the mobile device. This configuration may also be needed to determine which part of the image analysis is performed in the mobile device and which part is done at the server side. For example, the user may not need to send the image to the server at all (useful in areas with poor connectivity). Instead, the mobile device can perform some parts of the image analysis and then send features and metadata needed by the server to complete the analysis. Another issue that needs to be taken into account is the microprocessor speed. State of the art mobile device microprocessors feature up to 500MHz-600MHz, which may significantly slow down the image analysis process.

2.1. Related Work

In this section, we provide a review of some of the previous image analysis and segmentation methods for automatic food recognition.

Jimenez et.al [4] reported an automatic fruit recognition system, which can recognize spherical fruits in different situations such as shadows, bright areas, occlusions and overlapping fruits. A three-dimensional scanner is used to scan the scene to generate five digital images. Two images represent the azimuth and elevation angles ($AZ(x, y)$ and $EL(x, y)$), the distance or range is included in $RANG(x, y)$, the attenuation is in $ATTE(x, y)$ and the reflectance image $REFL(x, y)$. The image analysis process uses the images obtained from the scanner to detect the position of the fruits by thresholding and clustering. The Circular Hough Transform is used to identify the center and radius of the fruits. This paper also gives a comprehensive study of the previous fruit detection work.

A robust algorithm to segment the food items from the background of color images is proposed in [5]. The image is converted to a high contrast grayscale image from an optimal linear combination of the RGB color components and then the image is segmented using a global threshold which is estimated by a statistical approach to minimize the intraclass variance. The segmented regions are subjected to a morphological process to remove small objects, to close the binary image by the dilation followed by erosion and to fill the holes in the segmented regions.

The work presented in [6] uses a stick growing and merging method to segment complex food images. The image is first pre-processed by an edge-preserving smoothing technique and a large number of horizontal lines ("sticks") are built in which pixels are homogeneous by traversing the initial image. The sticks ends correspond to edge points. Once the sticks are built, adjacent sticks are merged to form a sub region based on the stick-stick homogeneity criterion C_{2ssh}. Then the sub regions are merged if it satisfies a minimal sub region merging criterion. The regions with non-stick areas appear as noises or edges which cause less smooth boundary. Boundary modification step is used to reduce the degree of boundary roughness. This method is also successful in segmenting many food complex food images including pizza, apple, pork and potato.

In [7], a snake model is proposed which is a controlled continuity spline under the influence of image forces and external constraint forces. Energy functionals are needed to make the snakes useful for vision problems. They process three different energy functionals to detect features such as lines, edges and terminations. Edges in images are detected by using a simple energy functional. The snake is used to detect contours with large image gradients by using the edge functional. This paper also explains by using the edge functional they were able to segment food items such as pear and potato.

The work presented in [8,9] discusses the different set of features, namely color, shape and texture features, that can be used for image retrieval. The authors in [10] proposes to use color and texture features and clustering technique to segment the images. This paper also indicates that the image segmentation algorithm based on the color and texture features performed well in low-resolution and compressed images. In the paper by Dorin et.al [11], they were able to produce high quality edge image by extracting all the significant colors.

2.2. Image Analysis System

In [3], we described methods to automatically segment and identify food items from an image acquired from a mobile device. In this section we overview the preliminary methods described earlier in [3] and use this to introduce the work we have done in developing a user interface for our system.

Automatic identification of food items is not an easy problem in that some foods may not be identifiable from an image. For example, the type of milk in a cup (e.g., low fat or skim milk) may not be determined from the image alone. This will require other types of "side information" be available to the system either through how the food was packaged (e.g., an image of the milk carton) or through inputs (manual or audio) from the user.

A block diagram of our image analysis system as reported in [3] is shown in Figure 2. Our goal is to automatically determine the regions in the image where a particular food is located. This step is known as image segmentation. Once a food item is segmented, we can identify the food item base one its features, a step known as image classification.

Image Segmentation—Our image segmentation approach is a two step process. In the first step the image is converted to a grayscale image and then thresholded with a threshold of 127 to form a binary image. It was determined empirically that the pixel values in the binary image corresponding to the plate had values of 255. For segmenting the food items on the plate, the binary image was searched in 8-point connected neighbors for the pixel value 0. Connected segments less than 1000 pixels were ignored because they correspond to the tablecloth. In this step we used a fixed threshold. Thus, pixels corresponding to the food items might be considered as the plate. As a result, we need a second step of segmentation.

In the second step, the image is first converted to the YCbCr color space. Using the chrominance components, Cb and Cr, the mean value of the histogram corresponding to the plate was found. Pixel locations which were not segmented during the first step were compared with the mean value of the color space histogram of the plate to identify potential food items. These pixels were labeled differently from that of the plate. Then 8-point connected neighbors for the labeled pixels were searched to segment the food items.

Image Features—Two types of features were extracted from each segmented food region, namely color features and texture features. For color features, the average value of the pixel intensity along with the two color components were extracted. The color components were obtained by first converting the image to the YCbCr color space. For texture features, we used Gabor filters to measure local texture properties in the frequency domain. Several Gabor techniques for texture-segmentation applications can be found in [12–15]. Gabor filters describe properties related to the local power spectrum of a signal and have been used for texture features [14]. In our work, we used a Gabor filter-bank proposed in [12]. It is highly suitable for our use where the texture features are obtained by subjecting each image (or in our case each block) to a Gabor filtering operation in a window around each pixel and then estimate the mean and the standard deviation of the energy of the filtered image. In our implementation, we divided each segmented food item in to $N \times N$ non-overlapped blocks and used Gabor filters on each block. We used the following parameters: 4 scales (S=4), and 6 orientations (K=6).

Classification—Once the food items are segmented and their features are extracted, the next step is to identify the food items using statistical pattern recognition techniques [16,17]. For classification of the food item, we used a support vector machine (SVM) [18–20]. SVM is a technique used for data classification. A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one class label and several "attributes" (features). The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes.

The feature vectors used for the SVM contain 51 values, 48 texture features and 3 color features. The feature vectors for the training images (which contain only one food item in

the image) were extracted and a training model was generated using the SVM. A set of 17 plastic food images were used as training images and each food item was given an unique label to generate a training model for plastic food. And a set of 12 real food images were used as training images to generate a training model for real food. In our work we used the LIBSVM [21] a library for support vector machines.

3. Experimental Results

3.1. Image Database

A database of approximately 2800 images was created using the following devices:

- Canon PowerShot SD200
- Canon PowerShot S3
- Canon PowerShot SD870
- Nikon 3700
- Nikon 7600
- Panasonic DMC-FZ4
- HTC P4351, Windows Mobile Device

3.2. Food Replicas

The database contains images of food replicas and real food items. For the food replicas, images were acquired using specific conditions, such that the foods were placed on a white plate on a checker-board (black and white) patterned tablecloth. The tablecloth was used as a fiducial object for estimating the dimensions and area of the food item. The white plates were used to assist the segmentation of the food items. The training images used were taken with only one food item and the testing image contained 2 or 3 food items. The database consists of 50 food replica images, 17 images were used for training and 33 images were used for testing. The average classification results indicated a 93.745% (Table 1) accuracy when 17 images were used as training images and 14 images containing 32 food items were used as test images. Examples of correctly classified objects and misclassified objects are shown in Figure 3.

3.3. Real Food

Real food the images were acquired under specific conditions, such that the foods were placed on a black tablecloth with a checked calibration pattern. The training images used were taken with only one food item and the testing image was taken with different lighting conditions, different resolution and different orientation. The database consists of 256 real food images, 11 images were used for training and 245 images were used for testing. The average classification results indicated a 57.55% (Table 1) accuracy when 11 images were used as training images and 245 images containing 11 food items were used as test images. The classification accuracy for test images with similar lighting conditions as the training images were better than the test images with no or bad lighting conditions as shown in (Table 2). Examples of correctly classified objects and misclassified objects are shown in Figure 4.

4. User Interface

In this section, a review of user interface design goals is provided along with the structure and navigation of the interface we designed for the Mobile Phone Food Record (mpFR) sub-

system that will be integrated into our overall system (shown in Figure 1). The details of the usability tests performed with the mpFR are also discussed.

A graphical user interface (GUI) is a type of interface which allows users to interact with electronic devices such as computers, hand-held devices (MP3 players, portable media devices, and gaming consoles), household appliances and office equipment. Graphic interfaces, with the help of images, texts and menus, try to approximate the "picture" a user has of a certain problem [22]. The best known example is the desktop metaphor, where the screen shows a typical office desk, with files in folders, a note pad, and a waste paper basket. Often the actions of the user can be reduced to typing a few words and pointing to images and menus with the help of a mouse or simply a finger.

4.1. Goals

Designing the visual composition and temporal behavior of a GUI is an important part of software application programming. Its goal is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline known as usability. Techniques of user-centered design ensure that the visual language introduced in the design is well tailored to the tasks it must perform [23]. A GUI may be designed for the rigorous requirements of a vertical market. This is known as an "application specific graphical user interface." The latest mobile telphones and handheld game systems also employ application specific touch screen GUIs. But the majority of user interfaces is designed to meet one or more of the following goals [24], even if often the goals are not made explicit.

- **Easy to use**: a user who has little or no knowledge about the system, but knows enough about the subject domain, should be able to use the application without much training.

- **Easy to learn**: The ease of learning can be measured in various ways. One measure could be the time required to master a certain task, another could be the degree to which users feel they understand the system after some period of time.

- **Easy to maintain**: Many programs are created for extended use, which nearly always means that they have to be updated a few times. If the users themselves or another non-programmer has to do this, the interface has to provide some form of support.

- **Fast/minimal effort**: The number of keystrokes (mouse clicks, mouse travel distance) required for performing some task should be minimal. Note that this is not the same as the response time. The response time is a measure of how fast the system can respond to user events.

- **Flexible**: Some systems are meant to be used by a wide variety of users in many different situations.

- **Simple screen layout**: Desirable if the system must be easy to describe.

- **Portability and Internationalization**: Since the application and the interface often use very different computer resources, porting a system to another computer may very well require changes to the one and not the other. Internationalization means translating commands and messages to other languages, changing some icons, sorting order, time and date display, fonts, etc. Most if not all of these changes can in many cases be accommodated by changes in the interface.

Many of these goals are actually contradictory. It is not possible to create a single interface that is usable without training and that also pleases the user.

4.2. Goals of a Mobile Phone Food Record

The set of user goals to be achieved for the Mobile Phone Food Record are:

- Record a new food(s) or drink(s) by selecting the corresponding occasion

- Capture and confirm the image before consumption

- Capture and confirm the image after consumption

- Label the food(s) or drink(s) in the image

- Edit label of the food(s) or drink(s) in the image at a later time

- Users should be able to record food(s) or drink(s) at a later time in the following scenarios:

  – The user forgot to take the image of their meal

  – The user did not have the mobile device food record with him during the meal

  – When it was not possible to take image

The overall structure of the application is relatively simple, as shown in Figure 5 and Figure 6. There are three main elements in the application which can be selected from the main form.

The screen in Figure 7(a) enables the user to start the Mobile Phone Food Record application. The screen in Figure 7(b) displays three options namely record a new meal, reviewing the meals or to use the alternate method.

Record A New Meal—This application lets the user select the meal occasion and capture images of their meal before and after consumption. The user can select between breakfast, lunch, dinner and snack (Figure 7(c)). Once the meal occasion is selected, the user is reminded to use the camera in the landscape mode (Figure 7(d)), before the camera is displayed (Figure 8(a)). This ensures that both the before and after images are consistent. Once the user captures the image of the meal, then the screen in Figure 8(b) lets the user to review the image. The user can save the image or select the option to retake the image of the meal.

In the next screen the user can select next once the food is consumed. The screens in Figure 8(d), Figure 9(a) and Figure 9(b), allows the user to capture the image of the meal after consumption, similar to the procedure used for capturing before meals. Once the "after image" is saved a summary screen (Figure 9(c)) is displayed, which has both the before and after images. By selecting finish the main screen (Figure 7(a)) is displayed.

Review A Meal—The user can review all their meals that have been recorded. The meal that needs to be reviewed can be selected from the list of all the meals that's been recorded from the screen in Figure 9(d). Once the meal to be reviewed is selected, the before image of the meal is displayed (Figure 10(a)). Food items can be labeled by adding new markers and placing them on the food item to be labeled. The user can start labeling the food item by selecting label from the menu. The food label screen (Figure 10(b)) lets the user to search for the food items from a database, with the help of intellisense. Similarly all the food items can be labeled by adding the food markers. Once all the food items are labeled, the meal summary screen is displayed (Figure 10(d)). Reviewing the meals can also be used to edit the food labels.

Alternate Method—This application lets the user to record food(s) or drinks(s) when capturing the image of their meals was not possible or forgotten. The user can select the date when they had consumed the meal and enter the food item. The meal occasion also needs to be selected as show in Figure 11(a). Once all the food items are added and finish is select, the summary screen (Figure 11(b)) is displayed.

4.3. Usability Test of Mobile Phone Food Record

The user interface for the Mobile Phone Food Record sub-system described above was implemented on mobile telephone and tested to assess its usability or ease of use and to quantify error. The usability test is a technique to evaluate a product. The tests were performed under controlled conditions with known food amounts. Young people 11-18 years old (44 subjects from 11-14 years old and 36 subjects from 15-18 years old) were recruited and served foods of known gram weights. Meals and/or snacks consumed were recorded using the food record system under observation. The subjects were asked to capture the before and after images of their meal using the mpFR and a digital camera.

The Mobile Phone Food Record was iteratively refined based on the feedbacks from the usability test. The feedbacks was based on the qualitative information we captured during testing and questionnaires completed by the users. Some of the qualitative information we collected included individual opinions about Mobile Phone Food Record quality, perceptions about ease of use. Some of the findings based on the feedbacks are that approximately 10% of the subjects captured the before and after images with different orientations and 2% of the subjects did not capture either the before or after image due to technical errors (program error) or human error. Based on these findings, the mobile phone Food Record is being refined.

5. Conclusion

In this paper we described the current status of our development of a dietary assessment system mobile devices. We are planning on field test of the complete system in the Summer of 2009.

Acknowledgments

References

1. Ogden C, Flegal K, Carrol M, Johnson C. Prevalence and trends in overweight among us children and adolescents, 1999-2000. JAMA. October; 2002 288(14):1728–1732. [PubMed: 12365956]

2. Fagot-Campagna A, Saadinem J, Flegal K, Beckles G. Diabetes, impaired fasting glucose, and elevated hba1c in US adolescents: the third National Health and Nutrition Examination Survey. Diabetes Care. 2001; 24:834–837. [PubMed: 11347739]

3. Zhu, F.; Mariappan, A.; Kerr, D.; Boushey, C.; Lutes, K.; Ebert, D.; Delp, E. Technology-assisted dietary assessment. Proceedings of the IS&T/SPIE Conference on Computational Imaging VI; San Jose, CA. January 2008;

4. Jimenez A, Jain A, Ceres R, Pons J. Automatic fruit recognition: a survey and new results using range/attenuation images. Jr Patt Recog Society. 1999; 32:1719–1736.

5. Mery D, Pedreschi F. Segmentation of colour food images using a robust algorithm. Jr Food Engr. 2005; 66:353–360.

6. Sun DW, Du CJ. Segmentation of complex food images by stick growing and merging algorithm. Jr Food Engr. 2004; 61:17–26.

7. Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. Inter Jr Comp Vision. 1998; 1:321–331.

8. Rui Y, Huang TS. Image retrieval: Current techniques, promising directions, and open issues. Jr Vis Commn Img Repn. 1999; 10:39–62.

9. Smeulders AW, Worring M, Santini S, Gupta A, Jain R. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence. December; 2000 22(12):1349–1380.

10. Chen J, Pappas TN, Mojsilovic A, Rogowitz BE. Adaptive perceptual color-texture image segmentation. IEEE Transactions on Image Processing. October; 2005 10(10):1524–1536. [PubMed: 16238058]

11. Comaniciu, D.; Meer, P. Robust analysis of feature spaces: Color image segmentation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; Los Alamitos, CA, USA. 1997. p. 750-755.

12. Unsupervised texture segmentation using gabor filters. Pattern Recognition. 1991; 24(12):1167–1186.

13. Manjunath B, Ma WY. Texture features for browsing and retrieval of image data. IEEE Transactions On Pattern Analysis and Machine Intelligence. August; 1996 18(8):837–842.

14. Kruizinga, P.; Petkov, N.; Grigorescu, SE. Comparison of texture features based on gabor filters. Proceedings of the 10th International Conference on Image Analysis and Processing; Washington, DC, USA. September 1999; p. 142

15. Liu C, Wechsler H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Transactions on Image Processing. April; 2002 11(4):467–476. [PubMed: 18244647]

16. Fukunaga, K. Introduction to Statistical Pattern Recognition. San Diego, Ca: Academic Press; 1990.

17. Duta, R.; Hart, P.; Stork, D. Pattern Classification. New York, NY: Wiley-Interscience; 2000.

18. Burges CJC. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery. 1998; 2(2):121–167.

19. Muller K, Mika S, Ratsch G, Tsuda K, Scholkopf B. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks. March; 2001 12(2):181–201. [PubMed: 18244377]

20. Cristianini, N.; Taylor, J. An introduction to support vector machines. Cambridge: Cambridge University Press; 2000.

21. Chang, CC.; Lin, CJ. LIBSVM: a library for support vector machines. 2001. software available at http://www.csie.ntu.edu.tw/cjlin/libsvm

22. Rogers, Y.; Sharp, H.; Preece, J. Interactive Design: Beyond Human-Computer Interaction. Wiley; 2007.

23. Gong, J.; Taraswewich, P. Guidelines for handheld mobile device interface design. Proceedings of the Decision Sciences Institute Annual Meeting; 2004.

24. Chong PHJ, So PL, Shum P, Li XJ, Goyal D. Design and implementation of user interface for mobile devices. IEEE Transactions on Consumer Electronics. November; 2004 50(4):1156–1161.

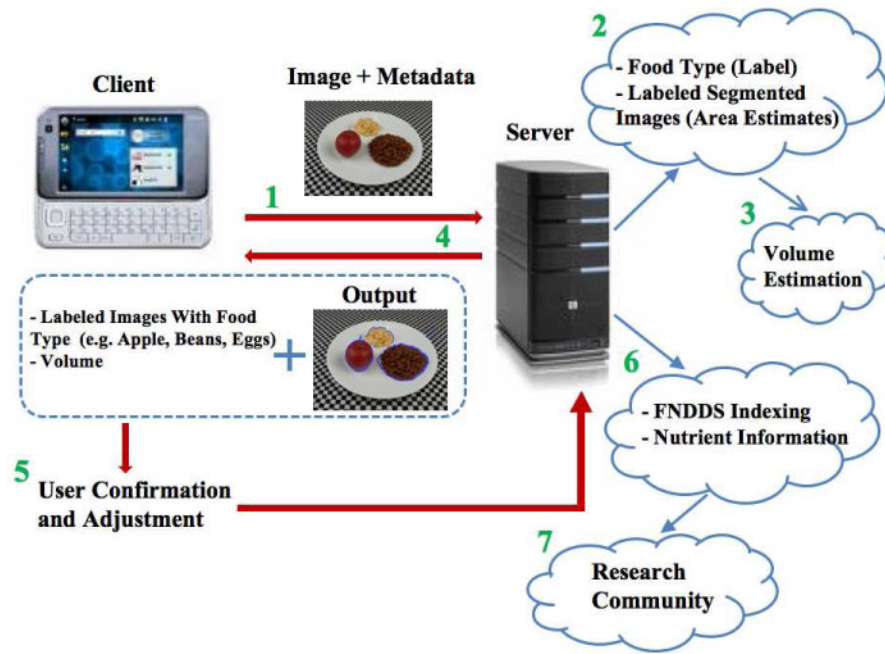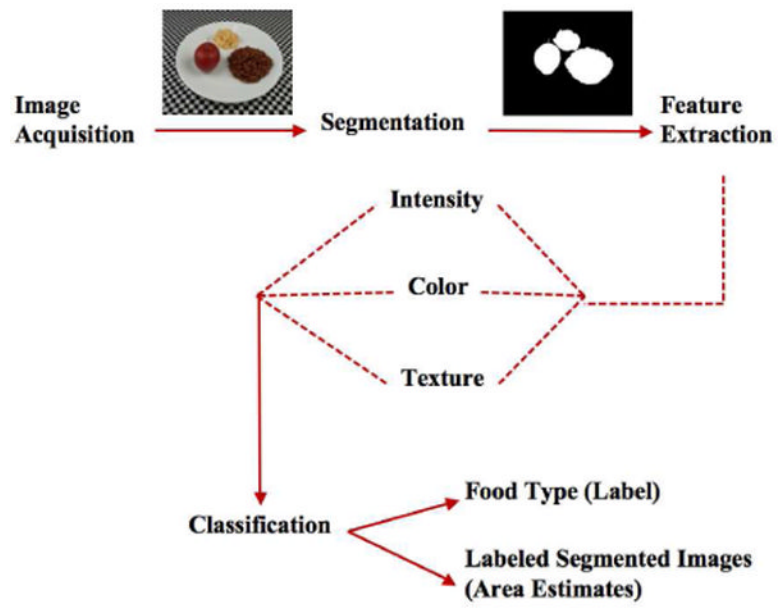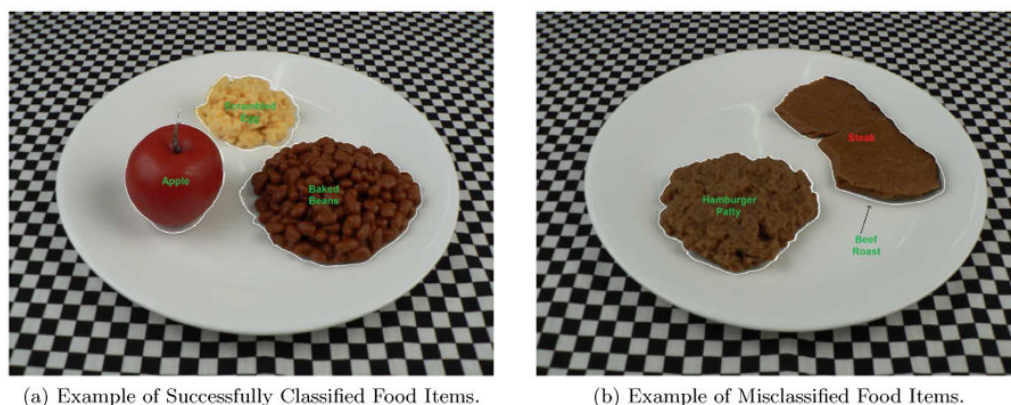**Figure 1.**
Dietary Assessment System Architecture.

Figure 2.
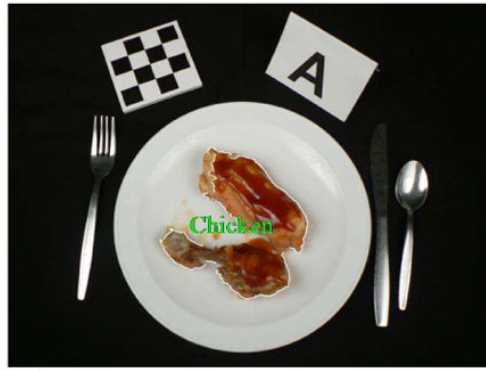Image Analysis System.
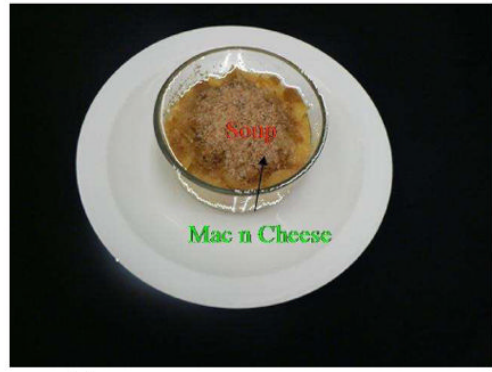
(a) Example of Successfully Classified Food Items.

(b) Example of Misclassified Food Items.

Figure 3.
Example of classified food items (a) All food items are successfully classified using a SVM, (b) Some food items are misclassified by the SVM, i.e. beef roast is misclassified as steak.
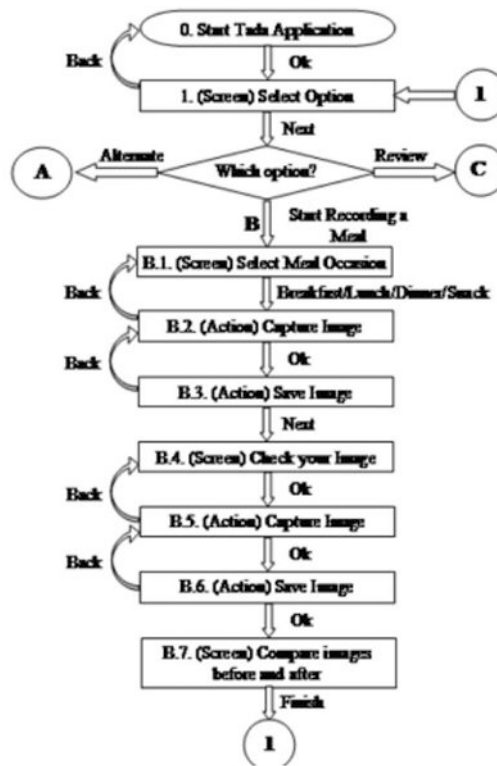
(a) Example of Successfully Classified Food Items.

(b) Example of Misclassified Food Items.

Figure 4.
Example of classified food items (a) Food items are successfully classified using a SVM, (b) Food items are misclassified by the SVM, i.e. Mac and Cheese is misclassified as soup.

Figure 5.
Structure for Recording a New Meal.
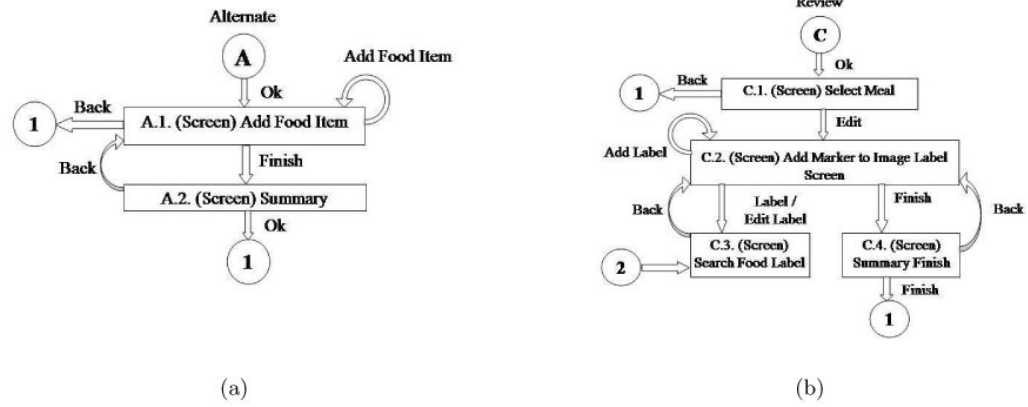
Figure 6.
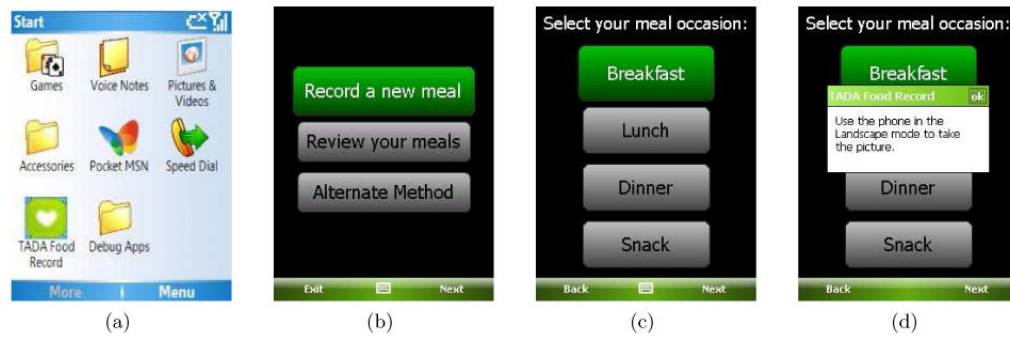Structure of the User Interface (a) Structure for Alternate Method (b) Structure for Reviewing a Meal.

Figure 7.
Screen Shots (a) Start Mobile Phone Food Record. (b) Select Action (c) Select the Meal Occasion (d) Information Box to remind the user to capture the image in landscape mode.
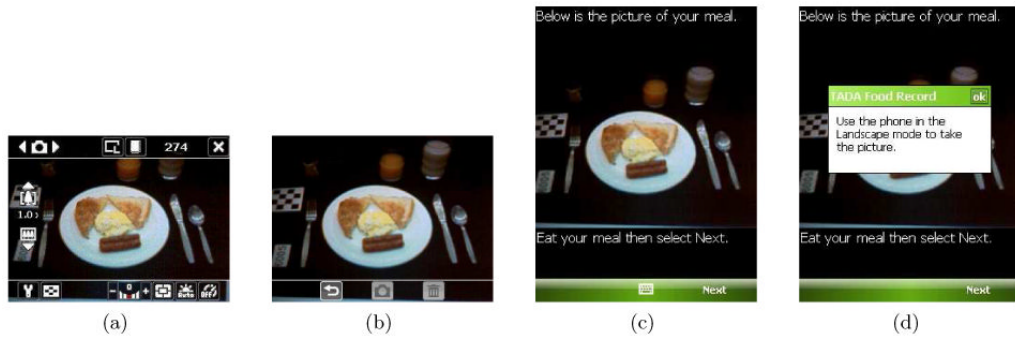
Figure 8.
Screen shots - (a) Show Camera and Capture Before Image (b) Review and Confirm Before
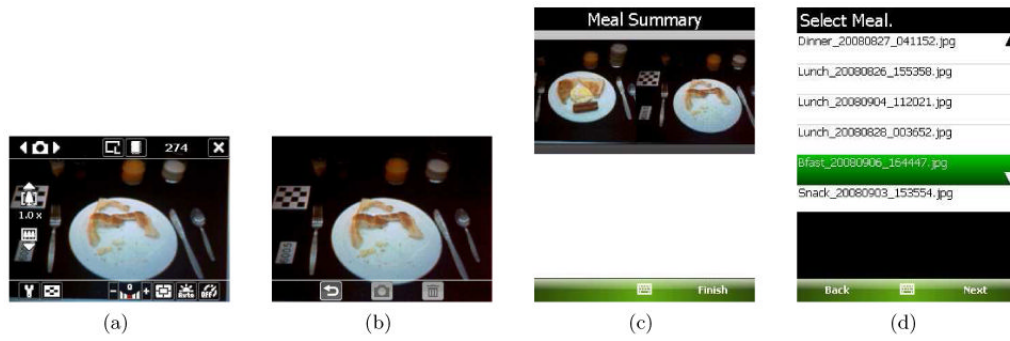Image (d) Information Box to remind the user to capture the image

Figure 9.
Screen shots - (a) Show Camera and Capture After Image (b) Review and Confirm After Image (c) Summary. (d) List of all meals recorded
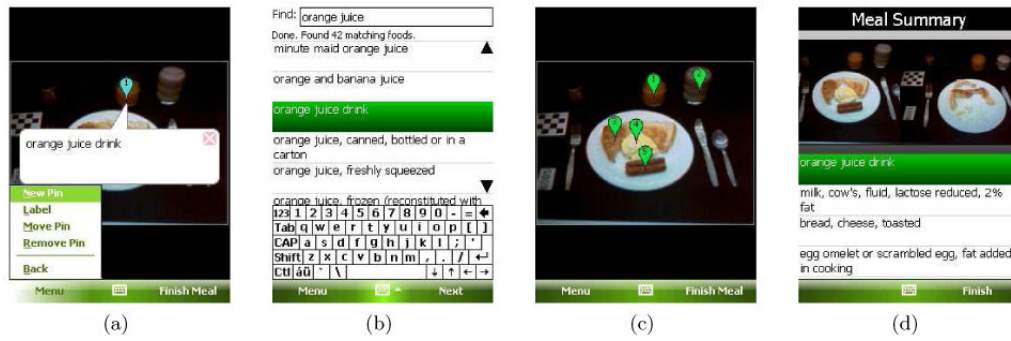
Figure 10.
Screen shots - (a) Marker added to the Image Label Screen (b) Search Food Label (c) All food items labeled (d) Summary of Food Items Labeled.
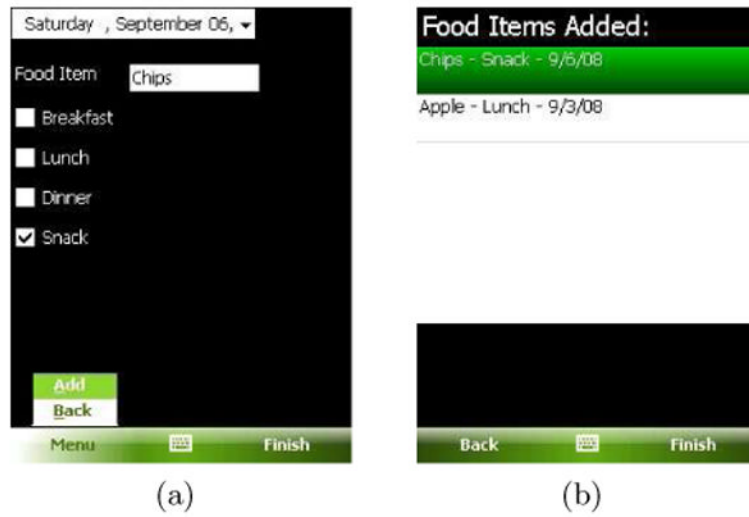
Figure 11.
Screen shots - (a) Add Food Item (b) Summary of Food Items Added.

Table 1

Classification Accuracy.

Food Type	Training Images	Testing Images	Classification Accuracy
Plastic Food	17	33	93.75%
Real Food	11	245	57.55%

Table 2

Classification Accuracy - Real Food.

Lighting Conditions	Training Images	Testing Images	Classification Accuracy
Good Lighting	11	116	70.68%
Bad / No Lighting	11	129	34.91%