Feasibility of Food Recognition and Calorie Estimation of Fast Food and Healthy Meals Available in the Philippines

K. Dy, J. Ligan and M. Cabatuan
De La Salle University.
melvin.cabatuan@dlsu.edu.ph

***Abstract*—This paper presents the design and development of a food recognition smartphone application which can also display the estimated calorie/s of the food itself. It is intended for people who would like to monitor their diet through food calorie intake measurement (i.e. user's daily calorie intake record). It is equipped with a food database consisting of typical fruits and vegetables commonly found in the Philippines. As part of the study, it also includes some of the meals in food chains (i.e. McDonald's, and The Healthy Corner) found in the Philippines where the calorie information is readily available. The result shows 82.86 % accuracy for the top-1 category, and 99.29 % for the top-5 category. The algorithm being used in this project is Artificial Neural Network (ANN) wherein the recognition process must properly be achieved. Furthermore, the aforementioned database is supported by TensorFlow which is an open-source software library for Machine Intelligence.**

***Index Terms*—Artificial Neural Network; Calorie; Food Recognition; TensorFlow.**

I. INTRODUCTION

In 2015, the WHO projected that 24 % of men and 44 % of women in the Philippines will suffer from obesity [1]. Many factors influence our body weight some of which are our genes, our physical activity, our environment, and especially our food intake [2]. But at its most basic, obesity is caused when someone regularly takes in more calories than needed. In the Philippines, restaurants are not required by the government to post the nutritional facts of the food in their menus. This hinders the average consumer from maintaining his/her daily diet. On October 16, 2008, the late Senator Miriam Defensor Santiago filed the Senate Bill No. 2682 otherwise known as The Nutritional Content in Menu Boards Act of 2008 which will make the posting of nutritional content information in menu boards of food establishments mandatory. More than six years have passed since the bill was filed but until now its approval is still pending in the committee [3]. This study aims to develop a mobile application that can provide its user's nutritional information on food commonly found in the Philippines.

II. LITERATURE REVIEW

A. Calorie Estimation from Fast Food Images

In this paper, the proponents applied food recognition with calorie estimation on fast foods using Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) algorithms. They also compared different features such as Histogram of Gradients (HOG) and Local Binary Pattern (LBP) [4]. Thus, it aims to predict the calorie content based on a given image.

When it comes to method, fine tuning has been used on a pre-trained CNN model. Backpropagation is also used for the optimization process which involves the Chain Rule in taking partial derivatives of functions. Results show that LBP is more accurate than HOG in terms of classification. That is, texture information is a bit better than shape information respectively based on its confusion matrices [4]. A linear SVM is implemented throughout the study to address concerns regarding overfitting. The accuracy percentage ranges from 72.53 % to as high as 97.17 %. As part of their future work, they intend to increase the dataset, and to explore other optimization methods as well.

B. Food Image Recognition with Deep Convolutional Features

In this paper, the researchers proposed the application of Deep Convolutional Neural Network (DCNN) with Fisher Vector (FV) on food recognition [5]. They have observed through experiments that combining these algorithms; and, by integrating it with conventional hand-crafted image features, FV with HoG and color patches, food recognition accuracy can boost greatly.

For its method, they have extracted the network signals just before the last layer of the pre-trained DCNN as a DCNN feature vector. Conventional features have also been extracted using RootHoG patches and color patches. Then, these features are coded into FV representation with three levels of Spatial Pyramid. FV is known as a state-of-the-art coding method. Moreover, they used one-vs-rest linear classifiers for 100-class food classification. As a result, they have achieved a 72.26 % for top-1 accuracy, and 92 % for the top-5 accuracy using a 100-class food dataset known as UEC-FOOD100 [5]. Lastly, they intend to implement the proposed framework on mobile devices as part of their future work.

III. OBJECTIVES

A. General Objective(s)

To develop an android application that can identify and provide the calories in food commonly found in the Philippines.

B. Specific Objectives

1. To achieve at least 93 % in the top-5 accuracy and at least 74 % in the top-1 accuracy when identifying food;

2. To develop a program using OpenCV that can identify and provide the caloric information of food;
3. To develop a database which stores the user's calorie intake;
4. To ensure that the android application is responsive and easy to use.

IV. Research Methodology

The method of the research is done in such a way that it starts with the image taken by the user being uploaded to the server for image processing. In the server, the user's image is segmented to single out all the different food items in the image. The items' sizes are then measured and recorded. The items are then run through TensorFlow one by one to classify their identity. The output of the classification is then recorded. The output from the size measurement, image segmentation, and image classification is then sent back to the Android application. In the Android application, the results are shown to the user and the user is given an estimate of the amount of calories of the food item based on its size and the available information in the application's food database. The user can then choose to add the item to his/her food history. Figure 1 shows a conceptual diagram of the usage scenario of the Android application.

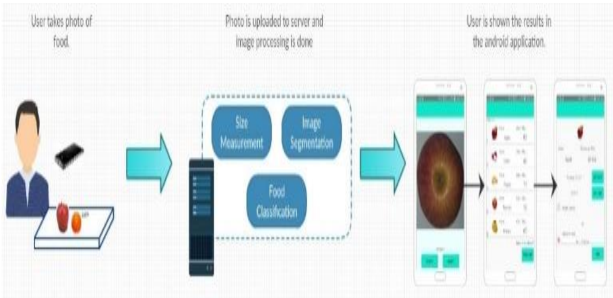

Figure 1: Conceptual diagram of the usage of the Android application

A. *Image Segmentation*

This study uses OpenCV to segment and measure the size of each object of interest in an input image. Figure 1 below shows the method used to detect the multiple objects of importance in an image. First, the image is converted into grayscale. Edge detection is then applied along with erosion and dilation to close the gaps between the edges of the objects. Figure 2 shows a block diagram of the image segmentation, size measurement, and food classification of the developed system. It can be seen that each object in the image is given a clear outline. Find contours is then implemented to single out each object in the image. After applying size measurement, each object will then be extracted and be put into its own image container.

B. *Size Measurement*

A prerequisite for the size measurement method used for this study is to have a reference object of known size to be at the left most part of the image. Thus, it has chosen to use a one peso (i.e. Philippine Peso, Php) coin as its reference object because it is a very common and accessible item. By knowing the actual size of the object in the image, the object's pixel to inches ratio can be calculated. Suppose that a one peso coin has a known width of 0.9 inches, and in the input image the coin's width covers 170 pixels.

$$PixelsPerMetric = \frac{170pixels}{0.9inches} = \frac{189pixels}{inches} \quad (1)$$

It can then be concluded that there are 0.9 inches per every 170 pixels in the image. By knowing this ratio, sizes of the other objects can also be calculated through calibration as shown in Equation (1).

The images that were retrieved from the segmentation done earlier are sized one by one from left to right. The first step is to encapsulate the minimum contour area of each object using bounding boxes. Points are then assigned to the top-left, top-right, bottom-right, bottom-left of the bounding box. The midpoints of the top-left and top right, bottom-left and bottom-right, top-left and bottom-left, top-right and bottom-right are then obtained. The Euclidean distance between the top and bottom, left and right midpoints are then calculated, and by doing so will give the height and the width of the object. Figure 3 shows an example of the bounding box with the calculated height and width of the object.

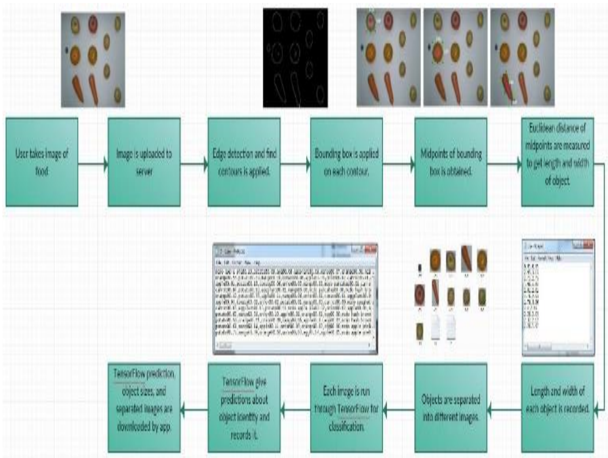

Figure 2: Block diagram of the image processing done by the developed system

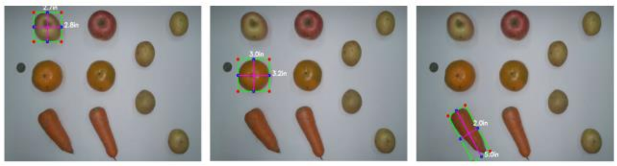

Figure 3: Object size measurement using bounding boxes

C. *Calorie Estimation*

Using the available information provided by the United States Department of Agriculture (USDA) Food Composition Database, a food item can be classified into three categories: small, medium, or large; based on its diameter or length. The recorded size measurements of the different food items are compared to the database and a calorie amount is suggested based on the objects category. If the category size of an object is not available, alternative ways to estimate its calorie are given. The calorie amount could be calculated by measuring the weight of the item in grams and comparing it to the amount of calories per 100 g of the given item. A serving size based calorie amount can also be obtained by using a specific serving size with its corresponding calorie amount. The information for both the calorie per 100 g and the serving size calorie have also been obtained using the available data from the USDA food composition database.

D. Image Classification using TensorFlow

The first step in using TensorFlow for image classification is to collect a database of images separated into folders of their own unique identity or category. Table 1 shows the total number of images in each food category. As a whole, the food database has 28 categories available.

For its purpose, the database will be used to retrain TensorFlow's Neural Network Inception-v3. By retraining TensorFlow it allows for the creation of a neural network that classifies images into custom categories defined by the developer. Before the images are used for retraining, TensorFlow divides the images of each category into 3 parts. 80 % of the images will be used for retraining, 10 % of the images will be used for validation accuracy, and the remaining 10 % is used for test accuracy. To test the accuracy of the neural network while being retrained, the validation set of images is fed to the training neural network to test for over-fitting or to test if the neural network's accuracy is being influenced by noise from the images. While retraining, TensorFlow sets aside the neural network training iteration with the highest accuracy. After training, the selected neural network iteration is then tested using the test set of images to get the real-world performance of the neural network. The test involving the test set of images is called the real-world performance, because the images included had no influence whatsoever during the retraining process of the neural network.

E. Additional Filipino Food Items

As part of future work, the proponents may extend the food database for recognition or detection with additional Filipino food items. Table 2 has a total of 51 categories.

Table 1
Tally of the Number of Images in the Database per Category

Food Category	No. of Images	Food Category	No. of Images	Food Category	No. of Images
Adobo	110	Egg Fried	209	Mcdo Fries	140
Apple	216	Eggplant	202	Mcdo Hash Brown	101
Banana	214	Garlic	219	Mcdo Spaghetti	106
Broccoli	221	Hotdog	101	Onion	291
Brown Rice	200	Liempo	101	Orange	246
Carrot	269	Mango	200	Potato	285
Chicken Breast	101	Mcdo Apple Pie	107	Rice	249
Corn	236	Mcdo Big Mac	101	Tomato	278
Cucumber	205	Mcdo Chicken	102		
Egg	194	Mcdo Chicken Nuggets	103		
				Total	5107

Table 2
Additional Filipino Food Items

Food Category	No. of Images	Food Category	No. of Images	Food Category	No. of Images
Adobo	110	Dinuguan at Puto	44	Pansit Palabok	95
Arroz Caldo	125	Empanada	92	Pastillas de Leche	88
Bagnet	184	Ensaymada	100	Pichi pichi	83
Balut	201	Fish Kinilaw	104	Pinakbet	105
Barbecue	83	Halo halo	96	Puto Bumbong	94
Betute	138	Kare kare	93	Rellenong Alimango	69
Bibingka	196	Kuhol sa Gata	77	Sinigang	165
Bicol Express	212	Kutsinta	99	Sinugno	42
Biko	169	Laing	70	Sisig	166
Binignit	160	Leche Flan	67	Suman Manga	87
Buko Pie	313	Lechon	114	Tapa	41
Bulalo	234	Liempo	127	Tinapa	75
Camaro Rebosado	90	Longaniza	106	Tinola	89
Champorado	100	Lumpiang Ubod	97	Tsokolate	67
Chicharon	85	Maja Blanca	100	Tuna Panga	119
Chicken Inasal Grilled	204	Pandesal	100	Turon	91
Crispy Pata	94	Pansit Habhab	83	Ube Jam	93
				Total	5836

F. Calorie Estimation using Volume

This part of the application will use a density database provided by "aqua-calc" whose source is the USDA National Nutrient Database for Standard Reference. By knowing the density and volume of the food item, its weight can be derived. The obtained weight can then be used to get a calorie estimate by multiplying it with the amount of calories of the item per 100 grams, as shown in Equations (2) and (3).

$$weight = volume\ (in^3) * density\ \left(\frac{g}{in^3}\right) \tag{2}$$

$$calories = \big(weight\ (g)\big) * \frac{calories_{per\ 100\ g}\ (kcal)}{100\ grams} \tag{3}$$

By using the length and width obtained from the size estimation section, the application can estimate the volume of a food item with the help of the user and the one peso coin. First, the application's food database is checked and if information about the density of the food item is available, the height parameters in the application, which were invisible, will be made visible. The user will then be able to choose from two methods to enter the height of the food item. The first method is straightforward, and will ask the user directly the height of the food item in inches. The second method, on the other hand, will make use of an estimate of the user using the one peso coin. The one peso coin will be placed standing in parallel beside the food item. The user will then estimate how many one peso coins the height of the food item is (e.g. 0.8/1/1.2). This one peso height estimate will then be used to get the estimated height of the food item by multiplying it to the known size of the one peso coin. By knowing the height of the item its volume can be obtained and from the volume, by using the equation above, its calories. This method requires the food item being measured to be in as much as a rectangular cuboid shape as possible to give the most approximate estimations. Figure 4 below shows the height parameters of the application, as well as the one peso height estimation method.

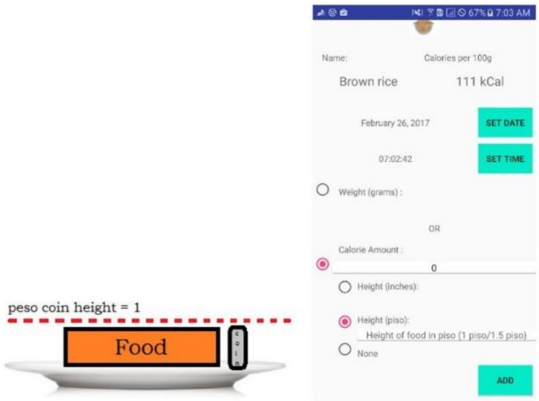

Figure 4: Height estimation using peso coin (left), and application height parameters (right)

V. THEORY

A. Image Processing

Image processing is a system in which the images are treated as two dimensional signals while applying already set signal processing methods to them. The system includes three basic steps, namely: importing, analyzing and manipulating of images, and finally the output based on image analysis. The purpose of image processing is divided into five groups: visualization, image sharpening and restoration, image retrieval, measurement of pattern, and image recognition [6].

As quoted, "machine learning is a type of Artificial Intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data [7]." The renewed enthusiasm to towards machine learning arises from the desire to produce smart actions in real time through automated model building [8].

Moreover, Computer Vision (CV) is related to machine learning in such a way that it acts as a bridge between the subject / object (e.g. image, video, etc.) and the interpreter (e.g. user, photographer, etc.). With the help of a sensing device like the eyes of the interpreter or the camera, CV processes the image through the computer and gives the output to the interpreter [9].

B. Artificial Intelligence

According to R. Sternberg, "intelligence is the cognitive ability of an individual to learn from experience, to reason well, to remember important information, and to cope with the demands of daily living." In particular, as B. Raphael puts it, artificial intelligence is the science of making machines do things that would require intelligence if done by man. The primary goal of AI is to create computer software and/or hardware systems that exhibit thinking comparable to that of humans; in other words, to display characteristics usually associated with human intelligence [10].

There are two approaches in this field of study, namely: weak AI and strong AI. Associated with the Massachusetts Institute of Technology, weak AI views any system that exhibits intelligent behavior as an example of AI. It does not matter whether the artifact performs its task in the same way humans do. Its sole criterion is that the program performs correctly. Represented by Carnegie-Mellon University, strong AI is primarily concerned with biological plausibility. That is, when an artifact exhibits intelligent behavior, its performance should be based upon the same methodologies used by humans. Proponents of strong AI are concerned with the structure of the systems they build [10].

C. Database Systems

Generally speaking, a database is a collection of data. In this context, data are known facts that can be recorded and that have implicit meaning. To be more specific, database can be defined according to these implied properties as quoted [11]:

- A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

Thus, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. Furthermore, a database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

D. Food Calorie Measurement

When it comes to calorie measurement, the word itself must first be defined. A calorie (i.e. abbreviated as cal) is a unit that is used to measure energy. One kilocalorie (i.e. abbreviated as kcal) which is equivalent to 1000 calories is the amount of energy needed to raise the temperature of one kilogram of water by one degree Celsius. In addition, one kcal is equal to 4.184 kj (i.e. kilojoules). There are basically two methods of measuring calorie. The original method which is usually done in laboratory experiments measures this energy directly by using an apparatus called a bomb calorimeter. It completely burns the food inside by the water surrounding the instrument, and the resulting rise in water temperature is measured correspondingly [12]. Another easier method is by using a table of reference values which is known as the Food Exchange List (FEL). An FEL measures the amount of carbohydrate, protein, fat and calories found in a food category in terms of the weight in grams of the food per serving [13].

VI. Testing and Analysis

A. Android Application Graphical User Interface

Figure 5 below depicts actual tests on the Graphical User Interface (GUI) prototype with its corresponding results (e.g. food recognition of a real apple). First, the image on the left show the gallery view where the user can accept or reject an entry. Second, the image on the center shows the Top Five results of TensorFlow. Finally, the image on the right shows the add food page with the calorie estimation result.

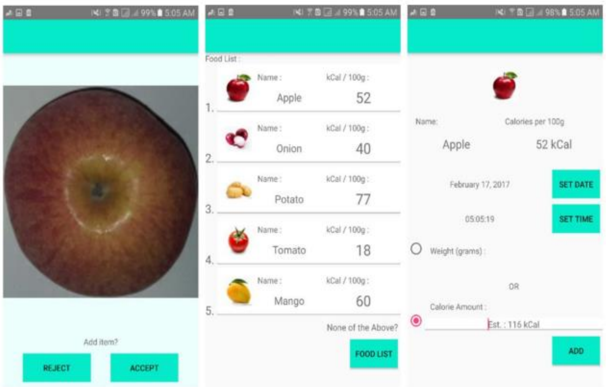

Figure 5: Screenshots of the gallery view (left), TensorFlow result (center), and add food page (right)

B. TensorFlow Training Accuracy

To evaluate the accuracy of the application's food classification, images of food items that were not included in the training were tested and fed through TensorFlow. The number of times TensorFlow correctly identifies the food item in the top one prediction has been recorded, as well as the number of times TensorFlow correctly included the food item in its top five predictions.

To obtain its confusion matrix, the images that were used during the retraining of TensorFlow were used again to retest the accuracy of the system. All images used in the training set, validation set, and test set were fed as inputs to TensorFlow and the top one predictions were recorded. Based on analysis, it can be seen that the TensorFlow classifier has been trained to achieve a very high classification accuracy. The TensorFlow Model is tested using 5107 images, and managed to give the correct top one prediction for 5022 images. Having only made 85 misclassifications out of 5107 images, the retrained TensorFlow model has been able to achieve a 98.34 % classification accuracy while classifying its training, validation, and test images.

C. TensorFlow Real-world Performance Accuracy

Figure 6 shows an example of the images that have been taken to test the real-world accuracy of the developed system. The application is able to separate all of the objects in the images, and run them one by one through the retrained TensorFlow classifier.

Overall, the system is tested with 280 images. Out of these 280 images, the system has been able to predict 232 items correctly as its top one prediction giving it a top one prediction accuracy of 82.86 %. On the other hand, two predictions out of the 280 that were made did not even include the item in the top five predictions thereby giving the system a 99.29 % top five prediction accuracy.

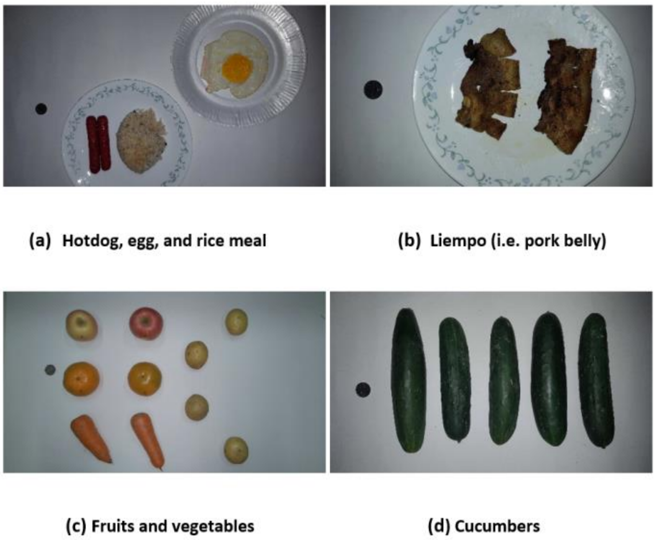

Figure 6: Sample real world image inputs

D. Average Processing Time

Table 3 shows the average processing time of the application from when the user takes a photo of the food until the user is presented the results of food classification and size estimation processes. The processing time of the application is affected by the number of unique food objects in the image. This is because each object is run through the TensorFlow classifier separately. By examining Table 3, it has been found that the average processing time per object in the image is 3.64 seconds. The download and upload speeds of the internet connection while testing is also taken into consideration, as shown in the same table.

Table 3
Average Processing Time of the Application's Image Processing and Size Estimation

Parameters	Value
Avg time per item (seconds)	3.64
Avg image size (kb)	594.47
DL speed (speedtest)	3.42 Mbps
UP speed (speedtest)	0.68 Mbps
Ping (speedtest)	29 ms

VII. Results and Discussions

A. Size Measurement Accuracy

To evaluate the accuracy of the application's calorie estimation, items with known food size categories have been tested on the application itself. The items are first measured manually, and placed into their respective size categories (small, medium, large). The items are re-evaluated using the application. The results of the size category placement obtained manually are then compared to the results of the application's size category placement.

The results of this section contain the comparison of the manually measured size of the item with the size estimated by OpenCV. The only items that have been included in these tests are items whose category size (small, medium, large) information was available. All information used to classify a food item to its respective size category are obtained from the USDA Food Composition Database. Five items from each food category included in this section are used to test for the size measurement accuracy.

Table 4
Food Size Measurement

Food Category	Measured Size (in)	Estimated Size (in)(Trial 1)	Estimated Size (in)(Trial 2)	Relative Error (%)
Apple	3.154	3.28	3.208	4.024
Banana	5.662	5.758	5.862	2.584
Carrot	5.63	5.588	5.672	2.012
Corn	7.394	7.546	7.5	1.944
Cucumber	8.094	8.334	8.492	3.91
Orange	3.114	3.218	3.124	4.346
Potato	3.986	3.742	3.868	4.438
Tomato	2.202	2.422	2.352	8.536
			Average Relative Error	3.97425

Table 4 contains the test results obtained when the measured and estimated sizes of the food items are compared. Eight food categories specifically apples, bananas, carrots, corn, cucumbers, oranges, potatoes, and tomatoes have been tested. These eight food categories are the only items included in the database whose size category information is available. Five items from each category have been tested twice using the Android application. Then, the estimated size and predicted size category are recorded. The average of the sizes obtained from the two trials are then compared to the measured size of the item, and the relative error in size measurement and estimation has been obtained. The tables also show the accuracy the system has in predicting the correct size category of the item.

Overall the system only made an error of more than 10 % from one item out of a total of forty and was able to achieve an average relative error rate of 3.97 %.

The accuracy of the system to place an item into its correct size category can also be obtained using the tables above. It can be seen that out of the 80 tests done on the system, it has been able to correctly place the item in its right size category 73 times. This gives the system a 91.25 % accuracy when predicting the size category of an item.

B. Calorie Comparison: Weighed versus Estimated

Figure 7 shows the data that compares the measured calorie values of the items with the estimated calorie values. This section uses the same five items per category from the previous section which showed the size estimation accuracy. The measured calorie values have been obtained by weighing the object, and multiplying this obtained weight with an item's amount of calories per 100 grams, as illustrated in Equation (4).

$$MeasuredCalorieValues = ItemWeight\ x\ \frac{CalorieAmount}{100\ grams} \tag{4}$$

The estimated calories presented in this section have been obtained by using each item's predicted size category and giving the corresponding amount of calories for that specific size category. The information about the amount of calories of each item's size category was obtained from the USDA food composition database.

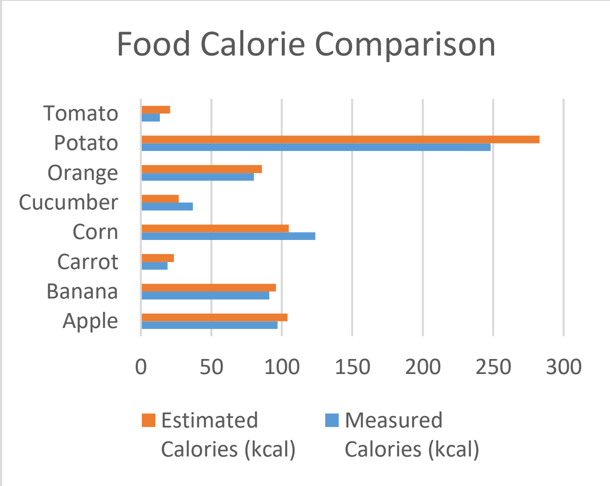

Figure 7: Food calorie comparison in bar graph

Table 5
Food Calorie Comparison in Tabular Form

Food Category	Weight (g)	Measured Calories (kcal)	Estimated Calories (kcal)	Relative Error (%)
Apple	187.2	97	104	7.332
Banana	103	91.2	96	7.4
Carrot	47.8	19	23.4	25.984
Corn	144.8	123.8	105	17.91
Cucumber	244.8	36.8	27	37.83
Orange	171.6	80.2	86	6.748
Potato	323.2	248.2	283	12.302
Tomato	78	13.4	20.8	35.344
			Average Relative Error	18.8563

By analyzing Table 5, the amount of error in the calories estimated by the system can be seen. Overall the average error between the measured and estimated calories obtained from testing these 8 food categories was found out to be 18.8 %. This means that it is possible for the application to overestimate or underestimate its calorie prediction of a food item by 18.8 %. The results are quite promising given that calorie intake estimation errors of more than 50 % have been observed from self-reported energy intake studies based on an article written in The New England Journal of Medicine [14].

C. Accuracy of Calorie Estimation using Volume

This section showcases the comparison of the measured calories of the food item versus its estimated calories which were obtained by using the estimated volume of the food item. The estimated calories in this section were obtained using the one peso height estimation method discussed in the methodology. By knowing the volume of the food item, its calories can be obtained by using the density database provided by "aqua-calc" whose source is the USDA National Nutrient Database for Standard Reference. The only items included in this test were the food items whose density information is available.

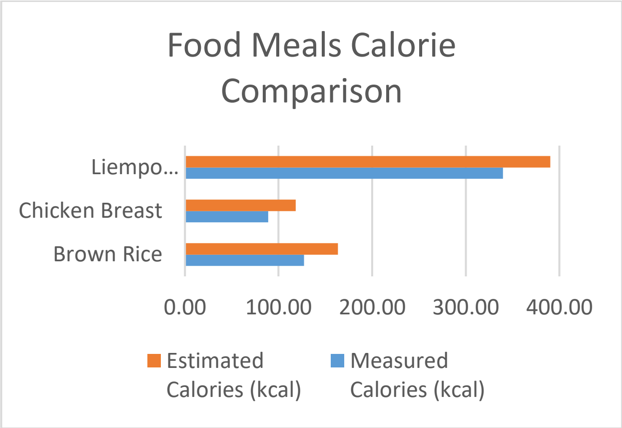

Figure 8: Food meals calorie comparison in bar graph

Table 6
Food Meals Calorie Comparison in Tabular Form

Food Category	Weight (g)	Measured Calories (kcal)	Estimated Calories (kcal)	Relative Error (%)
Brown Rice	114.7	127.33	163.33	21.39
Chicken Breast	54.33	89	118.33	24.34
Liempo (i.e. Pork Belly)	65.67	339.67	390.33	12.23
			Average Relative Error	19.32

Three food categories have been used for this test, namely: Brown Rice, Chicken Breast, and Liempo (i.e. Pork Belly). Three food portions from each category have been weighed, and the corresponding calories are measured. The food portions are then tested using the Android application to obtain their estimated calories.

By analyzing Table 6, the calorie estimation error of the Android application can be observed. Overall the average calorie estimation error of the system using volume was found to be 19.32 %. Considering that Google's research into calorie estimation using images, Im2Calories [15], is reported to get errors of more than 20 %, and that self-reported calorie intakes make errors of 50 % or more [14] means that the accuracy obtained is competitive.

D. Segmentation Accuracy

In this section, the accuracy of the segmentation process of the system is measured. To measure the accuracy, 100 random samples of actual testing were used. A segmentation is successful if all of the objects inside the image were able to be separated into their own image containers. A segmentation is deemed a failure if not all images were detected and segmented by the system.

Table 7
Segmentation Accuracy of the System

Images Segmented	Successful Segmentations	Relative Error (%)
100	93	7

By analyzing Table 7, it can be seen that out of the 100 images that went through segmentation 93 of the images were able to be segmented successfully. This means that the image segmentation of the system might make an error 7 % of the time, and that the overall accuracy of the system is 93 %. The left image in Figure 9 shows a successful segmentation process where all items were separated into different images. On the other hand, the right image of the same figure shows an unsuccessful segmentation process where an item has not been detected successfully.

Figure 9: Successful segmentation (left), and failed segmentation (right)

VIII. CONCLUSION

In this research, image recognition has been applied on actual food items. It is made possible through an Android application GUI prototype intended for food identification and calorie estimation. In order to make the application more intelligent, computer vision (i.e. OpenCV) and algorithms (e.g. ANN, edge detection, etc.) are used in the process. Moreover, the server side of the application contains databases where collection of food images and nutritional information (i.e. calories) are stored. It also requires a good internet connection to transmit data from the smartphone to the server, then vice-versa. The high accuracy objectives are reached through series of tests and data gathering. Confusion matrices have been created to tally the results. Food recognition accuracy yields a value of 82.86 % for the top-1 category, while a value of 99.29 % for the top-5 category. Lastly, calorie comparison between the measured (i.e. the ground-truth table) and estimated (i.e. based on the calorie estimation function of the application) values only give 18.8 % total relative error on the average as computed.

There are several prospects related in this research that may be extended for further studies. Thus, the suggested topics are listed as follows:

1. Volume estimation of food items using generic 3D models of the specific food category.
2. A novel way to easily extract the volume of a scalable 3D model.
3. Finding the most accurate method when classifying images.

REFERENCES

[1] World Health Organization, "The Impact of Chronic Disease in the Philippines," 2010. [Online]. Available: http://www.who.int/chp/chronic_disease_report/philippines.pdf

[2] Harvard T. H. Chan: School of Public Health, "Obesity Causes," [Online]. Available: http://www.hsph.harvard.edu/obesity-prevention-source/obesity-causes.

[3] M. Santiago, "Nutritional Content In Menu Boards Act of 2008," 2008 October. [Online]. Available: http://senate.gov.ph/lis/bill_res.aspx?congress=14&q=SBN-2682

[4] K. Ruan and L. Shao, "Calorie Estimation from Fast Food Images," 12 December 2015. [Online]. Available: http://cs229.stanford.edu/proj2015/151_report.pdf. [Accessed 19 February 2017].

[5] Y. Kawano and K. Yanai, "Food Image Recognition with Deep Convolutional Features," 13 September 2014. [Online]. Available: http://ubicomp.org/ubicomp2014/proceedings/ubicomp_adjunct/workshops/CEA/p589-kawano.pdf. [Accessed 26 February 2017].

[6] "EngineersGarage: Introduction to Image Processing," 2012. [Online]. Available: http://www.engineersgarage.com/articles/image-processing-tutorial-applications. [Accessed 1 December 2015].

[7] TechTarget, "Definition: Machine Learning," 2015. [Online]. Available: http://whatis.techtarget.com/definition/machine-learning. [Accessed 1 December 2015].

[8] SAS Institute Inc., "Machine Learning: What it is & why it matters," [Online]. Available: http://www.sas.com/en_us/insights/analytics/machine-learning.html. [Accessed 1 December 2015].

[9] F.-F. Li, "Machine Learning in Computer Vision," [Online]. Available: https://www.cs.princeton.edu/courses/archive/spring07/cos424/lectures/li-guest-lecture.pdf. [Accessed 1 December 2015].

[10] S. Lucci and D. Kopec, Artificial Intelligence in the 21st Century: A Living Introduction, Dulles, Virginia: David Pallai, 2013.

[11] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems: Seventh Edition, New Jersey: Pearson Higher Education, 2016.

[12] J. Painter, "How do food manufacturers calculate the calorie count of packaged foods?," 2016. [Online]. Available: http://www.scientificamerican.com/article/how-do-food-manufacturers/.

[13] American Dietetic Association and American Diabetes Association, "Food Exchange Lists," 2008. [Online]. Available: http://dtc.ucsf.edu/pdfs/FoodLists.pdf.

[14] S. W. Lichtman, K. Pisarska, E. R. Berman, M. Pestone, H. Dowling, E. Offenbacher, H. Weisel, S. Heshka, D. E. Matthews and S. B. Heymsfield, "Discrepancy between Self-reported and Actual Caloric Intake and Exercise in Obese Subjects," *The New England Journal of Medicine*, 1992.

[15] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang and K. Murphy, "Im2Calories: towards an automated mobile vision food diary," [Online]. Available: https://www.cs.ubc.ca/%7Emurphyk/Papers/im2calories_iccv15.pdf