# Neural Network Based Food Recognition and Calorie Calculation for Diabetes Patients

DiaWear Technical Report (CSD Senior Thesis Extended Abstract)

Geeta Shroff (Student), Asim Smailagic (Advisor)

March 21, 2009

## DiaWear

Living with diabetes today involves a substantial amount of effort on the patient's side in regularly and frequently recording data. This discourages patients from the careful management of their disease. Two target groups of persons who are most prone to neglecting this task are young children with Type I diabetes or busy adults with Type II diabetes. In many cases, such negligence may result in serious illness or death. A USA Today study shows that about 73 Million persons in the United States either have diabetes or are at a risk. With this widespread and growing number of diabetes  patients, we have identified the need for an automatic mobile wearable monitoring and assistive system for diabetes patients to use in order to manage their disease in a more efficient and convenient manner.

We propose DiaWear, a context aware wearable food and activity recognition system, as a solution to the above mentioned problem. As shown in the diagram below (**Figure 1**), this system will monitor and assist diabetic patients with regards to their calorie intake after running a special food image recognition algorithm on the taken pictures of food in the user's plate. User preference and environment context information will play a large role as pre-filters during the image classification step of this algorithm. The system will also monitor the patient's calories burnt through the use of wearable sensors, and will use this value to assist the user regarding how much food to eat depending on prior knowledge of the user's total calorie intake quota for the day, and calories available on the plate as determined by the image recognition algorithm. The status of calories burnt and consumed will be displayed in a convenient form to the user by outputting the results to a wearable e-Watch. More detailed calculation information, logs, and context menus for static user preference and dynamic context input through the DiaWear software will be available on a mobile device such as a cellphone.

Recorded and displayed data will assist patients and also help medical professionals to provide more accurate forms of treatment and medications. We also foresee the future of such a system to include and not limit itself to applications such as chronic pancreatitis and other diseases which have strict dietary and calorie consumption and regulation needs,  weight loss management, and fitness and diet control.

**DiaWear**

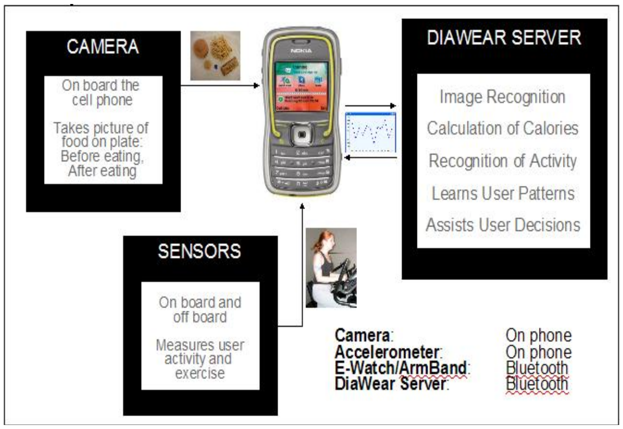

Figure 1: Overall DiaWear System

## Report Topic

This report describes our Neural Network based image training and recognition algorithms used to train, learn and recognize a small set of food items. After a brief discussion of previous work in food image recognition, the proceeding sections describe in detail the different steps involved in our algorithms. The following is a figure (**Figure 2**) demonstrating some of these steps.

**Image Processing Steps**

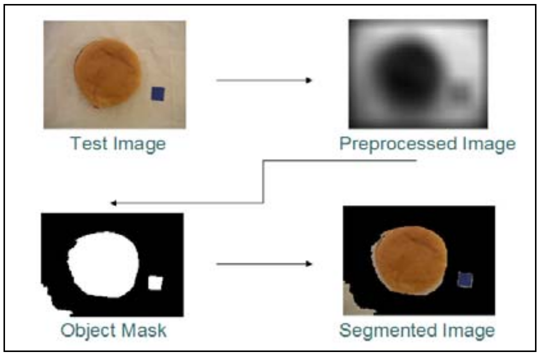

Figure 2: Image Processing Steps

## Previous Work

As for image recognition, an initial phase of this project involved studying different image recognition techniques being used to classify different foods (see references below). The effects of classification methods on color-based feature detection for food processing such as meats and fish was developed with a pre-filter to specially ignore the background of image objects. It classified images using a statistically based fast bounded box (SFBB) algorithm, which was successfully compared against common classification algorithms using SVM and neural networks [1]. A dish extraction method using a neural network based image recognition algorithm for food intake measurements for medical use was developed to determine the amount of food on a hospital tray with prior knowledge of food items and item positions in the tray [2]. A food recognition algorithm based on dish recognition has also been developed to adjust microwave settings according to food in a lunch box with predefined rules regarding positioning of foods in the lunch box according the the assigned categories for the different sections, called a food arrangement map. This group used both bottom up and top down recognition models and approaches in attacking this problem [3]. A simple simulation of trained data and feature vectors was created to better understand the concepts involved in image recognition, and a preliminary model using Support Vector Machines using Cornell University's freely available SVMLite package was studied to classify the generated vectors. After studying pattern recognition using neural networks [4], a neural network classification based approach was then created for the problem at hand using the following steps.

## Preprocessing

After scaling the input image to the desired size, minor lighting problems are resolved by adjusting the intensity of the gray scale version of the input image. We then perform adaptive equalization of the color histogram for different sub parts over the entire image, and finally pass it through a linearly averaging kernel to remove some level of noise and blur. Minor lighting problems are solved. The resized image has less amount of pixels, which allows for faster traversal of the algorithm. However, it also results in the loss of information. Although the equalized histogram allows for different lighting and shading effects to be equalized over different areas of the image, extreme lighting problems such as bright camera flashes are not resolved. The linear averaging filter does not solve this problem either and also leads to information loss. Since the mentioned negative effects do not affect the performance of the overall algorithm to a great extent under our assumptions as listed in **Table 1**, they are not considered for the purposes of the described experiments.

| **Images will contain only:** |
|---|
| Non-touching objects |
| One reference object |
| Complete objects |
| Darker objects than background |
| Single colored background |
| Minimal shadows |
| No flash lighting |

Table 1: General Image Assumptions

## Background Removal

Pixel values with minimum intensity are first removed, and the entire image is then normalized by dividing by pixel values of the maximum intensity. For correction of miscalculated masked and unmasked pixels, improper regions are flipped using an adaptive thresholding method. This correction technique is applied due to the variance in color, lighting and other intensities over different sections of the image. Sub-image matrices are extracted and an average threshold is calculated for each matrix. Corresponding pixels in the sub-image above the local threshold are marked as background or foreground depending on their color intensity values. Using this method, a black and white mask is generated in which background pixels are blacked out. This method only works for cases where the background is lighter than the foreground objects. In cases where touching objects are lighter or darker, the adaptive algorithm masks out a portion of the lighter object. This can still be ignored if the lighter intensity object is not too small. However, if very narrow, it will lead to a false positive region in the detected background mask. To overcome these limitations, we are only considering images that have lighter backgrounds with non-touching objects.

## Segmentation and Object detection

Disjoint background connected components are labeled using a 4-means connected neighbors algorithm. If the size of any of these connected components is below a threshold of 2% of the largest foreground connected component in the image, then the component is reversed to be a part of the foreground. The foreground connected components are similarly labeled, and if the size of any of these connected components is below 2% of the largest foreground component, then it is reversed to be a part of the background. After applying this small connected components removal algorithm to the adaptive normalized binary mask, it is then dilated and eroded to remove other unwanted noise. Dilation allows for the areas of foreground to grow and for the holes to become smaller. Erosion of the mask allows for removal of any extra dilation. The mask is then transposed onto the original image to obtain the segmented image with final detected objects separated from the background as shown in **Figure 2**. The final number of connected components is also noted to get a general idea of how many objects were detected in the image. This technique allows for the removal of unwanted components, and for the segmentation of detected objects from the background for further processing. Connected components may sometimes be false positives, but will be processed out in further steps of the algorithm. For objects touching each other in the image, further segmentation will have to be performed to separate the objects. This case is again ignored, since we are considering only non touching objects.

**Segmented Objects**

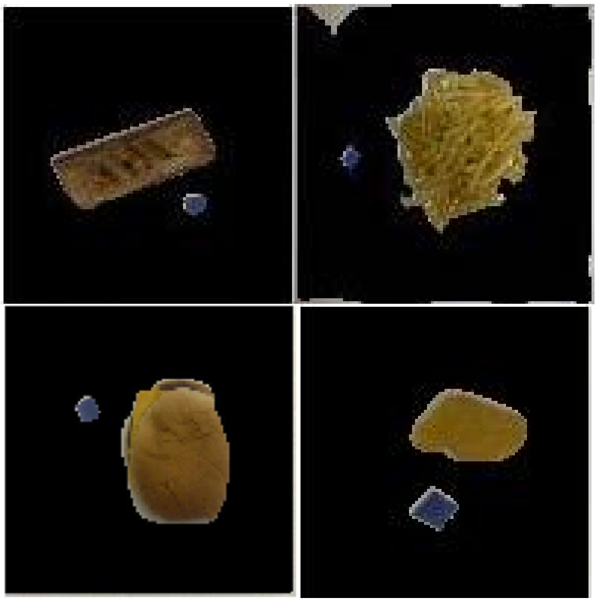

Figure 3: Segmented Objects with Blue Reference Object

## Color, Size and Texture Based Feature Extraction

A special reference object based feature extraction algorithm is run on the labeled connected components derived from the segmentation and labeling stage. Color based features include mean intensities along the red, green and blue spectrum as compared to the color based extracted features from the reference object included in the training or test image. Size based features consist of the ratio between the area of the detected object to that of the reference object. In addition, the texture of the object is characterized statistically by calculating the local entropy over different areas of the connected component. Color and size based features are easy to extract and keep the feature extraction process fast, while texture allows us to overcome overlaps in color intensities between all the brown region food ranges that we are considering in the described experiments.

## Neural Network Classifier Model

Before discussing our training and recognition algorithms and results, we describe our Neural Network classification model in detail. We have employed a 5-10-5 feed forward back propagation (FFBP) based neural network for our purposes of reliability and efficiency. The inputs to our network are limited to a range between the minimum and maximum values in our normalized feature vectors, which are guaranteed to be between -1 and 1 after normalization. The first layer of the net comprises of 5 inputs, each representing an element of the feature vector extracted from the detected object. The hidden layer of the network includes 10 neurons, since too many hidden neurons decrease efficiency while too few decrease reliability. The output layer consists of 4 neurons, one neuron representing each of the four food classes under consideration. We use a logarithmic sigmoid differentiable transfer function for the first layer, and a hyperbolic tangent sigmoid differentiable transfer function for the second layer to calculate the values at the output nodes. The training function of our feed forward back-propagation network is based on gradient descent momentum for an adaptive learning rate to update our weight and bias values. Our weight and bias learning function is also based on back-propagation techniques. The diagram in **Figure 4** represents our neural network classification model.

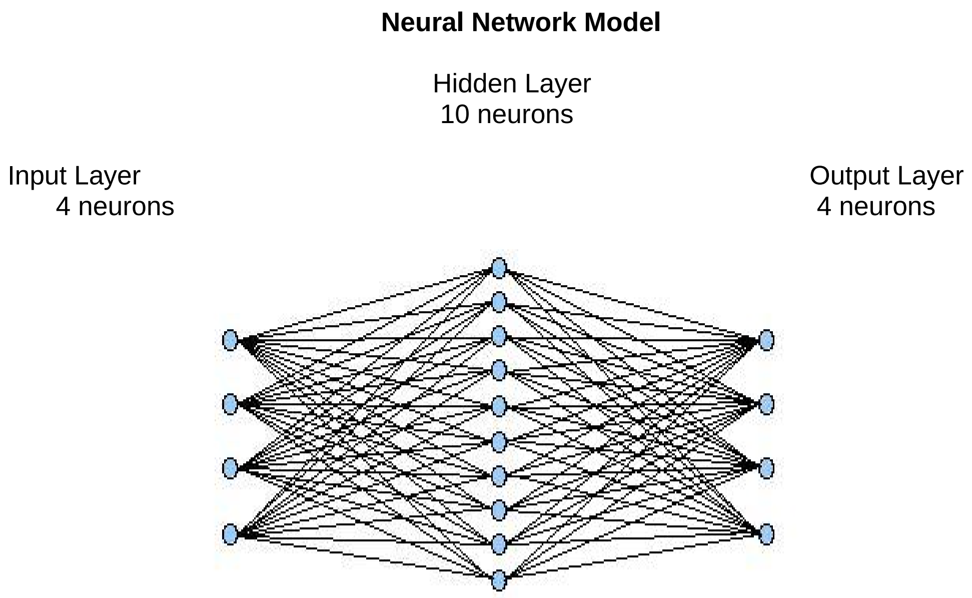

Figure 4: Neural Network Classifier Graphical Model

**Image acquisition:**

The Nokia N95 cell phone camera is set to 4.0 mega pixel resolution to capture the different images. A total of 200 images were captured in different conditions such as day light, indoor light, flash, shadows, camera blur, scale invariance and rotations. This set of images comprise of single food items (hamburger, apple pie, chicken nugget or fries) in which there exists a non touching, easily recognizable special blue reference object. The size of each of the images is scaled down to 100x100x3 RGB pixels. Of these images, 60% are used for training and verification while the remaining 40% are used for testing.

## Results

### Object Training Results:

Feature vectors extracted from the detected objects are normalized before passing through the Neural Network classifier. Each input is an n dimensional column vector of feature vectors of size m. n is the number of training images and m is the number of features extracted from each image. Each output is an n dimensional vector of target vectors of size 4. Each of the n training images has a corresponding output column vector with a 1 in the position for its target food class and a 0 in all the other 3 food class positions. The neural network is trained using 30 training images for each of the four food classes. With every new epoch, biases and weights are altered by the back propagation of mean-squared error (MSE), until we reach our desired training error goal. For instance, in the graph below, a maximum of 1295.0 epochs are required to reach an MSE of 0.0799949 on these 120 images using a momentum constant of 0.95. These values that are within our specified parameters demonstrate how the neural network classification of our training feature vectors ultimately converges to the desired target outputs within the specified error. The following graph in **Figure 5** represents this MSE performance of our neural network based training algorithm as we increase the number of epochs. At first there is random behavior, followed by a curve that eventually converges with the desired target behavior within our target MSE goal. After observing performance of the neural network over 10 trials, and after accounting for variability in performance, we allow for a maximum of 2500.0 epochs to reach our desired

MSE goal of 0.08.

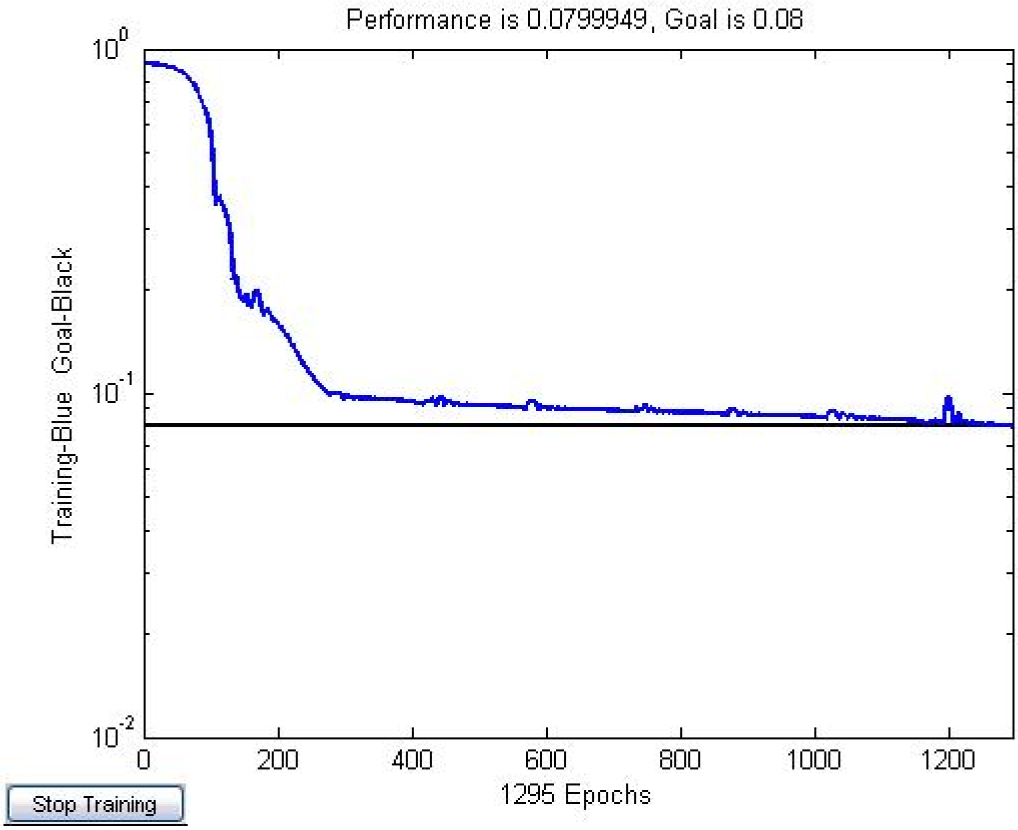

Figure 5: Mean-Squared Error (MSE) performance goal reached in 1295 epochs

**Object Recognition Results:**

After successful completion of training, the trained network is then verified against 20 random images from among the set of images that were used during the training step. After performing image preprocessing, background removal and feature extraction on the images, the feature vectors are normalized by dividing by the maximum values, and are then passed through the neural network for classification. The classification values are compared to the known values and errors and successes are noted. From these observations, the percent accuracy result is determined by counting the total number of true hits for that class and dividing by the total number of images tested for that class. The percent inaccuracy result is determined by counting the number of false classes detected in place of the known class, and dividing by the total number of images tested for that class. The percent false positive calculation is determined by counting the total number of times the class was falsely identified instead of another class, and dividing this number by the total number of images tested overall. The percent false negatives is determined by counting the number of images where the class was not identified when it should have been identified, and dividing by the total number of images tested for that class. The results of this verification step on 20 randomly selected images from among the training set of images for each food class are presented in **Table 2**. The trained neural network is further tested with 20 additional images for each item taken in different conditions that had not been included during training. The accuracy results of detecting and classifying objects in these 80 test images can be found in **Table 3.**

### Verification Results

| Food Category | % Accuracy= | % Inaccuracy | % False Positives | % False Negatives |
|---|---|---|---|---|
| Hamburger | 100*25/30=83.3333% | 100*1/30=3.3333% | 100*23/120=19.16667% | 100*4/30=13.3333% |
| Fries | 100*18/30=60.0000% | 100*5/30=16.6667% | 100*3/120=2.5% | 100*7/30=23.3333% |
| Chicken Nugget | 100*21/30=70.0000% | 100*5/30=16.6667% | 100*8/120=6.6667% | 100*4/30=13.3333% |
| Apple Pie | 100*17/30=56.6667% | 100*9/30=30.0000% | 100*1/120=0.8333% | 100*4/30=13.3333% |

Table 2: Verification results for object recognition over the 120 trained images

### Testing Results

| Food Category | % Accuracy | % Inaccuracy | % False Positives | % False Negatives |
|---|---|---|---|---|
| Hamburger | 100*15/20=75.0000% | 100*1/20=5.0000% | 100*35/80=43.7500% | 100*5/20=25.0000% |
| Fries | 100*14/20=70.0000% | 100*6/20=30.0000% | 100*6/80=7.5000% | 100*0/20=0.0000% |
| Chicken Nugget | 100*11/20=55.0000% | 100*8/20=40.0000% | 100*5/80=6.2500% | 100*1/20=5.0000% |
| Apple Pie | 100*10/20=50.0000% | 100*8/20=40.0000% | 100*0/80=0.0000% | 100*2/20=10.0000% |

Table 3: Testing results for object recognition over the 120 trained images

## Future Work

**Next Steps:**
Immediate next steps include creating a user friendly mobile cell phone application, a distant running matlab image recognition server, and connecting the mentioned client application to the server for communicating the captured image request to the server and the corresponding calorie response back to the user. Once the end to end system is complete, we will decide on relevant user groups and plan for user studies (including filling out necessary IRB forms). For this, user surveys and code to log useful information will be written.

**Other techniques:**
Other image recognition algorithm techniques to try include using the Scale Invariant Feature Transform (SIFT) algorithm for matching test images to a trained image database through a Support Vector Machine (SVM) classifier. Additional information to extract from objects to improve the performance of our current and future versions of the algorithm include more color, size, texture and shaped based features. Key point descriptors as in SIFT will also be useful. Some color based additions include relative mean intensity, relative standard deviation, and relative correlation between the detected objects and the reference object. Additional valuable size information will be the ration between the major and minor axes of the region properties ellipse of each detected connected component. Texture data can be perfected by calculating the number of hough transform lines and by using other texture filters such as local range and local

standard deviation filters. The gray level co-occurrence matrix contrast, correlation, energy and homogeneity will provide more accurate texture details that can not be obtained from the mentioned filters. Shape can be summarized using corner detection techniques based on region properties, detecting edges using edge detectors, measuring the eccentricity of the enclosing ellipse, and comparing perimeter to other parameters.

**Contextual information:**

Contextual information will provide key information to filter training image database sets so as to train the network with limited and more relevant training information. We will also provide the user with dynamically reordered options according to our contextually prioritized lists. Some such examples include using the context of other items detected (for example, only one main dish is possible), identifying user preferences (no sugar, no fat, no beef, only vegetarian, etc) and using information obtained from a user's grocery bill or restaurant bill. Location based GPS context to filter out the database (McDonald's versus Pizza Hut) and system time based context (breakfast items in morning) will also serve as very promising contextual information.

**Future research angles:**

Other research angles include detecting how much the user has eaten (before/after scenarios), logging all information in a usable manner (for patients, doctors, research, etc), combining our mobile cell phone based image recognition system with an easily accessible status bar on the e-Watch platform, combining our mobile cell phone based image recognition system with an activity recognition system, and including text recognition capabilities to detect information on items such as food packages and soda cans.

## References

[1] Effects of Classification Methods on Color-Based Feature Detection With Food Processing Applications (January 2007)

[2] Dish Extraction Method with Neural Network for Food Intake Measuring System on Medical Use (July 2003)

[3] Food recognition algorithm based on dish recognition (June 2005)

[4] Neural Networks for Pattern Recognition (1995)