Cloud-Based SVM for Food Categorization

Parisa Pouladzadeh[1, 2], Shervin Shirmohammadi[1, 2], Aslan Bakirov[2], Ahmet Bulut[2,] Abdulsalam Yassine[1]
[1]Distributed and Collaborative Virtual Environment Research Laboratory
University of Ottawa, Canada
Email: {ppouladzadeh, shervin, ayassine}@discover.uottawa.ca
[2] Colleges of Engineering and Natural Sciences, , Istanbul Şehir University, Turkey
Email: { shervinshirmohammadi, aslanbakirov, ahmetbulut}@sehir.edu.tr

Abstract- As people across the globe are becoming more interested in watching their weight, eating more healthily, and avoiding obesity, a system that can measure calories and nutrition in everyday meals can be very useful. Recently, due to ubiquity of mobile devices such as smart phones, the health monitoring applications are accessible by the patients practically all the time. We have created a semi-automatic food calorie and nutrition measurement system via mobile that can help patients and dietitians to measure and manage daily food intake. While segmentation and recognition are the two main steps of a food calorie measurement system, in this paper we have focused on the recognition part and mainly the training phase of the classification algorithm. This paper presents a cloud-based Support Vector Machine (SVM) method for classifying objects in cluster. We propose a method for food recognition application that is referred to as the Cloud SVM training mechanism in a cloud computing environment with Map Reduce technique for distributed machine learning. The results show that by using cloud computing system in classification phase and updating the database periodically, the accuracy of the recognition step has increased in single food portion, non-mixed and mixed plate of food compared to LIBSVM.

Keywords: Calorie measurement, Food Image processing, Cloud computing.

I. INTRODUCTION

Nutritional epidemiology is concerned with quantifying dietary exposures and the association of these exposures with risks for disease [1]. Diet represents one of the most universal biological exposures; however accurate assessment of food and beverage intake is difficult [2]. The availability of "smart" mobile telephones, improved memory capacity, network connectivity, and faster processors allow these devices to be used in health care applications. A dietary assessment application for a mobile telephone provides a unique mechanism for collecting dietary information. Measuring accurate dietary intake is considered to be an open research problem in the nutrition and health fields. The approach includes the use of image analysis for identification and quantification of food consumption based on images of food items.

In this paper, we propose a system by which we can automatically estimate the food consumed at a meal from images acquired by a mobile device. Each food item is segmented, identified, and its volume is estimated. "Before" meal and "after" meal images can be used to estimate the food intake. Using such information, the nutrients consumed can be determined using a food composition database. Our system uses more than 3000 images for the food classification, image segmentation identification, and calorie measurement. Images are taken under different conditions such as different cameras, lighting, and angles. We also use variety of food such as solid and liquid food, and mixed and non-mixed food. The proposed system uses the SVM method for object classification. SVM method is known for its robust and accurate classification. Existing works use much fewer images (typically hundreds) for very specific food, and also do not consider the variations of image conditions as in our system. For example, [3] has used the shape and texture features with only 180 images of food with very distinct shape and texture. The work in [4] uses only fruits in fruit salad, and [5] only considers 120 pizza images.

The process of segmentation and classification of food images is known to be complex and computationally intensive. Hence, a high power processor is needed to run some of the sophisticated segmentation methods and to test other SVM kernels to achieve accurate results. Furthermore, our system is intended to be available for users anywhere they find themselves need to enjoy a meal. Therefore, an online-based system for food recognition is crucial for users' convenience. We use the Mobile cloud-computing (MCC) for our proposed system, since it is targeted mostly for mobile devices. Mobile cloud computing is the combination of cloud computing and mobile networks to provide benefits for mobile users. Using MCC not only satisfies the computational complexity constraint of our system but also, it helps to achieve more accurate results. In other words, since the cloud has access to all other client's data, it can update its food database easier, leading to more accurate results.

The details of our proposed cloud based food recognition system and its implementation are presented in this paper. The results of the experiments show that the proposed system surpasses existing studies in different aspects in relation to food image segmentation, classification, identification, and calorie measurement. The main contributions made in this work are as follows:

- We propose a cloud-based SVM system for food categorization and calorie measurements that uses more than 3000 images and considers variety of food. This is rather significant compared to existing work which uses few hundreds of images for specific type of food;
- We use more features than other systems, including color, texture, size and shape, whereas most existing methods use only color and shape features. Results of our experiments show that when considering four features the accuracy can significantly increase compared to using fewer features;
- We design a method to apply Gabor filter for texture segmentation of food images. To do this, a bank of Gabor filters with different desired orientations and wavelength are applied to an image. Texture plays an important role in identifying different food portions;
- We provide a mechanism by which we periodically update the MAP reduce SVM model. By so doing, we made sure that the system is periodically trained so that it can correct any inaccuracies that may occur during the classification phase;

The rest of this paper is organized as follows; Section II covers related work in this area, while Section III presents a brief background of the model. Section IV presents our proposed system followed by the experimental results and analysis in section V. Finally, in section VI we conclude the paper and provide direction for future work.

II. RELATED WORK

In this section, a review of some of the most popular dietary assessment methods is provided. The objective here is to describe the advantages and major drawbacks of those methods. This will demonstrate the significance of our mobile food recognition system, which can be used for population and clinical based studies to improve the understanding of diet.

In 24-hour dietary recall method, the respondent is asked to remember and report all food consumed in the previous 24 hours. The recall is normally prepared through an in person or telephone interview. The interview usually needs specific probes to help the respondent remember all foods consumed in the day. In this method, the interviewer investigates daily reports to help the patient getting a better program for the other days [6]. While helpful this method has a major drawback related to underreporting. In [7] for example, it has been shown that features such as obesity, gender, and education, seeming health status, age and ethnicity are underreported. In [8], the authors also found that important information about food portions are underreported. Underreporting of food intake is discussed in other studies such as [9] . It has been observed that portion sizes have grown considerably in the past 20 to 30 years [8][9], this may be a contributor to underreporting. Obviously, there is a need for methods to capture accurate portion sizes as well as to collect accurate dietary information.

Existing methods such as those proposed in [8], [10][11], and [12] use semi-automated approach, which provides more accurate and faster ways to analyze food portions. In [8] and [10], the authors used image processing techniques to measure and analyze large food portions. A similar approach is also reported in [11], where the idea is to take pictures of the food, and based on a calibration card located inside the picture as a measurement pattern, the size of the food portions is calculated. In this study, the food is manually identified with the help of nutritional information retrieved from a database. Then, the calories are calculated for each picture and finally the complete set of information is stored in different database in the research facility. In this case, based on the known size of the calibration card, the portions can be translated into real life size, and the calculations are closely related with the real caloric content of each food. Martin et al. [12], proposed a system where the user captures the images, with the calibration card also, then the images are sent to a research center to be analyzed. This will, of course, come with its own shortcoming of offline data processing; meanwhile our system is intended to perform the analysis in the same place where the patient is located.

In our work, we use cloud computing model to have access and to update the database as much as the system needs. Some existing systems use cloud computing models and machine learning are mentioned here. Low and et al. [13], extended Graph Lab framework to support dynamic and parallel computation of graphs on cloud. The study implements extensions to pipelined locking and data versioning to avoid network congestion and latency. Graph Lab framework successfully deployed on large Amazon EC2 cluster to run performance tests. In [14], an experimental Page Ranking system, called Pregel, is implemented. The results show that Distributed Graph Lab performs better than Hadoop 20 to 60 times. Kraska et al. [15], introduced a new distributed machine learning system called MLBase. MLBase allows researchers to declare machine learning problems in very simple way and implements this algorithm in distributed and highly scalable manner without extensive systems knowledge. The optimizer in the framework converts ML jobs into an artificial learning plan and returns best answer to the user by improving result iteratively in the background.

III. BACKGROUND

Support Vector Machine (SVM) techniques have been used extensively in food recognition application [16], [17]. The qualities of SVM based classification have been proven to be remarkable. In its basic form SVM creates a hyperplane as the decision plane, which separates the positive and negative classes with the largest margin [18]. SVMs have shown a high level of accuracy in classifications due to their generalized properties. The evaluation results in [16][17] showed that SVM performs better than other classifiers in terms of accuracy. However, the training time of the SVM classifier is notably longer than that of other classifiers. It has been widely recognized that SVMs are computationally intensive when the size of the training dataset becomes large. The computation time in SVM training is quadratic in the number of training instances [19]. To speedup SVM training, distributed computing paradigms have been researched to partition a large training dataset into small parts and process each part in parallel, by utilizing the resources of a cluster of computers [18].

In the following subsections, we provide more discussion on the SVM model and the Map Reduce method.

a. *Support Vector Machine (SVM)*

Support vector machine is a supervised learning method in statistics and computer science. It is used to analyze data and recognize patterns as well as for classification and regression analysis. Consider for example the training data in Figure 1. The data are linearly separable, which allow us to select the two hyperplanes of the margin in a way that there are no points between them and then try to maximize their distance.

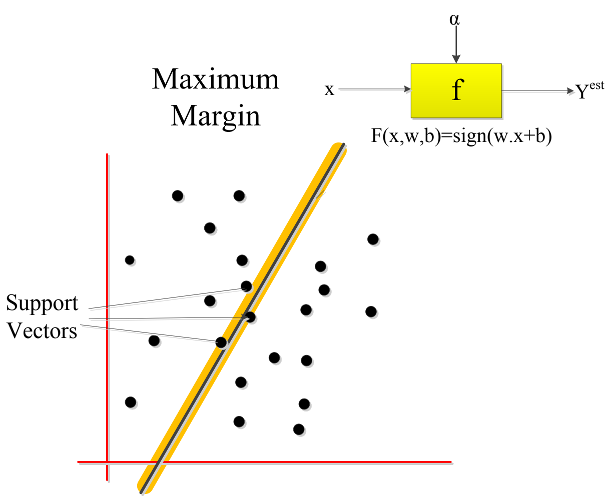

Figure 1 Linear SVM [21]

The objective of SVM is to separate the data with a hyperplane and to extend to non-linear boundaries using the kernel trick [18, 19]. The goal of calculating the SVM is to correctly classify all of the data. For the mathematical calculations we have,

$$x_i.w + b \geq +1 \qquad for\ y_i = +1 \tag{1}$$

$$x_i.w + b \leq -1 \qquad for\ y_i = -1 \tag{2}$$

Equations (1) and (2) can be combined into one set of differences:

$$y_i(x_i.w + b) - 1 \geq 0 \qquad \forall i \tag{3}$$

In the above equations, x is a vector point and w is a weight parameter, which is also a vector. In order to separate, the data, equation (1) should always be greater than zero. Among all possible hyperplanes, SVM selects the point with longest distance from the hyperplane. If the chosen hyperplane is located at the farthest possible point from the data, then this desired hyperplane, which maximizes the margin, also bisects the lines between the closest points on the convex hull of the two datasets. The distance of the closest point on the hyperplane to the origin can be found by maximizing x, as x is on the hyperplane. Similarly, for the other side points we have a similar algorithm. Thus, by solving and subtracting the two distances we get the summed distance from the separating hyperplane to the nearest points.

$$Maximum\ Margin\ = \ M\ = \ 2\ /\ ||w|| \tag{4}$$

b. *Map Reduce*

Map Reduce is a programming model derived from the map and reduce function combination from functional programming. Map Reduce model widely used to run parallel applications for large scale data sets processing. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key [14]. Map Reduce is divided into two major phases called map and reduce, separated by an internal shuffle phase of the intermediate results [15]. Simply, a Map Reduce job executes three basic operations as follows: The first task is Map function that processes in a parallel manner by each node without transferring any data to other nodes. In second operation, processed data by Map function is repartitioned across all nodes of the cluster. Lastly, Reduce task is executed in a parallel manner by each node with partitioned data. An overview of the Map Reduce system is shows in Figure 2.

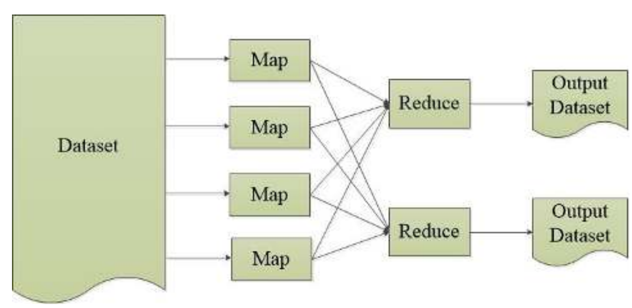

Figure 2 Overview of Map Reduce System [20]

A file in the distributed file system (DFS) is split into multiple chunks and each chunk is stored on different data-nodes. A map function takes a key/value pair as input from input chunks and produces a list of key/value pairs as output. The type of output key and value can be different from input key and value, equation (5) is showing the model.

$$\text{Map (key1; value1); list(key2; value2)} \tag{5}$$

A reduce function takes a key and associated value list as input and generates a list of new values as output which is shown in equation (6).

$$\text{Reduce (key2; list (value2)) list (value3)} \tag{6}$$

Each Reduce call typically produces either one value v3 or an empty return, though one call is allowed to return more than one value. The returns of all calls are collected as the desired result list. Main advantage of Map Reduce system is that it allows distributed processing of submitted job on the subset of a whole dataset in the network.

IV. PROPOSED SYSTEM

In this section, we discuss our proposed system in more detail. In Figure 3, we provide a block diagram of the proposed system. The system consists of the following stages: Image acquisition and pre-processing, image segmentation, classification, and measurements.

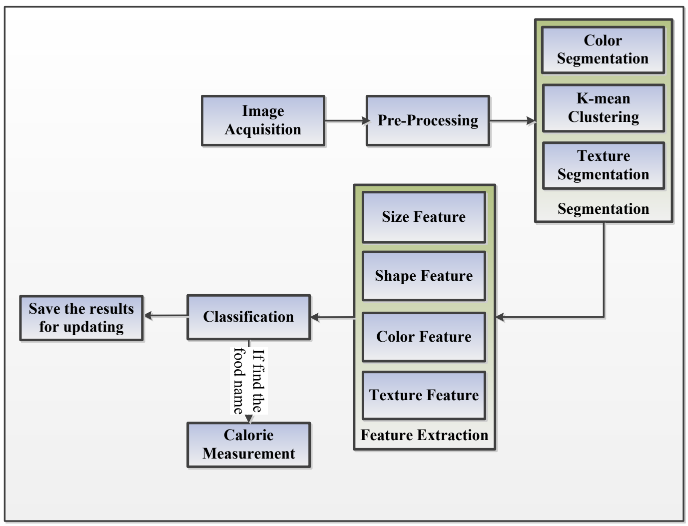

Figure 3 Image Analysis System

At the high level, the system works as follows: The user captures three pictures of the food with his/her thumb on a suitable position on the dish so the picture will not only contain the food item, but also the user's thumb, which is used for size calibration. The first two images are taken before the food consumption, one from the top view that will enable us to extract the portions and its corresponding areas, the other from the side of the dish, to analyze the height of the food items inside the dish. With these two measurements, we can obtain a better approximation for the volume, and its translation to calories and nutritional facts. The third picture must be taken at the end of food intake, to subtract from the calculations the food not consumed by the patient. The technique of using the thumb in a photo captured has an important usage in our system, because the thumb is considered as a standard for calculating the dimensions of the food items. Compared to the previous measuring method such as PDAs and the calibration card, thumb is more flexible, controllable and stable standard, giving to the patient the freedom to use the application without the need to carry around uncommon equipment or in this case measurement patterns. As an alternative to the thumb (for disabled patients who might not have a thumb), the user can pace a coin inside the image, so the system will use this coin instead of the finger, to translate the portions of the food from the picture size into real life size. The system is designed to store the patient's thumb size during its one-time calibration process. Once the food is recognized and the application suggests the type of food, the user is responsible to accept or correct the type of food from the application in the mobile device.

In the following subsection we will explain each step of Figure 3 in more detail.

A. Pre-processing

First of all, in order to have accurate results for our segmentation, a simple transformation must be performed on the image to change the image size into standard format. To do so, the size of each image will be compared with standard size categorizes. If the image size is not compatible with any size category, some cropping or padding techniques will be applied to the image. In this paper, we have defined one size category, i.e. 970×720 for simplicity. Larger images will be adjusted to this size, before performing any image processing technique.

B. Image Segmentation

At the segmentation step, each image is analyzed to extract various segments of the food portion. We paid significant attention to the segmentation mechanism design to ensure that images are processed appropriately. Particularly, we have used color segmentation, k-mean clustering, and texture segmentation tools. In this subsection, we show how these steps and the tools used lead to an accurate food separation scheme.

1) K-mean Clustering

There are many clustering algorithms in the open literature such as mean shift and k-means. The mean shift algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. The k-mean algorithm iteratively computes the mean of a set of clusters, until it converges to a stable set of cluster samples. In gray-scale images, areas are typically modeled as uniform intensity areas. Segmentation algorithms employ some form of Euclidean distance measure to determine pixel similarity either on a spatially local basis or on a global color basis. For color image processing, the clustering algorithms operate in complex multidimensional spaces.

Because of the added complexity of needing three variables to represent color pixels, the issue of region segmentation in color images is not as well defined as for gray-scale images.

In the segmentation step of this paper, we focus more on creating regions of similar color. This means that the choice of distance measure becomes very important since similarity depends very much on how distances between colors are being measured. In this case, all approaches found in the literature use some form of Euclidean distance to determine similarity between two color pixels.

2) *Texture Segmentation*

To obtain more accurate results in the segmentation stage, we added texture segmentation to the method. For texture features, we used a Gabor filter to measure local texture properties in the frequency domain. The Gabor filter describes properties related to the local power spectrum of a signal and has been used for texture analysis [21]. A Gabor impulse response in the spatial domain consists of a sinusoidal plane wave of some orientation and frequency modulated by a two-dimensional Gaussian envelope and is given by equation (7).

$$h(x,y) = \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right]\cos(2\pi Ux+\varphi) \tag{7}$$

It is suitable for our use, where texture features are obtained by subjecting each image to a Gabor filtering operation in a window around each pixel and then estimating the mean and the standard deviation of the energy of the filtered image. A Gabor filter-bank consists of Gabor filters with Gaussians of several sizes modulated by sinusoidal plane waves of different orientations from the same Gabor-root filter as defined in (7). In our implementation, a bank of Gabor filters with six different desired orientations and five wavelengths are applied to the image. Furthermore, we have included the spatial coordinates of the pixels as two additional features that we have in our segmentation to get an accurate result in this part. The outcome of each of these Gabor filters is a two-dimensional array, with the same size of input image. The sum of all elements in one such array is a number that represents the matching orientation and spatial frequency of the input image.

C. *Classification*

In this stage, the extracted features are classified in order to recognize each food portion. For this purpose, we used SVM, which is a popular technique used for data classification. A classification task usually involves training and testing data, which consist of some data instances. Each instance in the training set contains one class label and several features.

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. Afterwards, the food item has to be classified, using SVM mechanism [22][23]. SVM is one of the popular techniques used for data classification. A classification task usually involves training and testing data which consist of some data instances. Each instance in the training set contains one class label and several features. The goal of SVM is to produce a model that predicts target value of data instances in the testing set, which is given only by their attributes. In our model, we use the radial basis function (RBF) kernel, which maps samples into a higher dimensional space in a non-linear manner. Unlike the linear kernels, the RBF kernel is well suited for the cases in which the relation between class labels and attributes is nonlinear. In our proposed method, the feature vectors of SVM contain 5 texture features, 5 color features, 3 shape features, and 5 size features. The feature vectors of each food item, extracted during the segmentation phase, will be used as the training vectors of SVM. For increasing the accuracy, after the SVM module has determined each food portion type, the system can optionally interact with the user to verify the kind of food portions. For instance, it can show a picture of the food to the user, annotated with what it believes are the portion types, such as chicken, meet, vegetable, etc., as described in [24] and shown in Figure 4. The user can then confirm or change the food type. This changes the system from an automatic one into a semi-automatic one; however, it will increase the accuracy of the system.

Figure 4 The SVM module verifies with the user the type of foods it has determined. [25]

The system then measures the volume of each food portion and converts it to mass, using available density tables, and finally uses the mass and nutritional tables to measure the overall calorie and nutrients in the food. The two latter components; i.e., food portion volume measurement and calories measurement, are the focus of this paper and will be explained in the next section. The system also has a module that allows the user or the dietician to use the measurement results and manage the user's eating habits or clinical program. This module provides useful graphs such as daily intake, weekly intake, comparison between various dates, and percentage change in calorie consumption, as discussed in [24].

D. Proposed Measurement Method

As we know, CloudSVM is built on the LibSVM and implemented using the Hadoop implementation of MapReduce. The implementation of Map Reduce for the SVM model can be categorized into the following steps: First, statistics computation for features (color, size, shape etc.) and class objects. Second, transform the sample by implementing the SVM model, after that, computing statistics for new feature space and finally distributing the new samples and training the model in a random order with the reducer function. The SVM model is implemented in parallel with the help of Map Reduce mechanism wherein each instance is trained with a SVM model. The support vector of each subSVM are taken as input of next layer subSVM [20-I will Add at the end]. The non-support vectors are filtered with subSVMs.
Furthermore, CloudSVM is a MapReduce based SVM training algorithm that runs in parallel on multiple computers with Hadoop.

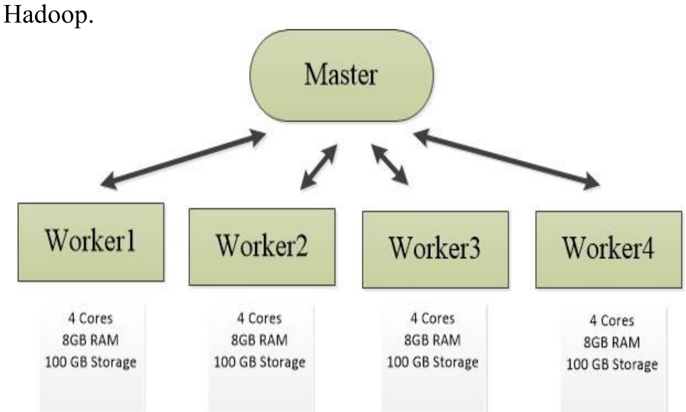

Figure 5 Cluster Configuration

1) Building SVM in parallel and in batch from scratch

In order to classify input images into different food categories, we use a support vector machine based classifier. For building an SVM and for scaling the actual build task with increasing number of images, we utilize a cluster of workers mastered by one head node; the model is shown in Figure 5. Our cluster contains five Nodes. One of these nodes is called "Master" and the remaining nodes are called Workers. Each and every node in our cluster has the same resource configuration: 4 core CPU, 8 GB RAM memory and 100 GB disk space as shown in Figure 5. Each machine runs Apache Hadoop version 1.0.4. Apache Hadoop is an actual implementation of MapReduce task execution framework [25], which is primarily used for efficiently processing very large datasets in parallel and offline. A Hadoop-based framework mainly consists of two main components: (1) Hadoop Distributed File System (HDFS) and (2) Hadoop MapReduce. HDFS is a distributed, fault-tolerant, and highly scalable file system. It requires a namenode that intercepts, accepts, and serve all file access requests. It commands an army of datanodes that are used to store the actual data blocks. On the cluster shown in Figure 5, we installed an HDFS headed by the master as the namenode and the workers acting as the datanodes.

A MapReduce job consists of a map task and an accompanying reduce task. Tasks operate on data that is stored in the HDFS. For a given map task, a core responsible to execute the task at any of the workers independently processes a data partition of size equal to the HDFS block size. Each map task runs in serial in itself. Once all map tasks are finished, the interim results produced by the map tasks are co-located around a key identifier at a destination node, where they are to be reduced to a final result by a reduce task. In order to build an SVM on Hadoop, we used a cascade-SVM implementation. The image features are stored on our HDFS cluster. In order to test the scalability .of building the final SVM, we ran a test on 250K, 500K, 750K, 1M and 1,5M images respectively. The results are shown in Table 1.

Table 1 Runtime for building a cascade-SVM on a cluster of 5 machines

Total number of images in the training dataset.	250K	500K	750K	1M	1.5M
Time required to build SVM distributed (in minutes).	3.5 mins	5.1 mins	19.3 mins	51.7 mins	66.4 mins

2) Maintaining an SVM online with incoming new images

Instead of building an SVM from scratch that takes more than an hour on 1.5M images, we defer the build task as long as the SVM accuracy is above a pre-determined threshold. This threshold dictates when it is acceptable to continue updating an already built SVM model with new image features and class labels online. The main algorithm that drives this aspect is outlined below in Algorithm 1.

Algorithm 1

Let S denote the set of images S accrued so far in the HDFS. We use a certain percentage of S for training denoted by L and the remaining part is used for testing denoted by T. Note that S satisfies S = L + T.

A. *Build a cascade-SVM denoted by M using the images in L.*
B. *Test M on the set of images T. Measure the classification accuracy, x as Benchmark.*
C. *With every new incoming image I,*

C.1. Add the image to the set of training images, i.e., L = L + I

C.2. Incrementally update M using I to obtain M'.
C.3. Test M' on T again and measure its classification accuracy as y.
C.4. If x > y, then retrain a new SVM from scratch on L + I, obtain model M''.
C.5. Test M'' on T again and measure its classification accuracy as z.
C.5.1 If z > x, use M'' else if z < x, use M for predictions.
C.6 If x < y, use M' for image classifications.

In the above algorithm we assumed x to be the threshold for Degree of Inaccuracy (DoI), it can be adjusted based on the accuracy constraints of the system.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A number of experiments were carried out to identify the accuracy and performance of the Map reduced SVM on food classification and comparing them with the LIBSVM. We have applied these two methods on different categories of food, named single food portion and food plate. In the following sub-sections we will explain our simulation settings and outcomes for each of these food categories.

A. Single food portion

First, we calculated the accuracy of our system on different single food portions, including various fruits and single piece of food. For LIBSVM approach, we chose 100 images for testing and 100 images for training phase. The results of LIBSVM method are shown in the third column of Table 2.

In another simulation we have applied the updating method by engaging Map reduced SVM, which follows Algorithm 1 for updating purpose. The accuracy results of this method for single food portions are shown in forth column of Table 2. The total average shows that we have increased in the results on different food portions, which are around 3% in only limited number of food. Also in the following we will see a huge increase in the accuracy of non-mixed and mixed food.

B. Food plate

In order to identify the accuracy of mentioned methods on food plate, we have considered two different categories of food, named mixed and non-mixed categories; some examples are shown in Figure 6. In our database, we have around 3000 Non-mixed and 500 mixed plate of food. For non-mixed food, we firstly made three groups of images, containing 1000, 2000, 3000 images, respectively. Secondly, for each group we kept 1000 images for test purpose. The system is trained with LIB SVM using half of the remained images in each group.

Table 2 Accuracy results of single food for LIBSVM and Map reduced SVM methods

No.	***Food items***	***Recognition Rate (%)***	
		Using All Features (10 fold cross validation)	***Using All Features (updating data-base periodically***
1	Red Apple	97.64	92.11
2	Orange	95.59	91.32
3	Corn	94.85	98.2
4	Tomato	89.56	93.82
5	Carrot	99.79	93
6	Bread	98.39	93.5
7	Pasta	94.75	98.42
8	Sauce	88.78	92.14
9	Chicken	86.55	90.12
10	Egg	77.53	92.53
11	Cheese	97.47	93.43
12	Meat	95.73	97.73
13	Onion	89.99	90
14	Beans	98.68	96.75
15	Fish	77.7	83.13
16	Banana	97.65	99.1
17	Green Apple	97.99	100
18	Cucumber	97.65	100
19	Lettuce	77.55	92
20	Grapes	95.7	97
21	Potato	88.56	95
22	Tangerine	97.59	98.58
23	Chocolate Cake	88.19	94.22
24	Caramel Cake	85.29	94.15
25	Rice	94.85	100
26	Green Pepper	97.99	98
27	Strawberry	83.47	90.48
28	Cooked Vegetable	92.62	96.5
29	Cabbage	77.55	89
30	Blueberry	83.47	92.4
Total Average		**91.304**	**94.5**

The simulation results for non-mixed food are shown in Figure 7. As shown in this figure, the Cloud SVM method outperforms the LIBSVM in all image categories. Furthermore, the accuracy increases as we hire more images in the training phase. We also have evaluated LIBSVM and cloud SVM methods on 500 mixed food. As the results in Figure 7 shows, although the overall accuracy is lower than non-mixed food category, we have gained around 20% accuracy over LIBSVM approach.

VI. LIMITATION

The measurement of the mass of the food needs to be improved to achieve higher accuracy. This can be achieved by:

a) Better estimation of the area of each food portion, which can be improved using more accurate segmentation methods.

b) Coming up with an approach to measure the depth of the food more accurately, instead of assuming that the depth is uniform throughout the food portion's area, which is what we assume now.

c) All of our simulations are performed on white plates with a smooth texture. We need to expand our work to various plates with different shapes, textures and colors as well [17].

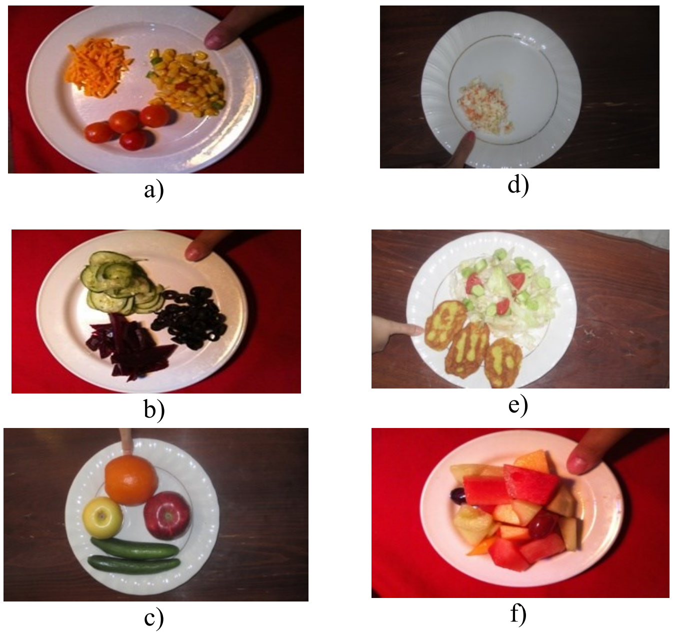

Figure 6 Non-mixed food (left) and mixed food (right)

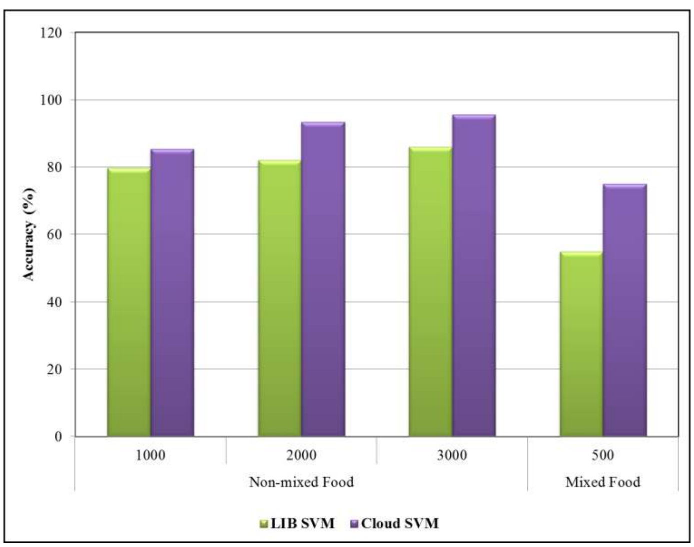

FIGURE 7 ACCURACY RESULTS OF NON-MIXED AND MIXED FOOD PLATE

VII. CONCLUSIONS AND FUTURE WORK

We have proposed distributed support vector machine implementation in cloud computing systems with MapReduce technique that improves scalability and parallelism of split data set training for food recognition and classification. The performance and generalization property of our algorithm are evaluated in Hadoop. Our algorithm is able to work on cloud computing systems. The algorithm is designed to deal with large scale data set training problems. It is empirically shown that the generalization performance and the risk minimization of our algorithm are better than previous results. In the future, we are going to increase database size for the training phase. We will also improve the segmentation part of the system, which plays an important role in this method.

VIII. REFERENCES

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

[2] M. Carrol, L. Curtin, M. Lamb, and K. Flegal C. Ogden, "Prevalence of high body mass index in us children and adolescents, 2007-2008," *Journal of the American Medical Association*, vol. 303, no. 3, pp. 242-249, January 2010.

[3] Yu Deng, Shiyin Qin, and Yunjie Wu, "An Automatic Food Recognition Algorithm with both," *Image Processing and Photonics for Agricultural Engineering*, vol. 7489 748905-1, 2009.

[4] S.A.Madival, and S.A.Madival Dr Vishwanath.B.C, "Recognition of Fruits in Fruits Salad Based on Color and Texture Features," *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, September 2012.

[5] Cheng-Jin Du and Da-Wen Sun, "Pizza sauce spread classification using colour vision," *Journal of Food Engineering*, pp. 137-145, 2005.

[6] Y. C. Probst and L. C. Tapsell, "Overview of computerized computerized dietary assessment programs for research and practice in nutrition education," *J. Nutr. Educ. Behav*, pp. 20–26, 2005.

[7] D. H. Wang, D. H. Kogashiwa, and S. Kira, "Development of a new instrument for evaluating individuals' dietary intakes," *J. Am. Diet. Assoc.* 106, pp. 1588–1593, 2006.

[8] L. Harnack, L. Steffen, D. Arnett, S. Gao, and R. Luepker, "Accuracy of estimation of large food portions," J.*Am.Diet.Assoc*, vol. 104, pp. 804–806, 2004.

[9] R. Johnson, R. Soultanakis, and D. Matthews, "Literacy and body fatness are associated with underreporting of energy intake in US low-income women using the multiple-pass 24-hour recall: a doubly labeled water study," *J.Am.Diet.Assoc*, vol. 98, pp. 1136–1140, 1998.

[10] L. Young and M. Nestle, "The contribution of expanding portion sizes to the us obesity epidemic," *American Journal of Public Health*, vol. 92, pp. 246-249, 2002.

[11] R. Patterson, A. Kristal, and C. Cheney S. Rebro, "The effect of keeping food records on eating patterns," *Journal of the American Dietetic Association*, vol. 98, pp. 1163-1165, 1998.

[12] J. Trabulsi and D. Schoeller, "J. Trabulsi and D. Schoeller, "Evaluation of dietary assessment instruments against doubly labeled water, a biomarker of habitual energy intake," *American Journal of Physiology - Endocrinology And Metabolism*, vol. 281, pp. E891– E899, 2001.

[13] Yucheng, et al. Low, "Distributed GraphLab: A framework for machine learning and data mining in the cloud," *Proceedings of the VLDB Endowment 5.8* , pp. 716-727, 2012.

[14] Grzegorz, et al. Malewicz, "Pregel: a system for large-scale graph processing," *in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010.

[15] A. Talwalkar, J.Duchi, R. Griffith, M. Franklin, M.I. Jordan T. Kraska, "MLbase: A Distributed Machine Learning System," *in Conference on Innovative Data Systems Research*, 2013.

[16] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi, "A Novel SVM Based Food Recognition Method for Calorie Measurement Applications," *in Proc. International Workshop on Interactive Ambient Intelligence Multimedia Environments, in Proc. IEEE International Conference on Multimedia and Expo*, Melbourne, Australia, 2012, pp. 495 - 498.

[17] P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi, "Measuring Calorie and Nutrition from Food Image", *IEEE Transactions on Instrumentation and Measurement*, February 25 2014, 10 pages. DOI: 10.1109/TIM.2014.2303533

[18] N.A., Liu, H.,Sung K Syed, "Incremental learning with support vector ma chines," *in Proceedings of the Fifth ACM SIGKDD International Conferenceon Knowledge Discovery and Data Mining (KDD)*, San Diego, California, 1999.

[19] C. C. Chang and C. J. Lin, "LIBSVM : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27, 2011.

[20] T. Mitchell, *Machine Learning, McGraw-Hill Computer Science Series*., 1997.

[21] P. Kruizinga, N. Petkov, and S. E. Grigorescu, "Comparison of texture features," *in Proceedings of the 10th International Conference on Image Analysis and Processing*, Washington DC, USA, September 1999, pp. 142–148.

[22] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge*, vol. 2, no. 2, pp. 121–167, 1998.

[23] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, March 2001.

[24] G. Villalobos, R. Almaghrabi, B. Hariri, and S. Shirmohammadi, "A Personal Assistive System for Nutrient Intake Monitoring," *in Proc. ACM Workshop On Ubiquitous Meta User Interfaces, in Proc. ACM Multimedia, Arizona*, USA, 2011, pp. 17 - 22.

[25] M.Victor Jose S.Sathya, "Application of Hadoop MapReduce Technique to Virtual Database System Design," *Processing of ICETECT*, vol. 11, no. 9, *pp. 892-896, 2011.*