Automated Self-Monitoring of Calorie Estimation on Food

D.D. Pukale, Shreya Agnihotri, Srushti Sameer Tikekar, Thorat Himani Popat, Shreya Sahay

Abstract :

The lifestyle of peoples is changing day by day across the globe. Now-a-day, people are choosing junk food over healthy food which is one of the major cause for health issues. One of the major issues is obesity. Research shows that food logging is beneficial in promoting weight loss and hence to solve these problems we came up with an idea to develop a web-based application “Calorie Estimation using Food logging”. In “Calorie Estimation using Food logging” user needs to register with his/her weight, height, activeness and other parameters which would be used to calculate BMI of the user. BMI would be further used to calculate calorie needed per day. The user needs to upload the images of the meal intake in the whole day. Feature extraction process would be initiated to categorize different food items of the meal. The extracted features would be compared with the trained dataset. Accordingly, calorie for each food item would be calculated. Sum of all the calories calculated over a that would be compared with the calories required by his body and on bases of the result we will predict that the user is underweight, overweight or obese. By considering their categories we will show them required diet chart and fitness activities for the respective user.

I. INTRODUCTION

Every human body requires a specific amount vitamins, calories, carbohydrates, etc. If these things get imbalanced then these will result in a health problem. Hence to keep a person away from diseases caused by inadequate consumption of calories. Therefore we propose a solution through the web-based application “**Automated Self Monitoring Of Calorie Estimation on Food**” which will help people to self monitor their calorie intakes without investing for any nutritionist or dietitian. The system requires the user to upload the image of eatables which will be used to calculate calories gained by the individual. As a prerequisite of the project, training of different food samples are done on the dataset. This training is done using four layers of **CNN** algorithm. After training, we got various features of the images i.e height, width, and color which is stored in the database in the form feature vector. To use services, the user needs to register by filling form and login into the system using his/her credentials. The health parameters i.e height, weight, age, and activeness are accepted for further calculation of BMI . According to the calculated BMI the required calorie intake is displayed.The image of food is uploaded by the user from the device. After uploading, the image is redrawn and resized into a specific dimension(i.e 500*500). The new image is fed into **CNN** and features are extracted to predict the food item. The feature vector of the uploaded image is compared with the feature vectors of trained images. The Euclidean Distance of the uploaded image vector with the trained image feature vector is calculated and the item with minimum distance is selected and the item is classified to that category. To obtain more accurate result we used Naive Bayes classifier. The features from CNN are given to classifier to predict the appropriate food category. The calorie of food category detected is subtracted from the required calorie intake and accordingly a generalized diet plan is generated.

Keywords
Food image, classification, recipes(ingredient).

RELATED WORK

Crowd-sourcing was used to control the food type, food size, and calorie by using Amazon Mechanical Turk. In this, tasks were repeatedly completed by workers using the platform to provide a nutritional workflow. Results from these tests indicate that using crowdsourcing to determine the nutritional value of meals is nearly as accurate as trained dieticians. Crowd-sourcing was also used for dietary rating of food images. For robustness, the scale was used so that registered user can rate each image and results show there was a high association between user scores and specifies that crowd-sourcing can be used for dietary feedback. Similar research used a traffic light diet approach to assess the nutritious quality of images.

Literature Review:

1. Paper introduced Plate Mate, a system for crowd-sourcing nutritional analysis (calories, fat, carbohydrates, and protein) from photographs of meals using Amazon Mechanical Turk. Complex tasks like this are hard problems for crowd-sourcing, as workers may vary drastically inexperience and reliability. To achieve accurate estimates, it proposed a workflow in which the overall problem is decomposed into small, manageable, and veritable steps. Plate Mate uses this work flow to assign tasks to contributors, to validate and combine results, and to appropriately route tasks for further processing. Plate-Mate allows users to upload food photographs and receive nutrition estimates within a few hours. The estimates consist of a list of foods in the photograph, with associated measurements of serving size, calories, fat, carbohydrates, and protein for each food item.

Estimates are generated from a series of tasks on Amazon Mechanical Turk. Crowd sourcing nutritional analysis presents several challenges in interface and workflow design. First,Turkers are in experienced, and may thus produce unreliable estimates. Second, most Mechanical Turk tasks are simple, and Turkers may be unaccustomed to performing complex operations like nutritional analysis if presented as a single, complex task. Finally, any individual Turker may be biased in their estimates or have trouble recognizing certain foods contained in a photograph, making it necessary to select from or combine the outputs of multiple workers.

Outcome:

Plate Mate builds on a concept of remote food photography developed by the nutrition community. While the original method relies on expert dieticians providing the estimates, Plate Mate uses Amazon Mechanical Turk to make this approach more affordable and scalable.

2. To assess if untrained individuals can accurately crowd source diet quality ratings of food photos using the Traffic Light Diet (TLD) approach Participants were recruited via Amazon Mechanical Turk and read a one-page description on the TLD. The study examined the participant accuracy score (total number of correctly categorized foods as red, yellow, or green per person),the food accuracy score (accuracy by which

each food was categorized), and if the accuracy of ratings increased when more users were included in the crowd sourcing. For each of a range of possible crowd sizes (n¼15, n¼30, etc.), 10,000 bootstrap samples were drawn and a 95% confidence interval (CI) for accuracy constructed using the 2.5th and 97.5th percentiles. Participants rated 10 foods as red,yellow, or green. Raters demonstrated high red/yellow/green accuracy (>75%) examining all foods. Mean accuracy score per participant was 77.614.0%. Individual photos were rated accurately the majority of the time There was little variation in the 95%.CI for each of the five different crowd sizes, indicating that large numbers of individuals may not be needed to accurately crowdsource food.The traffic light system is easy to follow because it's relatively simple—there's little room for interpretation.

- If the light is green, you proceed.
- If the light is yellow, you slow down and stop.
- If the light is red, you stay stopped.

Remember those three rules and you can safely navigate through intersections. Those rules can also make it much easier to eat a healthier diet. That's according to a new study from The University of Bonn in Germany. Researchers there were able to get people to make vastly healthier eating choices when they replaced traditional food labels with red, yellow, and green stickers.

The present study examined the use of the Traffic Light Diet (TLD)16,17 as a diet rating method using crowd sourcing. The goal of the TLD approach is to ''provide the most nutrition with the least number of calories, 'categorizing foods as red (eat very rarely, low-nutrient-dense, high calorie), yellow (eat in moderation),and green (low in calories, high-nutrient-dense).The TLD has been mainly used in assisting children with dietary self-monitoring to encourage the intake of low energy-dense foods and promote weight loss. The TLD approach has also been widely used to assist adults with making healthier food point-of-purchase decisions, such as in cafeterias at concession stands and on food labels. More recently, there has been an interest in using the TLD approach for self-monitoring with adults, as the TLD can be used with low-literacy populations. Previous research has also demonstrated that rating foods with a traffic light system has the potential to promote long-term changes in dietary intake and can provide a salient nutrition label that triggers processes within the brain as detected by functional magnetic resonance imaging (fMRI) that are used by adults who are successful at making healthy diet choices. The present study had five main objectives, including examining:

I if users could accurately crowd source photos of foods as red, yellow, or green after receiving a brief training on the TLD

II. if the accuracy of the ratings of foods categorized as red, yellow, or green differed from one another

III. if the accuracy of the crowd sourced food categories increased by adding more participants to crowd source the foods; which demographic characteristics, technology use, and/or nutrition knowledge factors were associated with correctly categorizing foods; and

IV.How users perceived the difficulty level of using various dietary self-monitoring methods.

Outcomes:

A total of 75 participants completed the survey. This study examined the extent to which a smaller number of users included in the crowd sourcing lead to greater variability in the mean ratings obtained. Large crowd sizes such as 75 produced mean ratings falling mostly in a narrow range, with 95% of means in the bootstrap analysis with n¼75 falling between 74.4 and 80.7.

3. This study assessed how closely crowd sourced ratings of foods and beverages contained in 450 pictures from the Eatery mobile app as rated by peer users using a simple 'healthiness' scale were related to the ratings of the same pictures by trained observers (raters). In addition, the foods and beverages present in each picture were categorized and the impact on the peer rating scale by food/beverage category was examined. Raters were trained to provide a 'healthiness' score using criteria from the 2010 US Dietary Guidelines Mobile health (mHealth) technologies hold promise as a way to provide individuals with the ability to self-monitor diet and receive feedback wherever they are. Generally, studies requiring participants to self-monitor diets have utilized paper journal methods, which can be time consuming and tedious for participants. Recently, smartphone cameras have made photographing foods a possibility, making just-in-time food recording possible. Recording food through photographs may be one way to reduce the participant burden for recording foods Finding ways to provide quick and low-cost feedback to users based on food photographs has been a challenge. Users take pictures of their foods with the Eatery app, rate their meals using a sliding scale from fit (healthy) to fat (unhealthy), and are then prompted to rate the photographs of foods and beverages from other users. In addition, users receive peer feedback as an average healthiness score for their own foods and beverages.

Outcome:

This study represents the first step in assessing the utility and accuracy of using crowdsourcing to provide very general diet feedback. The results of this study found that when basic feedback on diet quality by peer raters is crowd sourced, it is comparable to feedback from expert raters and that peers rate bot healthy and unhealthy foods in the expected direction.

4. Some researchers have attempted to evaluate overall restaurant healthiness using more standardized measures, such as nutrient profiling. Nutrient profiling is a method that categorizes food healthiness by evaluating the quantity of nutrients and/or the quality of ingredients. But these higher-quality quantitative measures are time and labour intensive. Thus, researchers who use these methods typically evaluate only a small sample of restaurants or limit a review to one point in time. When researchers have evaluated trends in restaurant foods over time, they have looked only at selected nutrients, and not whole food components (i.e. fruit and vegetable quantity). The aim of the present study was to create a process that combines nutrient- and food-based methods of analysing foods using a cost- and time-efficient procedure. We sought a process

that could easily be repeated quickly, as menus are updated. To accomplish this, we used a combination of computer science techniques and nutritional science.

Outcome:

In applying the approach to assess 22422 foods, only 3566 could not be scored automatically based on MenuStat data and required further evaluation to determine healthiness. Items for which there was low agreement between trusted crowd workers, or where the FNV amount was estimated to be >40%, were sent to a registered dietitian. Crowdsourcing was able to evaluate 3199, leaving only 367 to be reviewed by the registered dietitian. Overall, 7% of items were categorized as healthy. The healthiest category was soups (26% healthy), while desserts were the least healthy (2% healthy).

5. This work investigates the feasibility of crowd sourcing to provide support in accurately determining calories in meal images. Two groups, 1. experts and 2. non-experts, completed a calorie estimation survey consisting of 15 meal images. Descriptive statistics were used to analyse the performance of each group. Collectively, non-experts could determine which meals had larger amounts of calories and analysis showed that meals with greater calories resulted in greater standard deviations of non-expert estimates. Secondary experiments were completed that used crowdsourcing to adjust user calorie estimations using non-expert calorie estimations. Five-fold cross validation was used and results from the calorie adjustment process show a reduced overall mean calorie difference in each fold and the mean error percentage decreased from 40.85% to 25.52% in comparing original mean estimations against adjusted mean estimations. As such, there is credibility in adjusting calorie estimates from a crowd as opposed to simply taking a central measure such as the mean. Food logging consists of a person recording their food intake and recent methods use smart phones apps to make food logging process more convenient. Other methods use images for food logging, which can remove much of the complexity of food logging. The aim of this work was to investigate the feasibility of crowd sourcing of non-experts and experts in accurately determining calorie content in images of meals for food logging. The objectives were, 1) To determine if a crowd of experts and non-experts can accurately estimate the calorie content in images of meals, 2) To determine if analysing a group non-experts calorie estimations can be used to adjust calorie content in images of meals to promote accuracy.

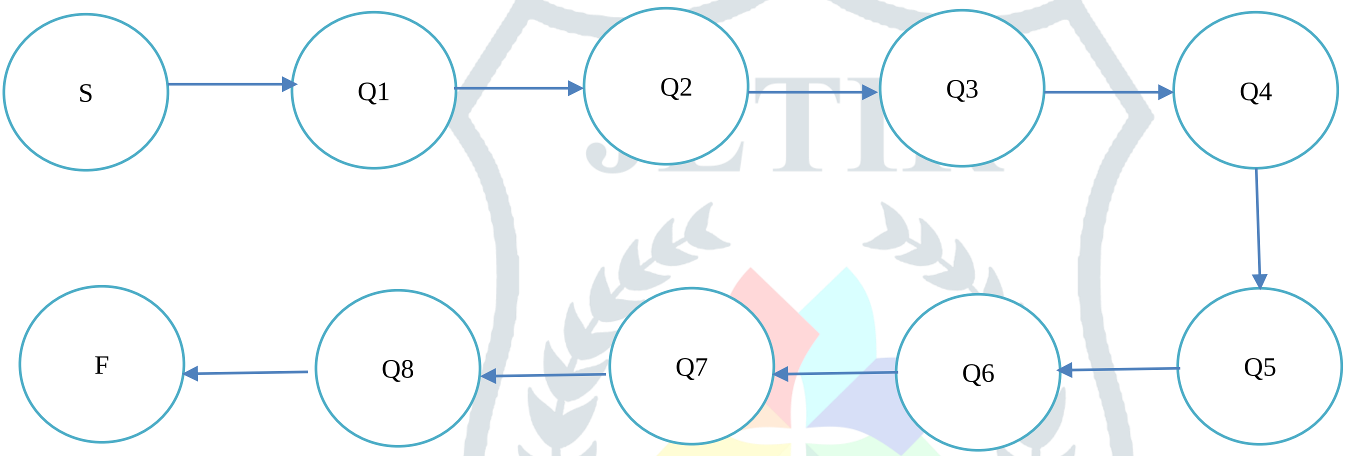
Outcome:

The work presented in this paper explores how crowdsourcing can be used determine calorie content in food images. The aim of this paper was to investigate the feasibility of using experts and non-experts to determine calorie content in meals and how ‘collective wisdom’ can be used to adjust calorie estimations. The research presented in this work highlight how crowdsourcing calorie predictions and measuring their accuracy has the potential to be used for calorie correction to adjust estimations for more accurate dietary management.

Motivation:

It is found that weight gain is a serious health problem now a day which leads to a disease such as obesity, diabetes. To avoid weight gain food journaling acts as an effective tool but existing diet application is too slow and difficult for patient use. Therefore we had proposed this paper calorie system on food. We have protected this increase of diseases by recommending a perfect diet for the users, but it depends on which type of food user is taking.

Mathematical Model



S = {S1, E, Q1, Q8, F, A}

S = Start the program

Q1 = Register / Login to the system for using the services

Q2 = Filling / editing of health form for appropriate BMI calculation

Q3 = Calculate BMI of the person along with predicting the person is healthy or not.

Q4 = Calculate calorie intake on daily basis for the person.

Q5 = Upload food image into the system for prediction

Q6 = Predict the food image using CNN and Naïve Bayes.

Q7 = Fetch the standard calorie of food predicted and subtract the calorie from the person's needed calorie.

Q8 = Give track of food which is consumed by user for the day along with food consumed.

F = Total Calorie consumption and diet chart

A = Successful Result

First, user provide food image.

System extracts features with the help of Convolutional neural network.

Let F be the set of features

F= {F1, F2, ..., Fn}

These features are compared with extracted features of training dataset images. The classifier classifies these features and determines whether the given food is sufficient or not and if it is insufficient then we have to recommend the diet plan.

These features are mapped into feature space with n dimensions. Distances from all mapped objects are calculated by using distance formula as given below,

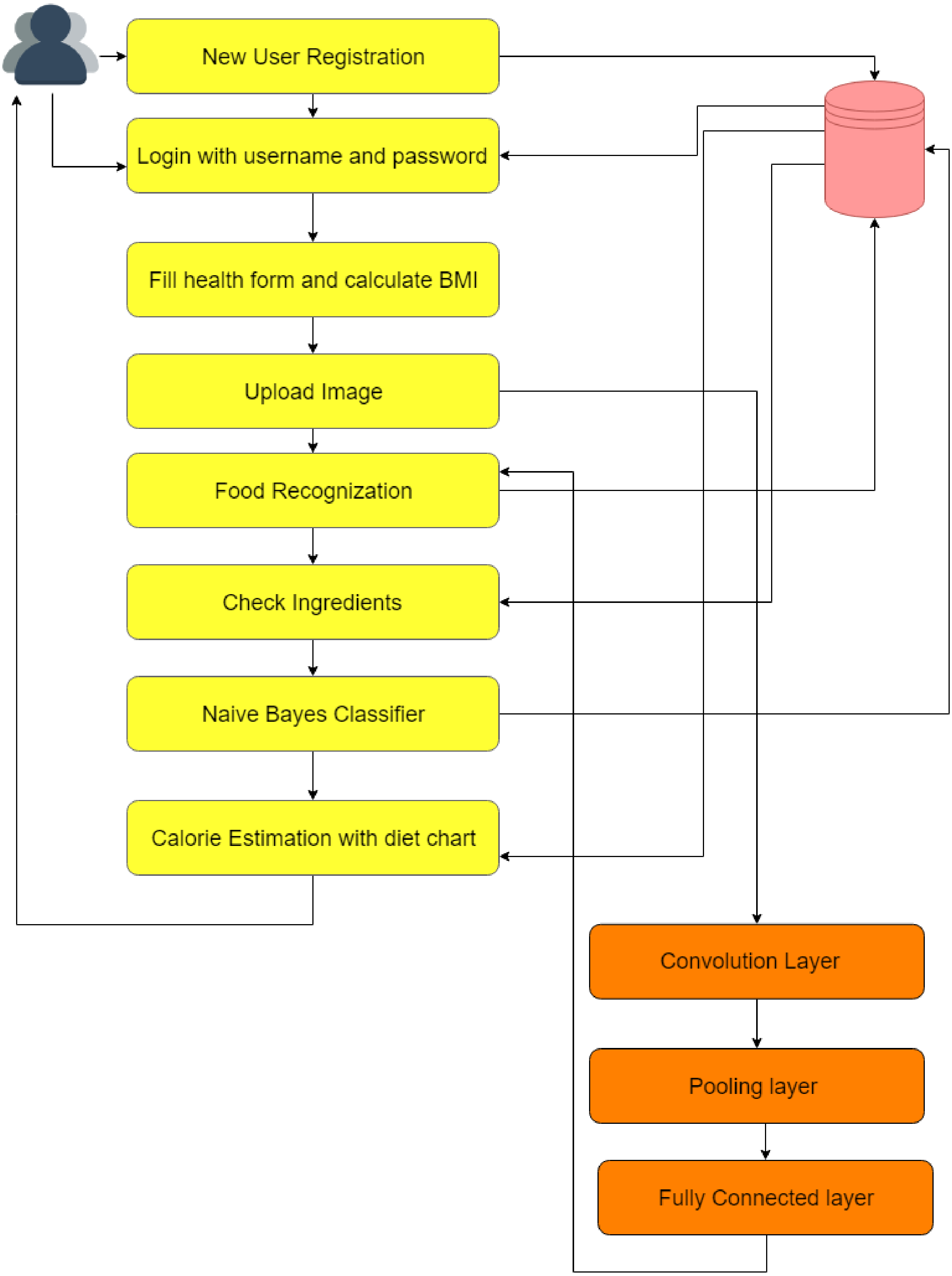
$D= Sq.rt((F_{11}-F_{12})^2+(F_{21}-F_{22})^2+...+(F_{n1}-F_{n2})^2)$

Fixed n number of objects is then extracted. Label for each object is then checked. The label with highest count among nearest neighbours is then provide the prediction.

Time Complexity of CNN = back propagation, O (n^5) O (n5), is much slower than the forward propagation, O (n^4) O (n4).

Space Complexity of CNN= O(n 2) where m is the training set size

System Architecture:



After going through all the papers which included different methodologies for determining whether the food is healthy or not. The papers include several advantages as well as disadvantages. So after going through merits and demerits of all the papers we have come up with a proposed system which will help the user to determine his or her calorie intake according to their BMI. The system will ask user different parameters for BMI calculations and according to the BMI will categorize the individual as Underweight, Overweight or Obese and will provide an appropriate diet chart.

In this system we follow Model View Controller (MVC) architecture.

In DbConnection class we need to creates the Database connection and returns the connection, so that we can avail the returned connection wherever is required.

In the model we use the private variables and implements pub

lic setters and getters. Just binding the data received from user. This bean class representing the User table in our Database (where each column in the user table has a corresponding instance variable with a setter and a getter method).

● User Registration

In this registration form, we will have a form to fill all the details which will contain First name, Last name, Email, password, contact number. This form will help us to register with the application. They take all our details and store it in a database.

1. After filling all these details we have submit button, on click of that button it takes all the inputs from the user and sending them to Controller Servlet in Form Action.

2. On the Controller Servlet we have to get all the form values from register page and set those values with the getter setter's methods and call DbConnection method to insert all the details into the database by MySQL insert Query.

● User Login

1 In this login form, give the email id and password which you have been registered at the registration form.

2 After filling the email id and password we have submit button, on click of that button it takes the email id and password from the user and sends them to LoginController Servlet in Form Action.

3 On the LoginController Servlet we have to get the email id and password from login form and set those values with the getter setter's methods and call DbConnection LoginUser method to retrieve the details from the database by MySQL select Query.

4 If the user records are stored in the database we will them a response the forward to the next page.

● Upload Image

1. To achieve this form method attribute should be set to POST method and the form enctype attribute should be set to multipart/form-data. To select an image from local PC we will use input type as a file.

2. On submit button action servlet will be called and a request will be processed into that and file will be read and write into the servlet.

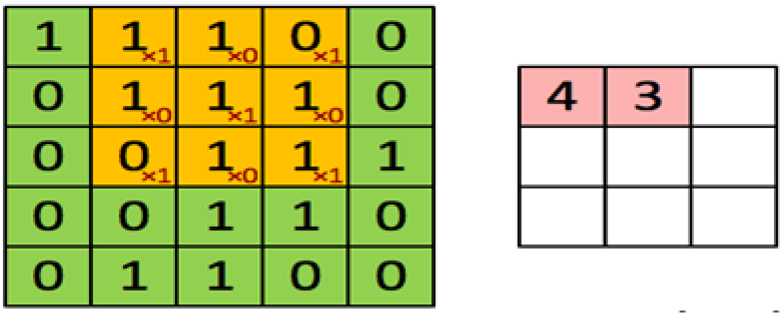
3. On the Servlet first annotate your servlet with @MultipartConfig in order to let it recognize and support multipart/form-data requests and thus get the file getPart() to work. Now with needed information the file will be stored in the database.

4. For the uploaded image CNN algorithm will be applied. Convolutional neural network(CNN) deals with the image processing. Machine learns from the data provided and based on that machine recognizes the images. For example, if we provide a certain set of different images of food to convolutional neural network model, it correctly recognizes the image type of food. Our objective is to predict the type of food an image has.The process of building a Convolutional Neural Network always involves four major steps.

Step - 1 : Convolution

When presented with a new image, the CNN doesn't know exactly where these feature will match so it tries them everywhere, in every possible position. In calculating the match to a feature across the whole image, we make it a filter. The math we use to do this is called convolution, from which Convolutional Neural Networks take their name.

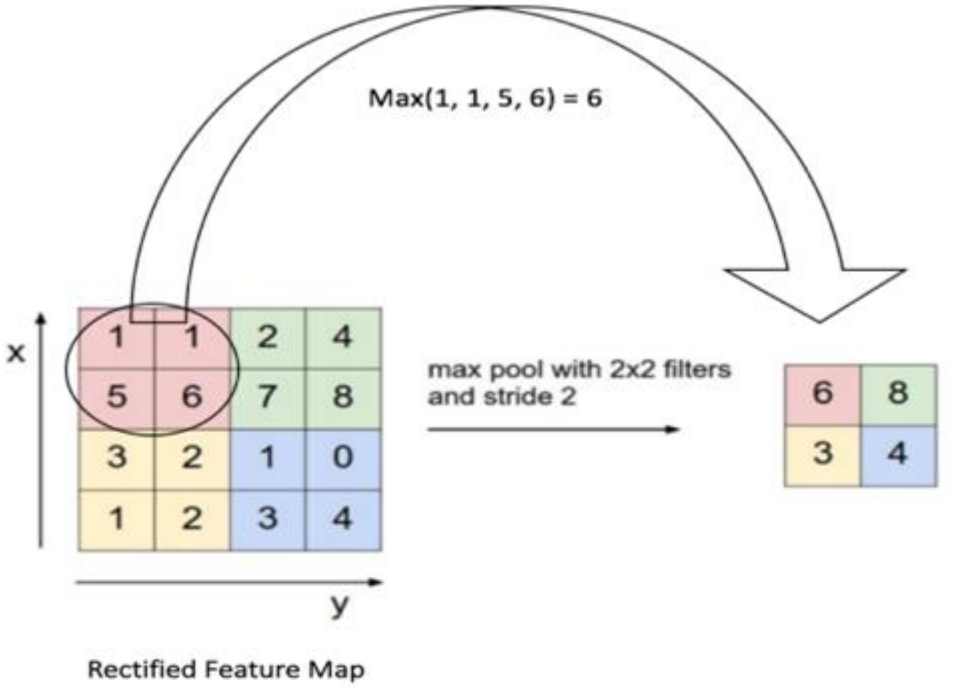
The math behind convolution is nothing that would make a sixth-grader uncomfortable. To calculate the match of a feature to a patch of the image, simply multiply each pixel in the feature by the value of the corresponding pixel in the image. Then add up the answers and divide by the total number of pixels in the feature. If both pixels are white (a value of 1) then 1 * 1 = 1. If both are black, then (-1) * (-1) = 1. Either way, every matching pixel results in a 1. Similarly, any mismatch is a -1. If all the pixels in a feature match, then adding them up and dividing by the total number of pixels gives a 1. Similarly, if none of the pixels in a feature match the image patch, then the answer is a -1.



Step - 2 : Pooling

Another power tool that CNNs use is called pooling. Pooling is a way to take large images and shrink them down while preserving the most important information in them. The math behind pooling is second-grade level at most. It consists of stepping a small window across an image and taking the maximum value from the window at each step. In practice, a window 2 or 3 pixels on a side and steps of 2 pixels work well.

After pooling, an image has about a quarter as many pixels as it started with. Because it keeps the maximum value from each window, it preserves the best fits of each feature within the window. This means that it doesn't care so much exactly where the feature fit as long as it fit somewhere within the window. The result of this is that CNNs can find whether a feature is in an image without worrying about where it is. This helps solve the problem of computers being hyper-literal.



Step - 3: Full connection

● CNNs have one more arrow in their quiver. Fully connected layers take the high-level filtered images and translate them into votes. In our case, we only have to decide between two categories, X and O. Fully connected layers are the primary building block of traditional neural networks. Instead of treating inputs as a two-dimensional array, they are treated as a single list and all treated identically. Every value gets its own vote on whether the current image is an X or and O. However, the process isn't entirely democratic. Some values are much better than others at knowing when the image is an X, and some are particularly good at knowing when the image is an O. These get larger votes than the others. These votes are expressed as weights, or connection strengths, between each value and each category.

● When a new image is presented to the CNN, it percolates through the lower layers until it reaches the fully connected layer at the end. Then an election is held. The answer with the most votes wins and is declared the category of the input.

● One of the desirable properties of CNN is that it preserves 2D spatial orientation in computer vision. Texts, like pictures, have an orientation. Instead of 2-dimensional, texts have a one-dimensional structure where words sequence matter. We also recall that all words in the example are each replaced by a 5-dimensional word vector, hence we fix one dimension of the filter to match the word vectors (5) and vary the region size, *h*. Region size refers to the number of rows – representing word – of the sentence matrix that would be filtered.

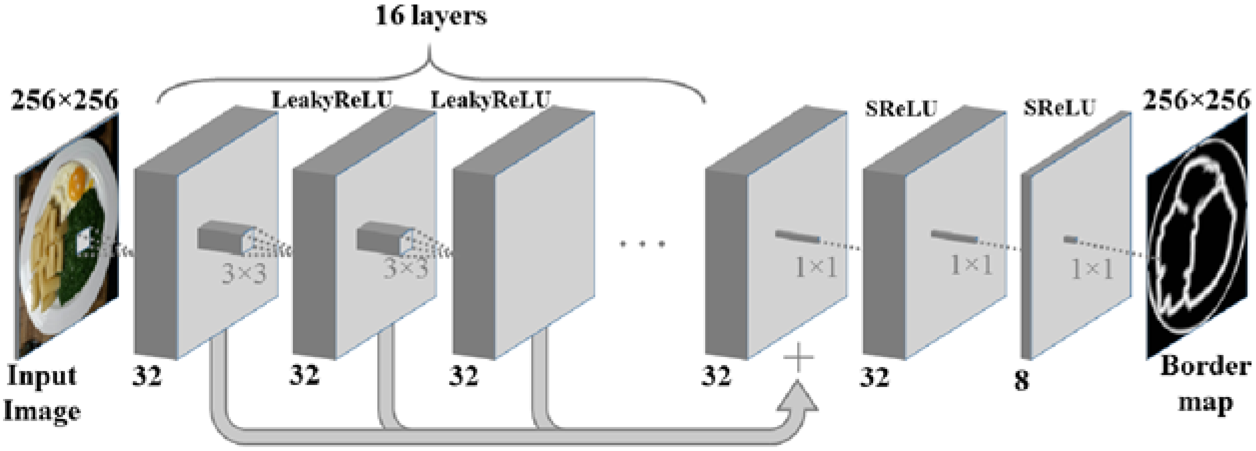


Figure 1. Architecture of the proposed CNN for food border detection

1. After the successful execution of CNN Classification, we will get a test image. In the next step naïve bayes will be applied to get the ingredient details.

2. The **Naive Bayes Classifier** technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.

The new image is fed into CNN and features are extracted to predict the food item. The feature vector of the uploaded image is compared with the feature vectors of trained images. The Euclidean Distance of the uploaded image vector with the trained image feature vector is calculated and the item with minimum distance is selected and the item is classified to that category. To obtain more accurate result we used Naive Bayes classifier. The features from CNN are given to classifier to predict the appropriate food category. The calorie of food category detected is subtracted from the required calorie intake and accordingly a generalized diet plan is generated.

After the execution of naïve bayes we will get an output as a total ingredient and finally, a diet will be given to the user

Conclusion:

The system comprises of taking health details for calculating BMI and calories according to international standards. The existing systems are working on crowd-sourcing and feedbacks. This system is implemented using CNN and Naive Bayes algorithm which gives 70% accuracy. System also provides diet plan for making a person fall into a healthy life..

.

References

[1] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos, “Platemate: Crowdsourcing Nutritional Analysis from Food Photographs,” in Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, 2011, pp. 1–12.

[2] G. M. Turner-McGrievy, E. E. Helander, K. Kaipainen, J. M. Perez-Macias, and I. Korhonen, The use of crowdsourcing for dietary self-monitoring: crowdsourced ratings of food pictures are comparable to ratings by trained observers., J. Am. Med. Inform. Assoc., pp. 16, 2014.

[3] G. M. Turner-McGrievy et al., “Crowdsourcing for self- monitoring: Using the Traffic Light Diet and crowdsourcing to provide dietary feedback, Digit. Heal., vol. 2, no. 0, 2016.

[4] L. I. Lesser, L. Wu, T. B. Matthiessen, and H. S. Luft, “Evaluating the healthiness of chain-restaurant menu items using crowdsourcing: a new method,” Public Health Nutr., pp. 1–7, 2016.

[5]Patrick McAllister1, Anne Moorhead2*, Raymond Bond1, Huiru Zheng1*,”Automated Adjustment of Crowdsourced Calorie Estimations for Accurate Food Image Logging” 2017 IEEE