FOOD CALORIE ESTIMATION USING IMAGE PROCESSING

A PROJECT REPORT

Submitted by

BHARATHI.M	**412816104014**
DEEPAK.B	**412816104020**
HARIKUMAR.G	**412816104039**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRM VALLIAMMAI ENGINEERING COLLEGE,

KATTANKULATHUR

ANNA UNIVERSITY: : CHENNAI 600 025

APRIL 2020

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**FOOD CALORIE ESTIMATION USING IMAGE PROCESSING"** is the bonafide work of "**M.BHARATHI (412816104014), B.DEEPAK (412816104020), G.HARIKUMAR (412816104039)**" who carried out the project work under my supervision.

SIGNATURE

Dr. B. Vanathi, M.E., Ph.D.,

HEAD OF DEPARTMENT

Department of CSE,
Valliammai Engineering College,
ValliammaiEngineeringCollege,
SRM Nagar,
Kattankulathur-603 203

SIGNATURE

Ms.Suma,M.E,

SUPERVISOR

Assistant Professor

Department of CSE,

SRM Nagar,
Kattankulathur-603 203

Submitted for the viva-voce examination held on _____ at Valliammai Engineering College, Kattankulathur-603 203.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We sincerely express our gratitude in depth to our esteemed founder, the chair person and authorities of **VALLIAMMAI ENGINEERING COLLEGE** for the patronage on our welfare rooted in the academic year.

We express our sincere gratitude to our respected Principal **Dr. B. Chidambara Rajan, M.E., Ph.D.,** for his constant encouragement, which has our motivation to strive towards excellence.

We thank our sincere Vice Principal **Dr. M. Murugan, M.E., Ph.D.,** for his constant encouragement, which has our motivation to strive towards excellence.

We extend our hand of thanks to our Head of the Department, **Dr. B. Vanathi, M.E., Ph.D.,** and Professor for her unstinted support.

We are thankful to Project Coordinators, **Dr. M. Senthil Kumar, M.E., Ph.D.,** Associate Professor and **Dr. V. Dhanakoti, M.E., Ph.D.,** Associate Professor for providing us with the essential facilities.

We are grateful to our internal guide **Ms. S.Suma M.E.,** Assistant Professor without whose invaluable guidance and encouragement, this project would not have been a success.

We also like to thank all teaching and non-teaching staff members of our department, for their support during the course of the project.

We finally thank our friends and family for their support during the course of project.

ABSTRACT
(ENGLISH)

As important attribute for a healthy body depends on the number of calories consumed, hence monitoring calorie intake is necessary to maintain good health. Obesity and overweight have traditionally being linked to intake of high calorie food and lifestyle. Obesity may be a medical condition during which excess body fat has accumulated to the extent that it's going to have an adverse effect on health.Changes to diet and exercising are the most treatments. Diet quality are often improved by reducing the consumption of energy-dense foods, like those high in fat or sugars, and by increasing the intake of dietary fiber.There are also other reasons that lead to monitoring of food calorie intake.In this project, we propose an efficient way to classify food objects and to measure the calorie intake by recognizing a food image using object recognition and deep learning neural networks. We propose a food calories estimation system to estimate the number of calories in an image of food taken based on the average calorie value of each detected category.

ABSTRACT
(TAMIL)

ஆரோக்கியமான உடலுக்கான முக்கியமான பண்பு நுகரப்படும் கலோரிகளின் எண்ணிக்கையைப் பொறுத்தது , எனவே நல்ல ஆரோக்கியத்தை பராமரிக்க கலோரி அளவை கண்காணிப்பது அவசியம். உடல் பருமன் மற்றும் அதிக எடை ஆகியவை பாரம்பரியமாக அதிக கலோரி உணவு மற்றும் வாழ்க்கை முறையை உட்கொள்வதோடு இணைக்கப்பட்டுள்ளன . உடல் பருமன் ஒரு மருத்துவ நிலையாக இருக்கலாம், இதன் போது அதிகப்படியான உடல் கொழுப்பு ஆரோக்கியத்திற்கு பாதகமான விளைவை ஏற்படுத்தும் அளவிற்கு குவிந்துள்ளது. உணவு மற்றும் உடற்பயிற்சிக்கான மாற்றங்கள் இதற்கான சிகிச்சைகள். கொழுப்பு அல்லது சர்க்கரைகள் அதிகம் உள்ளதைப் போன்ற ஆற்றல் அடர்த்தியான உணவுகளின் நுகர்வு குறைப்பதன் மூலமும் , உணவு நார்ச்சத்து உட்கொள்வதை அதிகரிப்பதன் மூலமும் உணவுத் தரம் பெரும்பாலும் மேம்படுகிறது. உணவு கலோரி உட்கொள்ளலைக் கண்காணிக்க வழிவகுக்கும் பிற காரணங்களும் உள்ளன .இந்த திட்டத்தில், உணவுப் பொருள்களை வகைப்படுத்துவதற்கும், கலோரி அளவை அளவிடுவதற்கும் ஒரு திறமையான வழியை நாங்கள் முன்மொழிகிறோம் , பொருள் அங்கீகாரம் மற்றும் ஆழமான கற்றல் நரம்பியல் நெட்வொர்க்குகளைப் பயன்படுத்தி உணவுப் படத்தை அங்கீகரிப்பதன் மூலம் . கண்டறியப்பட்ட ஒவ்வொரு வகையினதும் சராசரி கலோரி மதிப்பின் அடிப்படையில் எடுக்கப்பட்ட உணவின் படத்தில் உள்ள கலோரிகளின் எண்ணிக்கையை மதிப்பிடுவதற்கு உணவு கலோரிகளின் மதிப்பீட்டு முறையை நாங்கள் முன்மொழிகிறோம்.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

CHAPTER-1

INTRODUCTION

1.1. SYSTEM OVERVIEW

Well-being is becoming a topic of great interest and an essential factor linked to improvements in the quality of life.According to the World Health Organization, worldwide obesity has nearly tripled since 1975. In the India, almost 75% of the population is overweight and more than half of the population is obese (OECD). Today, many diseases that were previously thought as hereditary are now shown to be seen connected to biological malfunctions related to nutrition. Although being healthy and eating better is something the vast majority of the population want, doing so usually requires great effort and organization. The lack of an easy and simple way to track nutrition information about the food you eat can easily lead to low engagement. By providing a very easy and fun way to keep track of what the user eat, we can largely improve engagement, and directly attack on of the largest health problems in the world.Modern information technologies have brought a new dimension to this topic. Due to the widespread use of low-cost imaging devices like smart phone cameras, more and more applications are being developed in computer vision to facilitate automatic object recognition, among which food image recognition has recently gained much attention. Nowadays, people,especially diabetes patients, are increasingly cautious about their diet for improved health care. Food image recognition provides a simple means to estimate the dietary caloric intake and evaluate people's eating habits, by using cameras to keep track of their food consumption. An accurate estimation of daily nutritional intake provides a useful solution for keeping healthy and to prevent diseases. The application, which we focus on in this paper, is to build a computer program to automatically recognize foods from images of eating food and then estimate the energy of intake in terms of calorie amount. Development and deployment of such a computer program would provide a chance to facilitate medical treatment of obese patients and others by (semi)automatically logging food intakes and giving a summary of consumed food items and estimated calories per meal, day and so on. The program consists of one part that models certain foods of interest by analyzing a number of input images

and the other part that analyzes new images or outputs recognition results and calorie estimation.

1.2. OBJECTIVE OF THE SYSTEM

1. To recognize the food item present in the user input image.

2. To estimate the standard calories present in same food item.

1.3. EXISTING SYSTEM

Puriet al. proposed a pairwise classification framework that takes advantage of the user's speech input to enhance the food recognition process. Recognition is based on the combined use of color neighborhood and maximum response features in a texton histogram model, feature selection using Adaboost, and SVM classifiers. Texton histograms resemble BoF models, using though simpler descriptors, such that histograms of all possible feature vectors can be used. In this way, the feature vector clustering procedure can be omitted; however, less information is considered by the model which might not be able to deal with high visual variation. Moreover, the proposed system requires a colored checker-board captured within the image in order to deal with varying lighting conditions. In an independently collected dataset, the system achieved accuracies from 95% to 80%, as the number of food categories increases from 2 to 20.

DISADVANTAGES OF EXISTING SYSTEM:

- The used patches are sampled with the SIFT detector which is generally not a good choice for image classification problems, and described by the standard grayscale SIFT that ignores any color information.

- The last few years food recognition has attracted a lot of attention for dietary assessment, most of the proposed systems fail to deal with the problem of the huge visual diversity of foods, so they limit the visual dataset considered to either too few or too narrow food classes, in order to achieve satisfactory results.

- The BoF model appears to be an appropriate way for dealing with generic food description, but still there is no systematic investigation of

the various technical aspects related to feature extraction and classification.

1.4. **PROPOSED SYSTEM**

CNN BASED APPROACH FOR FOOD RECOGNITION:

We propose a multi-food image recognition system with CNN which can handle 100 kinds of food. As can be seen in Figure 1, this is a simple structure of the system. Given an input food image, first, the CNN can accomplish all the recognition steps including feature extraction, shift and distortion in-variance and classification, and then, gives the label.

After recognizing the food, the weight of the food is taken as input and the standard calorie value of the food for the given weight is calculated and displayed as output.

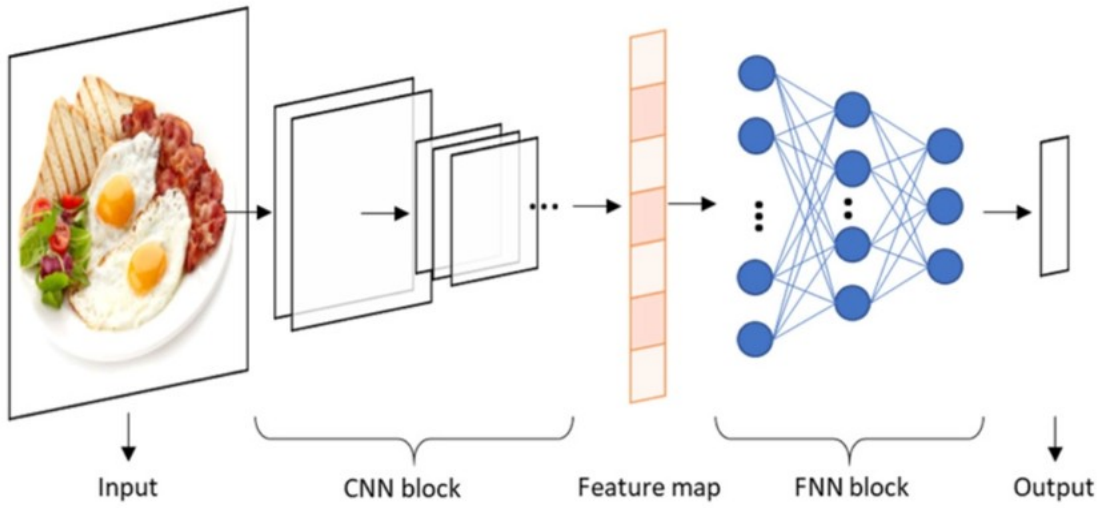


Fig-1.1: A simple structure for Food Image Recognition.

1.5. LITERATURE SURVEY

1.5.1. Using Graph Cut Segmentation for Food Calorie Measurement

Authors[1] have proposed combination of Graph cut and texture segmentation along with Support Vector Machine(SVM) classification model. Graph cut based method is efficient, robust and capable of finding the best contour of objects in an image. In this method, the object and the background pixels are grouped and identified individually in the image and then a cut is made to differentiate the food object from the background. Texture segmentation is used 659 to identify and extract texture features from the food image. The output from the segmentation method is given to SVM to classify the recognized food.

1.5.2 Using Distance Estimation and Deep Learning to Simplify Calibration in Food Calorie Measurement

Authors[2] have proposed a method which is user-friendly calibration of the dimension of the food portion sizes, which is needed in order to measure food portion weight and its ensuring amount of calories. In deep learning method, first we train our food images with a deep neural network, then we generate the model file. Every time the user submits a food picture for calorie measurement, the system performs segmentation and extracts features, which are further written into hidden layers in the deep neural network. Training the deep neural network in this way, will allow us to make the necessary changes to the weight and bias, without majorly affecting the output of the network and giving us the desired results. For Distance estimation, during the registration phase, the user is prompted to enter his height details (in feet or cm), which is used during the method to measure of the distance of food object from users phone. System then calculates the mobile phone's orientation using the rotation matrix and calculates angle to the target food object. Based on the angle value and the height entered by the user (example 165 cm), we are able to obtain the distance of the target food object from the mobile phone.

1.5.3 Food Calorie Measurement Using Deep Learning Neural Network

Authors[3] have proposed Deep Learning Neural Network method handles the training and testing requests at the top layers, without affecting the central layers. Firstly the segmentation is done using Graph Cut

segmentation followed by deep learning method. Here user's thumb is used for size calibration. In this method, the first step is to generate a pre-trained model file and then system is trained with positive set of images. In the second step, we re-train the system with the set of negative images (images that do not contain the relevant object). In this system, once the model file is generated from the training, we load it into the application and test it against the images captured and submitted by the user. The system then performs the image recognition process and generates a list of probabilities against the label name. The label with the highest probability is prompted to the user in the dialog box, to confirm the object name. Once the object name is confirmed, the system performs the calorie computation part by calculating the size of the food item with respect to the thumb in the frame. It finally prints the output to the user with the required calorie.

1.5.4 Automatic Food Recognition System for Diabetic Patients

In this paper, authors[5] approach is to capture food image and fed it into Dense SIFT method, this method extracts key point and visual vector from an image. Extracted visual vector are clustered using K-means clustering technique. Then SVM classifier is used for classification of food image and measures the carbohydrate level. In this system, Food recognition based Bag of Feature (BoF) model consists of two stages. First stage contains four steps; in this stage a set of characteristics representing the visual content of the image is extracted and quantified. This set provides input to the second stage, where a classifier assigns the image to one class out of a predefined set of food classes. The design and development of both stages involves two phases: training and testing. During the training phase, the system learns from the acquired knowledge, while during the testing phase the system recognizes food types from new, unknown images.

1.5.5 Measuring Calorie and Nutrition From Food Image

Authors[6] have proposed system in which images are taken by the user with a mobile device followed by a pre-processing step. Then at the segmentation step, each image will be analyzed to extract various segments of the food portion using color and texture segmentation. For each detected food portion a feature extraction process has to be performed. In this step various food features including size, shape, color, and texture will be extracted. The extracted features will be sent to the classification step where using the

support vector machine (SVM) scheme the food portion will be identified. Finally, by estimating the area of the food portion and using some nutritional tables the calorie value of the food will be extracted.

1.6. COMPARISONS OF IMPLEMENTED TECHNIQUES

S.No.	Title	Method Used	Advantages	Disadvantages
1.	Using Graph Cut Segmentation for Food Calorie Measurement	Graph-cut segmentation	Efficient , robust and finds best contour of objects	Problem in detecting mixed food because the size of food portions in mixed food are not similar.
2.	2. Using Distance Estimation and Deep Learning to Simplify Calibration in Food Calorie Measurement estimation	Deep learning method and Distance estimation method	Deep learning method and Distance estimation method	System is not able to find the real dimension of food object.

	.			
3.	Food Calorie Measurement Using Deep Learning Neural Network	Deep Learning Neural Network	Deep learning does not affect central layers and works on top layers.	Determining the dimension of food portion based on image captured, as it depends on the distance from which the photo was taken.
4.	Automatic Food Recognition System for Diabetic Patients.	Bag of feature (Dense SIFT and SVM) and K-mean clustering	K-mean clustering has been used to reduce time needed for both training and testing	Food items were recognized, it spent long time to find food names.
5.	Measuring Calorie and Nutrition From Food Image	Color and texture segmentation	Calculating the area of the food portion with thumb is more flexible, controllable and reliable.	System assumes that the depth of the food is uniform throughout the food's portion.

Table 1.1: Comparison of Implemented Techniques

CHAPTER-2
SYSTEM REQUIREMENTS AND SPECIFICATIONS

2.1. FUNCTIONAL REQUIREMENTS

A **software requirements specification** (**SRS**) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.The software requirements specification document lists sufficient and necessary requirements for the project development.

A **functional requirement** defines a **function** of a system or its component, where a **function** is described as a specification of behavior between outputs and inputs

There are set of requirements that needs to be satisfied while getting the input from the user:

1. The image capture should be possible and should not be much time consuming.

2. In order to get the results, the user should be following simple steps:

    i. The user should upload the food image into the food recognition model.

    ii. While the model computes,detects and recognizes the food and find the calories, the user must wait.

    iii. Once the model returns the result, output is displayed to the user.

3. The user gets the calorie amount present in the food.

2.2. INTERNAL REQUIREMENTS

- Building a food database is a starting point for developing and testing food recognition programs for obesity study.

- Fast foods, which are standardized and have exact calorie info published online, are chosen as food objects in the database collection.

- By collecting and studying fast food data, however, we do not claim or conclude any relation between fast foods and obesity in this paper.
- We expect that this available database and our preliminary results would benefit research communities to study obesity in terms of fast foods and image processing techniques.

2.3. OTHER NON-FUNCTIONAL REQUIREMENTS

- In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.
- A non-functional requirement defines the performance attribute of a software system.
- The environment should have a good lighting while capturing food image, because the more clearer the pixels the more accuracy in detection.
- The performance of the system depends upon the accuracy obtained.

CHAPTER-3

SYSTEM DESIGN

3.1. SYSTEM ARCHITECTURE

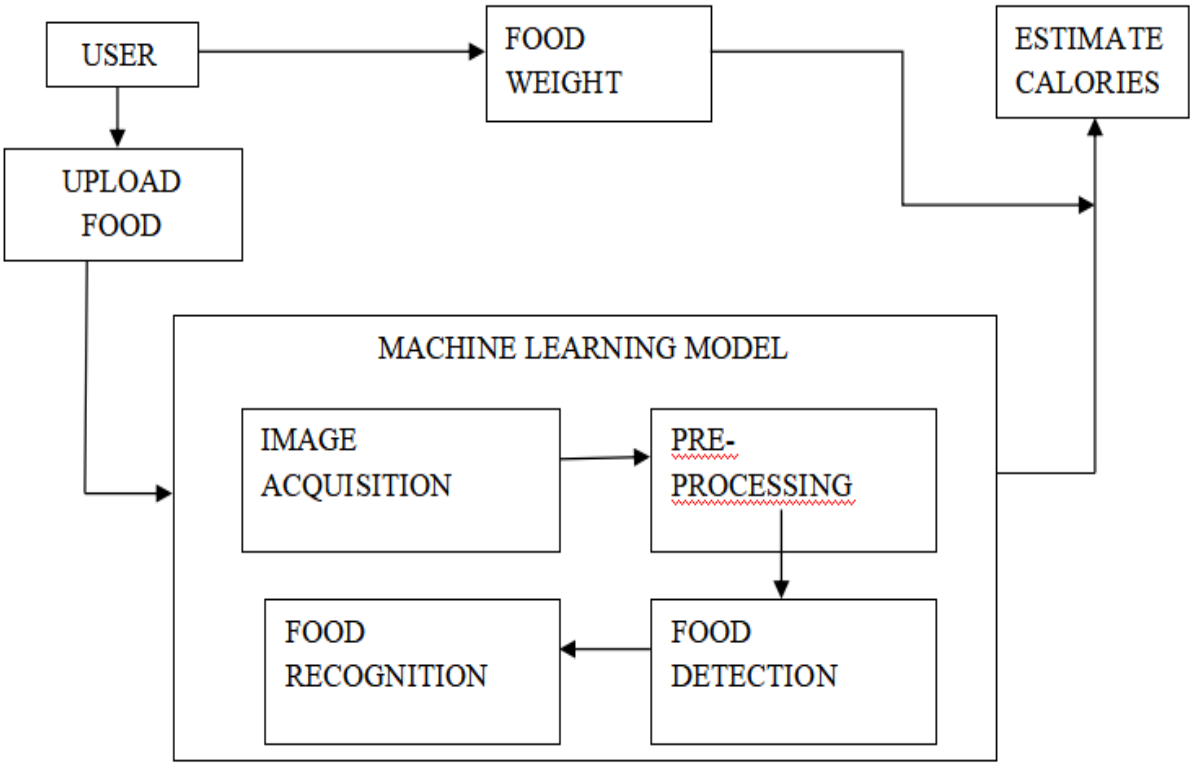


Fig-3.1: Architecture Diagram

1. The user uploads the food image and weight of the food into the machine learning model.

2.The machine learning model consists of 4 internal functions:

i. Image acquisition

ii. Pre-processing

iii. Food Recognition

iv. Food Detection

3.Now, the machine learning model returns the label and the weight is given to the model.

4.The model calculates the calorie value and gives to the user as output.

3.2. USE-CASE DIAGRAM:

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The use case diagram helps in finding the functional requirements of the project.

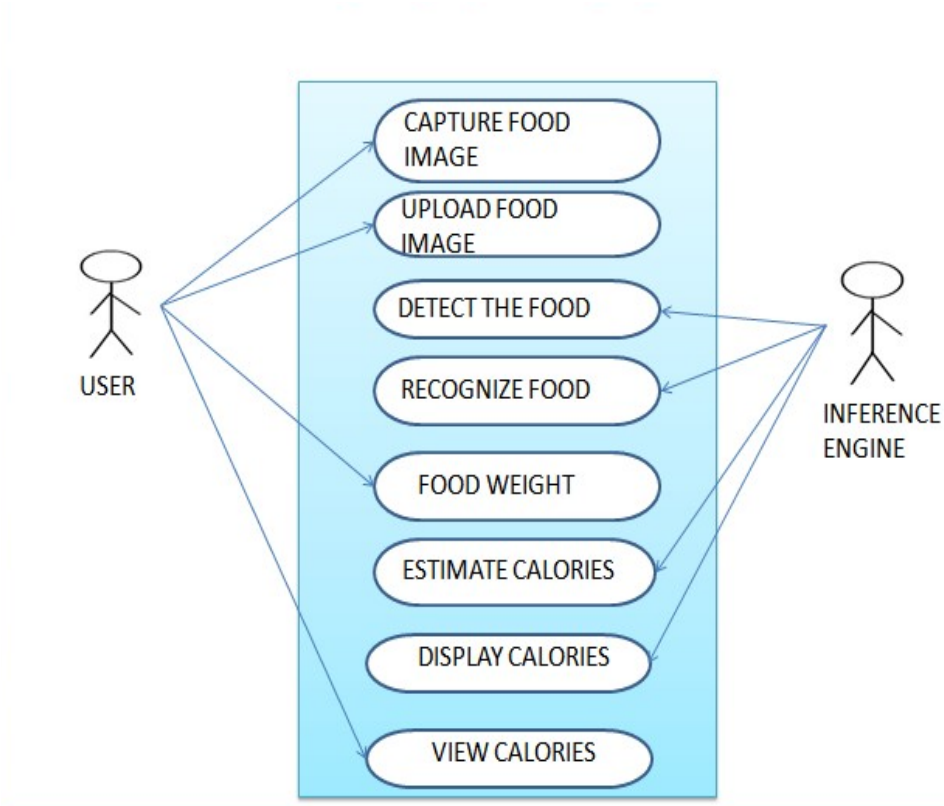


Fig-3.2: Use-case Diagram

3.3. ENTITY-RELATIONSHIP DIAGRAM:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

1. The user should upload the food image into the food recognition model.

2. The Model recognizes the food and the amount of calories present in the food.

3. The output is displayed on the user interface(UI)

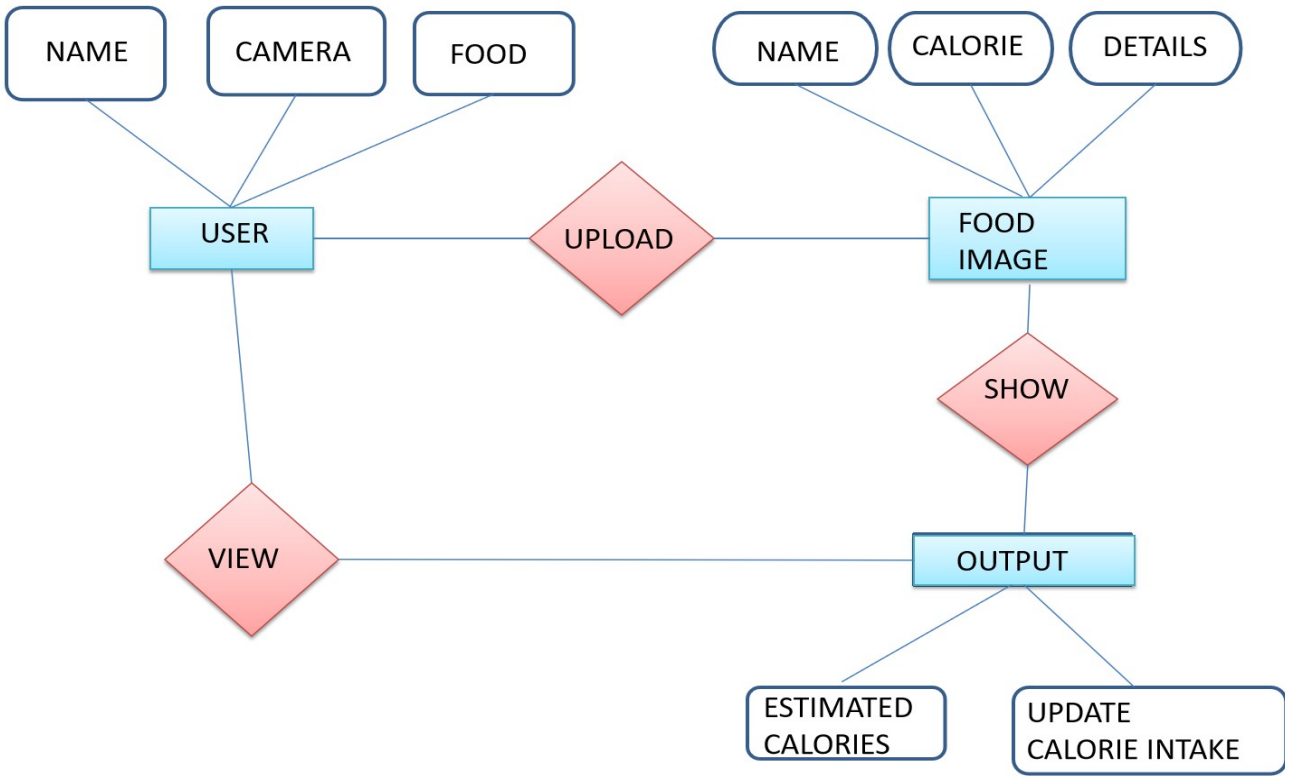


Fig-3.3:Entity-Relationship Diagram

CHAPTER-4

IMPLEMENTATION

4.1. METHODOLOGY

The project consist of two steps:identifying food from an image,and converting the food identified into a calorie estimation.We performed food image classification using CNN(Convolutional Neural Network).

Steps followed:

● **Preprocessing :** Some basic preprocessing has been performed to clean the dataset where the irrelevant and noisy images of 15 categories have been removed. Also, data augmentation has been performed -

○ Pixel values rescaled in the range of [0,1].

○ Random rotations upto 40 degree.

○ Random zoom applied.

○ Shear angle in counter-clockwise direction in degrees

● **Trained the model :** We trained the model of images for 15 categories using the classifier CNN(Convolutional Neural Network) which is a class of deep, feed forward artificial neural networks that has successfully been applied to analyzing visual imagery.

4.2. CONVOLUTIONAL NEURAL NETWORK

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that *convolve* with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution.The Convolutional Neural Network (CNN) offers a technique for many general image classification problems. It has been applied in food classification and resulted in a high accuracy.CNN is widely used in food recognition and provides better performance than the conventional methods.

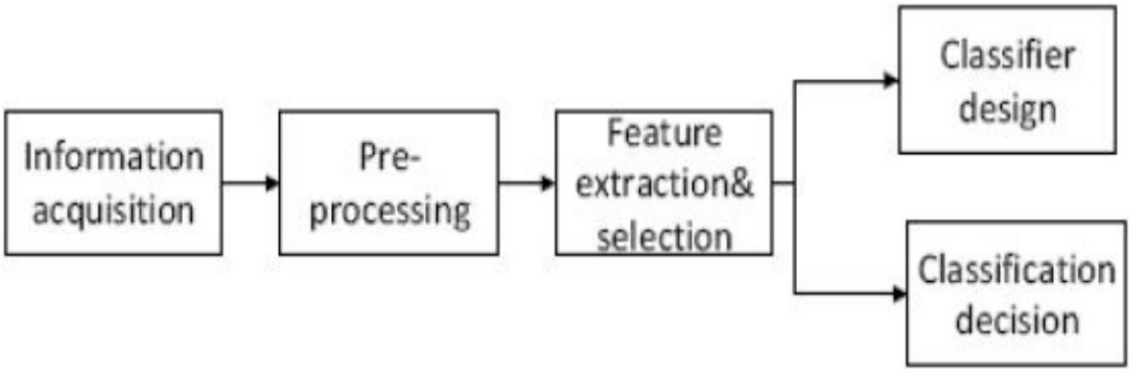


Fig-4.1: Tradition Model and CNN

Over the last few years, due to the advancements in the deep learning, especially in the convolutional neural networks, the accuracy in identifying and recognizing images has been increased drastically. This is not only because larger datasets but also new algorithms and improved deep architectures. Convolutional Neural Network (CNN) is also known as LeNet due to its inventor.CNN mainly comprises convolutional layers, pooling layers and sub-sampling layers followed by fully-connected layers. The very first architecture of CNN takes an input image and applies convolution followed by sub-sampling. After two such computations, the data is fed into the fully connected neural network,where it performs the classification task. The main advantage of CNN is the ability to learn the high-level efficient features and in addition to that, it is robust against small rotations and shifts.

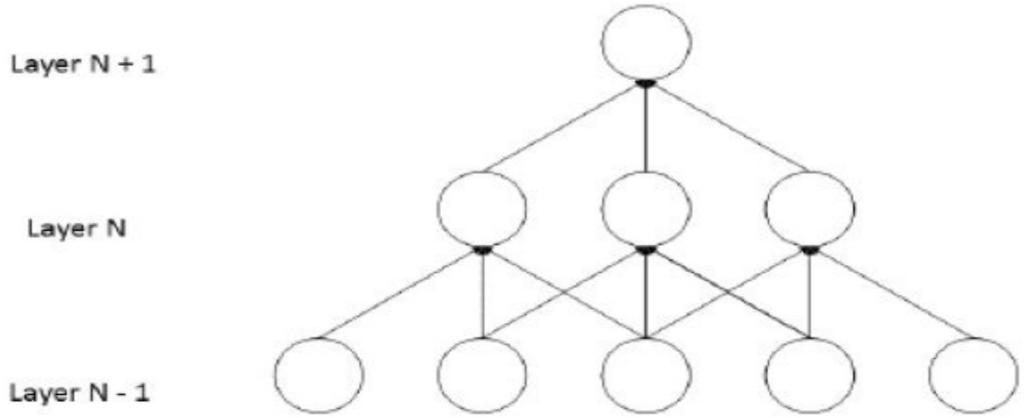


Fig-4.2: The schematic diagram of sparse connectivity

The schematic diagram of sparse connectivity In pattern recognition perspective image recognition can be abstracted into the form in below figure.

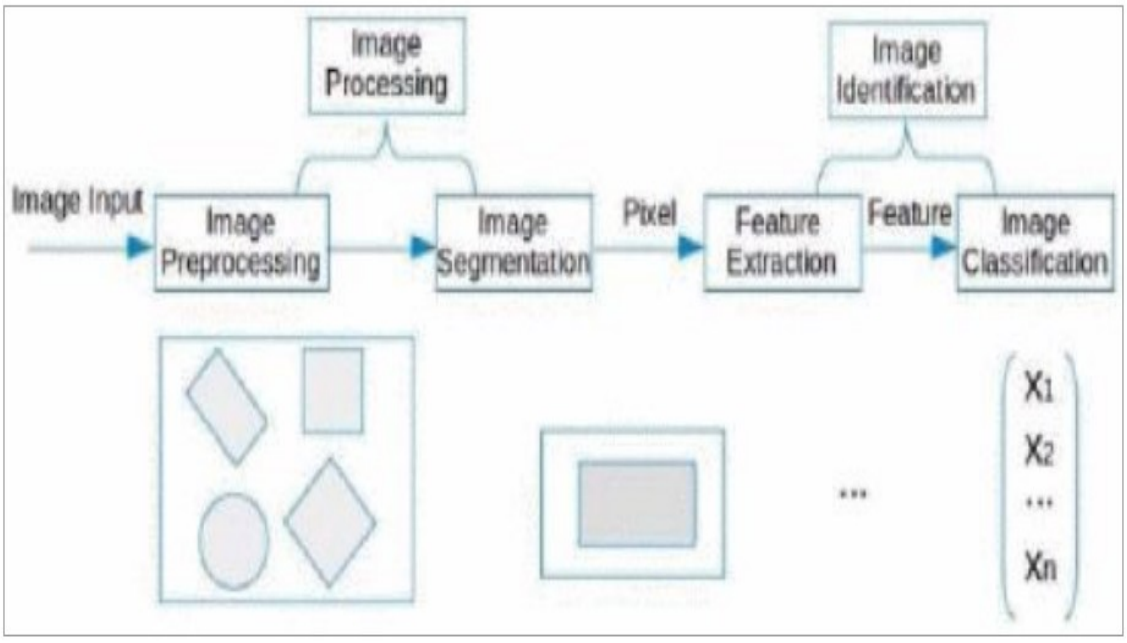


Fig-4.3:Four Steps Of Image Segmentation

Four steps of image recognition specific to the field of image recognition the process can be summarized in below figure.

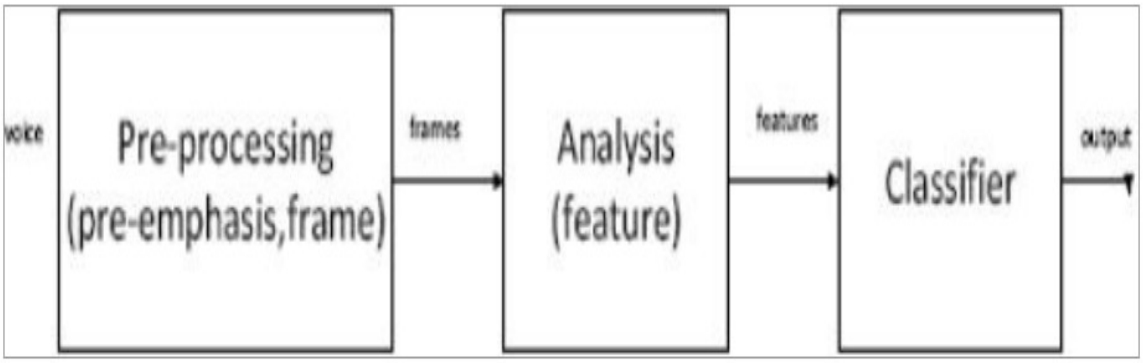


Fig-4.4: Classifier Phases

4.3. FOOD 101 DATASET

The dataset contains a number of different subsets of the full food-101 data. For this reason the data includes massively downscaled versions of the images to enable quick tests. The data has been reformatted as HDF5 and specifically Keras HDF5Matrix which allows them to be easily read in. The file names indicate the contents of the file.There are 101 categories represented, with 1000 images, and most have a resolution of around 512x512x3 (RGB, uint8). We have used 15 categories in our project. They are Apple Pie, Club Sandwich, Grilled Cheese Sandwich, Tacos, Hamburger, Samosa, French Fries, Pizza, Ravioli, Cake, Spring Rolls.

4.4. PACKAGES USED

(i) KERAS

Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.* Keras has the following key features:

● Allows the same code to run on CPU or on GPU, seamlessly.

● User-friendly API which makes it easy to quickly prototype deep learning models.

● Supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, etc. This means that Keras is appropriate for building essentially any deep learning model, from a memory network to a neural Turing machine.

● Is capable of running on top of multiple back-ends including Tensorflow, CNTK, or Theano.

(ii) JPEG

This package provides an easy and simple way to read, write and display bitmap images stored in the JPEG format. It can read and write both files and in-memory raw vectors.

(iii) CARET

The Caret package (short for _C_lassification _A_nd _RE_gression _T_raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

● data splitting

● pre-processing

● feature selection

● model tuning using re-sampling

● variable importance estimation

(iv) SHINY

Shiny is an R package that makes it easy to build interactive web apps straight from R. This package can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards This package can also extend Shiny apps with CSS themes, html widgets, and JavaScript

actions. It is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.

(v) SHINYJS

Perform common useful JavaScript operations in Shiny apps that will greatly improve your apps without having to know any JavaScript. Examples include:

Hiding an element, disabling an input, resetting an input back to its original value, delaying code execution by a few seconds, and many more useful functions for both the end user and the developer. 'shinyjs' can also be used to easily call your own custom JavaScript functions from R.

(vi) SHINYTHEMES

Bootstrap can use alternative CSS files in place of the stock bootstrap.css file.When the shinythemes package is loaded, it makes these alternate themes available to Shiny applications in a relative URL under shinythemes.Themes for use with Shiny. Includes several Bootstrap themes from which they are packaged for use with Shiny applications

(vii) DEVTOOLS

The aim of **devtools** is to make your life as a package developer easier by providing **R** functions that simplify many common tasks. .Devtools makes package development a breeze: it works with R's existing conventions for code structure, adding efficient tools to support the cycle of package development. With devtools, developing a package becomes so easy that it will be your default layout whenever you're writing a significant amount of code.

(viii) RSCONNECT

The rsconnect package provides a programmatic deployment interface for RPubs,shinyapps.io, and RStudio Connect. Supported contents types include R Markdown documents, Shiny applications, plots, and static web content.

(ix) CATOOLS

Contains several basic utility functions including: moving (rolling, running) window statistic functions, read/write for GIF and ENVI binary files, fast calculation of AUC, LogitBoost classifier, base64 encoder/decoder, round-off-error-free sum and cumsum, etc.

(x) E1071

Functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier.

(xi) KNN

This function provides a formula interface to the existing knn() function of package class. On top of this type of convenient interface, the function also allows normalization of the given data.

- n *k-NN classification*, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors.

4.5. MODEL ARCHITECTURE

The first layer is the Convolutional 2D layer which consists of 32 kernels of size 3x3 taking an input of size 100x100x3 where 100x100 is the rescaled size of our images and 3 denotes the color aspect (RGB) of the image. The

next layer is the max pooling layer with a pool size of 2x2. The above two layers are again repeated to get better filtered convolved images and better feature extraction by the max pooling layer. The above layers have been repeated one last time where the kernels have been increased from 32 to 64 to get more filtered images for the fully connected layers.Two fully connected layers are used next with 128 and 90 neurons respectively and Dropouts have been added of 0.01 in between the dense layers to prevent overfitting by making the weights of some random neurons to zero so as to prevent overfitting on some particular neurons.All the convolutional 2D layers and the fully connected layers have an activation function of ReLu (Rectified Linear Unit). The last layer is the output layer consisting of 15 neurons equivalent to the number of categories and each neuron has an output of a probability corresponding to that particular neuron. The CNN predicts the category to be the one with the highest probability.

CHAPTER-5

TESTING

5.1. SYSTEM TESTING AND MAINTANANCE

SYSTEM TESTING is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

What do you verify in System Testing:

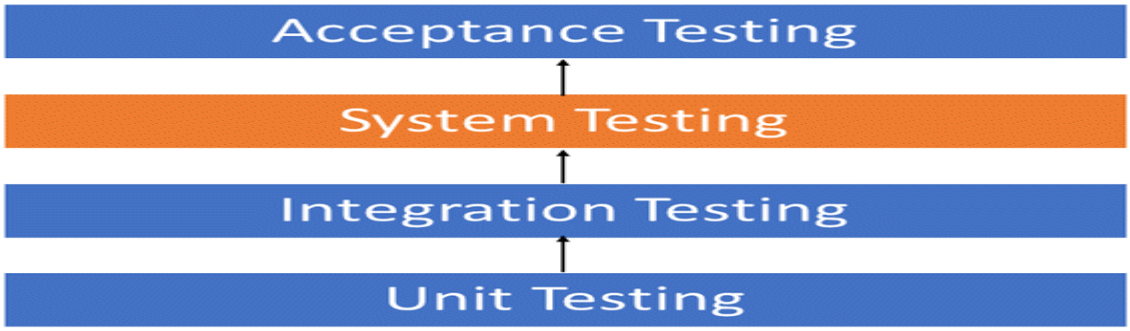
System Testing involves testing the software code for following

- Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.

- Verify thorough testing of every input in the application to check for desired outputs.
- Testing of the user's experience with the application.

That is a very basic description of what is involved in system testing. You need to build detailed test cases and test suites that test each aspect of the application as seen from the outside without looking at the actual source code.

5.2. Software Testing Hierarchy



As with almost any software engineering process, software testing has a prescribed order in which things should be done. The following is a list of software testing categories arranged in chronological order. These are the steps taken to fully test new software in preparation for marketing it:

- Unit testing performed on each module or block of code during development. Unit Testing is normally done by the programmer who writes the code.
- Integration testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model.
- System testing done by a professional testing agent on the completed software product before it is introduced to the market.

- Acceptance testing - beta testing of the product done by the actual end users.

5.3. Different Types of System Testing

There are more than 50 types of System Testing. For an exhaustive list of software testing types click here. Below we have listed types of system testing a large software development company would typically use

1. Usability Testing- mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives
2. Load Testing- is necessary to know that a software solution will perform under real-life loads.
3. Regression Testing- involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
4. Recovery testing - is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.
5. Migration testing- is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
6. Functional Testing - Also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.
7. Hardware/Software Testing - IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing.

5.3.1. Types of System Testing We Use:

There are over 50 different types of system testing. The specific types used by a tester depend on several variables. Those variables include:

- Who the tester works for - This is a major factor in determining the types of system testing a tester will use. Methods used by large companies are different than that used by medium and small companies.
- Time available for testing - Ultimately, all 50 testing types could be used. Time is often what limits us to using only the types that are most relevant for the software project.
- Resources available to the tester - Of course some testers will not have the necessary resources to conduct a testing type. For example, if you are a tester working for a large software development firm, you are likely to have expensive automated testing software not available to others.
- Software Tester's Education- There is a certain learning curve for each type of software testing available. To use some of the software involved, a tester has to learn how to use it.

Testing Budget - Money becomes a factor not just for smaller companies and individual software developers but large companies as well.

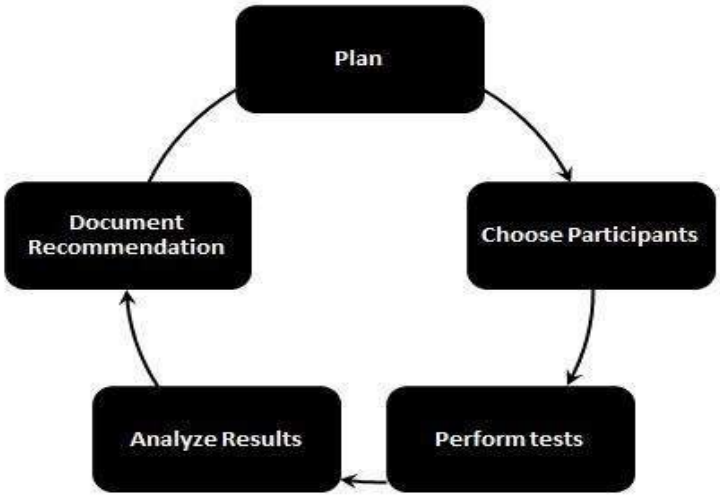
5.4. TYPES OF TESTING

1. Usability Testing
2. Integration Testing
3. Functional Testing
4. System Testing
5. White Box Testing
6. Black Box Testing
7. Alpha Testing
8. Acceptance Testing

5.4.1. USABILITY TESTING:

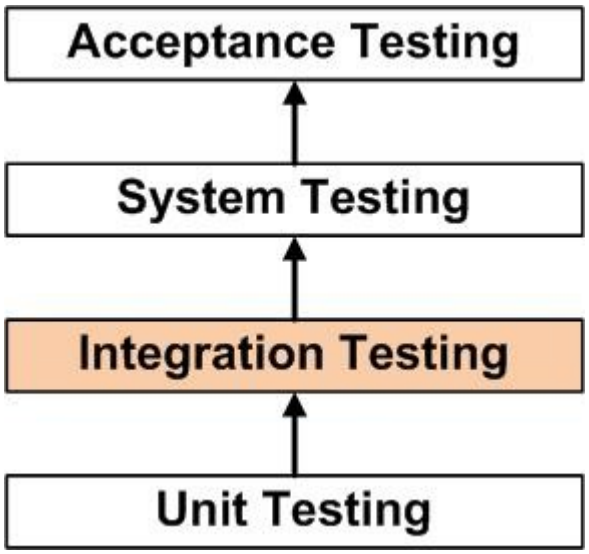
Usability testing, a non-functional testing technique that is a measure of how easily the system can be used by end users. It is difficult to evaluate and measure but can be evaluated based on the below parameters:

- Level of Skill required to learn/use the software. It should maintain the balance for both novice and expert user.
- Time required to get used to in using the software.
- The measure of increase in user productivity if any.
- Assessment of a user's attitude towards using the software.



5.4.2. INTEGRATION TESTING:

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.



- **integration testing:** Testing performed to expose defects in the interfaces and in the
  interactions between integrated components or systems. See also component integration
  testing, system integration testing.
- **component integration testing:** Testing performed to expose defects in the interfaces and
  interaction between integrated components.
- **system integration testing:** Testing the integration of systems and packages; testing
  interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

5.4.3. FUNCTIONAL TESTING:

Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications.

Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

During functional testing, Black Box Testing technique is used in which the internal logic of the system being tested is not known to the tester.

Functional testing is normally performed during the levels of System Testing and Acceptance Testing.

Typically, functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function’s specifications.
- Determine the output based on the function’s specifications.
- Execute the test case.
- Compare the actual and expected outputs.

Functional testing is more effective when the test conditions are created directly from user/business requirements. When test conditions are created from the system documentation (system requirements/ design documents),

the defects in that documentation will not be detected through testing and this may be the cause of end-users' wrath when they finally use the software.

5.4.4. WHITE BOX TESTING:

White Box Testing is testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

It is one of two parts of the **Box Testing** approach to software testing. Its counterpart, **Blackbox testing**, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

5.4.5. BLACK BOX TESTING:

Black Box Testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications. In BlackBox Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

5.4.6. ALPHA TESTING:

Alpha Testing is defined as a type of software testing performed to identify bugs before releasing the product to real users or the public. It is a type of acceptance testing.

This testing is referred to as an alpha testing only because it is done early on, near the end of the development of the software, and before Beta Testing. Check Differences between Alpha testing and Beta testing

The main objective of alpha testing is to refine the software product by finding (and fixing) the bugs that were not discovered through previous tests.

Alpha testing is typically performed by in-house software engineers or QA staff. It is the final testing stage before the software is released into the real world.

Who is involved in Alpha testing?

Alpha testing has two phases,

1. The first phase of testing is done by in-house developers. They either use hardware-assisted debuggers or debugger software. The aim to catch bugs quickly. Usually while alpha testing, a tester will come across to plenty of bugs, crashes, missing features, and docs.
2. While the second phase of alpha testing is done by software QA staff, for additional testing in an environment. It involves both black box and White Box Testing.

5.4.7. ACCEPTANCE TESTING:

ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

5.5. ACCURACY TESTING

The input images were scaled down to 80 × 80 and cropped to 64 × 64 pixels randomly by the cuda-convnet python module. For comparison, the baseline was the food detection system.This method uses an SVM as the classifier and employs color features, circular boundary features, and SIFT–BoWfeatures as image features. In our experiment using CNN,we divided all food and non-food images in the dataset into10 groups. Eight groups were used for training, one for validation and one for testing. Training ended when the validation error ceased to evolve. We used the hyper parameters marked with "*" in Table 2. We conducted 10-fold cross validation. For the experiment using the baseline method,we randomly selected the same number (1,200) of food im-ages and non-food images, with 1,000 being used for trainingthe SVM and the remaining 200 being used for testing. Wecomputed the average result over 15 trials.

Layers	Number of kernels	Size of kernels	Normalization (LRN)	Dataset	Avg.	SD
2	32-32	9-7	1 time	Same dataset and parameters	72.39%	1.48%
*2	32-32	9x9, 7x7	1 time	6-fold cross validation	71.17%	0.83%
2	32-64	9x9, 7x7	1 time	6-fold cross validation	72.94%	0.68%
2	64-64	9x9, 7x7	1 time	6-fold cross validation	70.07%	2.98%
3	32-32-64	5x5, 5x5, 5x5	2 times	6-fold cross validation	69.82%	0.73%
4	32-32-32-64	7x7, 7x7, 5x5, 2x2	3 times	6-fold cross validation	66.57%	0.91%
2	32-32	7x7, 5x5	1 time	6-fold cross validation	72.86%	1.35%
2	32-32	5x5, 3x33	1 time	6-fold cross validation	72.88%	1.68%
2	32-32	5x5, 5x5	1 time	6-fold cross validation	73.69%	1.28%
2	32-32	5x5, 5x5	1 time	6-fold cross validation	**73.70%**	0.81%
SPM + Color + SVM				6-fold cross validation	54.63%	N.A.
GIST + SVM				6-fold cross validation	52.53%	N.A.
ScSPM				6-fold cross validation	60.47%	N.A.

Table 5.1: Accuracy of CNNs for various hyper-parameter values compared with three existing techniques

Table 3 compares the accuracy of the baseline method andCNN, with CNN achieving 93.8% accuracy, which is significantly higher than that for the baseline method.

Method	Accuracy
Baseline	$89.7 \pm 0.73\%$
CNN	$\mathbf{93.8 \pm 1.39\%}$

Table 5.2: Comparison between CNN and the baseline method for the detection task

6. FUTURE SCOPE

- The web application can be converted into mobile app for more user convenience
- More categories of food can be trained in future
- Multiple layers in single food item can be recognized
- Calorie can be estimated with the help of volume

7.CONCLUSIONS

In practice, the traditional models in machine learning are not attaining much accuracy when it comes to image classification. In this project, the CNN model is applied in image recognition.Much of data augmentation and segmentation has to be performed as well and clean pixel values are not necessary in CNN as it on it's own learn the generalized pattern required to identify and recognize new images.So using CNN model, the accuracy is comparatively a lot higher than all other traditional models.

APPENDIX-1

CODING:

The User Interface

```
#install.packages('devtools'
)
                                #devtools::install_github("rstudio/keras")

                                #install.packages('reticulate')

                                #library(keras)

                                #install_keras()

                                #install.packages('shinyjs')

                                #install.packages('shinythemes')

                                #install.packages('shinydashboard')

                                library(shiny)

                                library(shinyjs)

                                library(shinythemes)


                                #to do -> upload it on shiny.io

                                fluidPage(

                                shinyjs::useShinyjs(),

                                theme = shinytheme("slate"),

                                titlePanel("Food Recognition and Calorie Estimation"),

                                sidebarLayout(

                                sidebarPanel(

                                fluidPage(h4("Hey! Want to calculate calories of the food you're
                                eating?"),br()),

                                fluidRow(

                                fileInput("myFile", "Choose a file", accept = c('image/png', 'image/
                                jpeg'))

                                ),

                                actionButton("submit","Recognize")
```

```
,
actionButton("reset", "Reset"),
fluidPage(br()),
imageOutput("myImage"),
shinythemes::themeSelector()
), mainPanel(

# Output: Data file ----
#img(src="food.png"),
includeHTML("slideshow.html"),
h2(textOutput("result1")),
h4(textOutput("result3")),
h4(textOutput("result2"))




)
)
)
```

For the SlideShow:

```
<!DOCTYPE
html>
        <html>
        <head>
        <meta name="viewport" content="width=device-width, initial-
        scale=1">
        <style>
        * {box-sizing: border-box;}
```

```
body {font-family: Verdana, sans-serif;}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
max-width: 1000px;
position: relative;
margin: auto;
}

/* Caption text */
.text {
color: #f2f2f2;
font-size: 15px;
padding: 8px 12px;
position: absolute;
bottom: 8px;
width: 100%;
text-align: center;
}

/* Number text (1/5 etc) */
.numbertext {
color: #f2f2f2;
font-size: 12px;
padding: 8px 12px;
position: absolute;
```

```
top: 0;
}

/* The dots/bullets/indicators */
.dot {
height: 15px;
width: 15px;
margin: 0 2px;
background-color: #bbb;
border-radius: 50%;
display: inline-block;
transition: background-color 0.6s ease;
}

.active {
background-color: #717171;
}

/* Fading animation */
.fade {
-webkit-animation-name: fade;
-webkit-animation-duration: 1.5s;
animation-name: fade;
animation-duration: 1.5s;
}

@-webkit-keyframes fade {
from {opacity: .4}
```

```
to {opacity: 1}
}

@keyframes fade {
from {opacity: .4}
to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
.text {font-size: 11px}
}
</style>
</head>
<body>


<div class="slideshow-container">

<div class="mySlides fade">
<div class="numbertext">1 / 5</div>
<img src="pizza.jpg" style="width:101%">
<div class="text"><h3>Pizza</h3></div>
</div>

<div class="mySlides fade">
<div class="numbertext">2 / 5</div>
<img src="burger.jpg" style="width:101%">
```

```
<div class="text"><h3>Burger</h3></div>
</div>

<div class="mySlides fade">
<div class="numbertext">3 / 5</div>
<img src="cake.jpg" style="width:101%">
<div class="text"><h3>Cake</h3></div>
</div>

<div class="mySlides fade">
<div class="numbertext">4 / 5</div>
<img src="fries.jpg" style="width:101%">
<div class="text"><h3>Fries</h3></div>
</div>

<div class="mySlides fade">
<div class="numbertext">5 / 5</div>
<img src="waffles.jpg" style="width:101%">
<div class="text"><h3>Waffles</h3></div>
</div>

</div>
<br>

<div style="text-align:center">
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
```

```
<span class="dot"></span>
<span class="dot"></span>
</div>

<script>
var slideIndex = 0;
showSlides();

function showSlides() {
var i;
var slides = document.getElementsByClassName("mySlides");
var dots = document.getElementsByClassName("dot");
for (i = 0; i < slides.length; i++) {
slides[i].style.display = "none";
}
slideIndex++;
if (slideIndex > slides.length) {slideIndex = 1}
for (i = 0; i < dots.length; i++) {
dots[i].className = dots[i].className.replace(" active", "");
}
slides[slideIndex-1].style.display = "block";
dots[slideIndex-1].className += " active";
setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>

</body>
</html>
```

Code To Train Model

```
# Installing Theano

# pip install --upgrade --no-deps git+git://github.com/Theano/Theano.git


# Installing Tensorflow

# Install Tensorflow from the website:
https://www.tensorflow.org/versions/r0.12/get_started/os_setup.html


# Installing Keras

# pip install --upgrade keras


# Part 1 - Building the CNN


# Importing the Keras libraries and packages

from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense


# Initialising the CNN

classifier = Sequential()


# Step 1 - Convolution

classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation =
'relu'))
```

```
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a second convolutional layer
classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Step 3 - Flattening
classifier.add(Flatten())

# Step 4 - Full connection
classifier.add(Dense(output_dim = 128, activation = 'relu'))
classifier.add(Dense(output_dim = 128, activation = 'relu'))
classifier.add(Dense(output_dim = 128, activation = 'relu'))
classifier.add(Dense(output_dim = 3, activation = 'softmax'))

# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics
= ['accuracy'])

# Part 2 - Fitting the CNN to the images

from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('UPMC_Food101/train',
target_size = (64, 64),
batch_size = 32,
class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('UPMC_Food101/test',
target_size = (64, 64),
batch_size = 32,
class_mode = 'categorical')

classifier.fit_generator(training_set,
samples_per_epoch = 2030,
nb_epoch = 20,
validation_data = test_set,
nb_val_samples = 679)

classifier.summary()

from keras.models import load_model
classifier.save('my_model.h5') # creates a HDF5 file 'my_model.h5'
#del classifier # deletes the existing model
# returns a compiled model
# identical to the previous one
classifier2 = load_model('my_model.h5')

classifier2.summary()
res=classifier.predict(k)
```

```
model <- keras_model_sequential()

layer_conv_2d(model,32,3,3,input_shape = c(64,64,3),activation='relu')

layer_max_pooling_2d(model,pool_size=c(2,2))

layer_conv_2d(model,64,3,3,activation='relu')

layer_max_pooling_2d(model,pool_size=c(2,2))

layer_flatten(model)

layer_dense(model,units = 128,activation = 'relu')
layer_dropout(model,rate=0.01)
layer_dense(model,units = 90,activation = 'relu')
layer_dropout(model,rate=0.01)
layer_dense(model,units = 5,activation = 'softmax')

model %>% compile(
loss = 'categorical_crossentropy',
optimizer = 'adam',
metrics = c('accuracy')
)

train_datagen=image_data_generator(rescale = 1./255, rotation_range = 40, shear_range = 0.2,
```

```
zoom_range = 0.2, horizontal_flip = TRUE)

test_datagen=image_data_generator(rescale = 1./255)


training_set=flow_images_from_directory('fdataset3/train',generator = train_datagen, target_size
= c(64,64),batch_size = 32, class_mode = 'categorical')

test_set=flow_images_from_directory('fdataset3/test',generator = test_datagen, target_size =
c(64,64),batch_size = 32, class_mode = 'categorical')


final=fit_generator(model,training_set,steps_per_epoch = 179,epochs=30,validation_data =
test_set)
```

Model Prediction

```
prediction<-
function(x){

        k=load_model_hdf5("my_model3.h5", custom_objects = NULL, compile =
        TRUE)

        img=image_load(x,target_size = c(100,100))

        data=image_to_array(img)

        dim(data) <- c(1, 100, 100, 3)

        ans=predict(k,data)

        return (ans)

        }
```

Server.R

```
source("model_prediction.R"
)

                library(shiny)

                library(keras)
```

```
getwd()
server <- function(input, output,session){

#Can save a file to a directory
observeEvent(input$myFile, {
inFile <- input$myFile
if (is.null(inFile))
return()
file.copy(inFile$datapath,
file.path(paste(getwd(),"/PredictImages",sep=""), inFile$name) )
# file.copy("F:/FinalMinor/Minor/PredictImages",inFile$name)

})

output$myImage <- renderImage({
inFile <- input$myFile
if (is.null(inFile))
return(list(src="www/insert.png",contentType="image/png"))
else{
list(
src = inFile$datapath,
width=256,
height=256,
contentType = "image/jpeg",
alt = "Face"
)}
},deleteFile = FALSE )
```

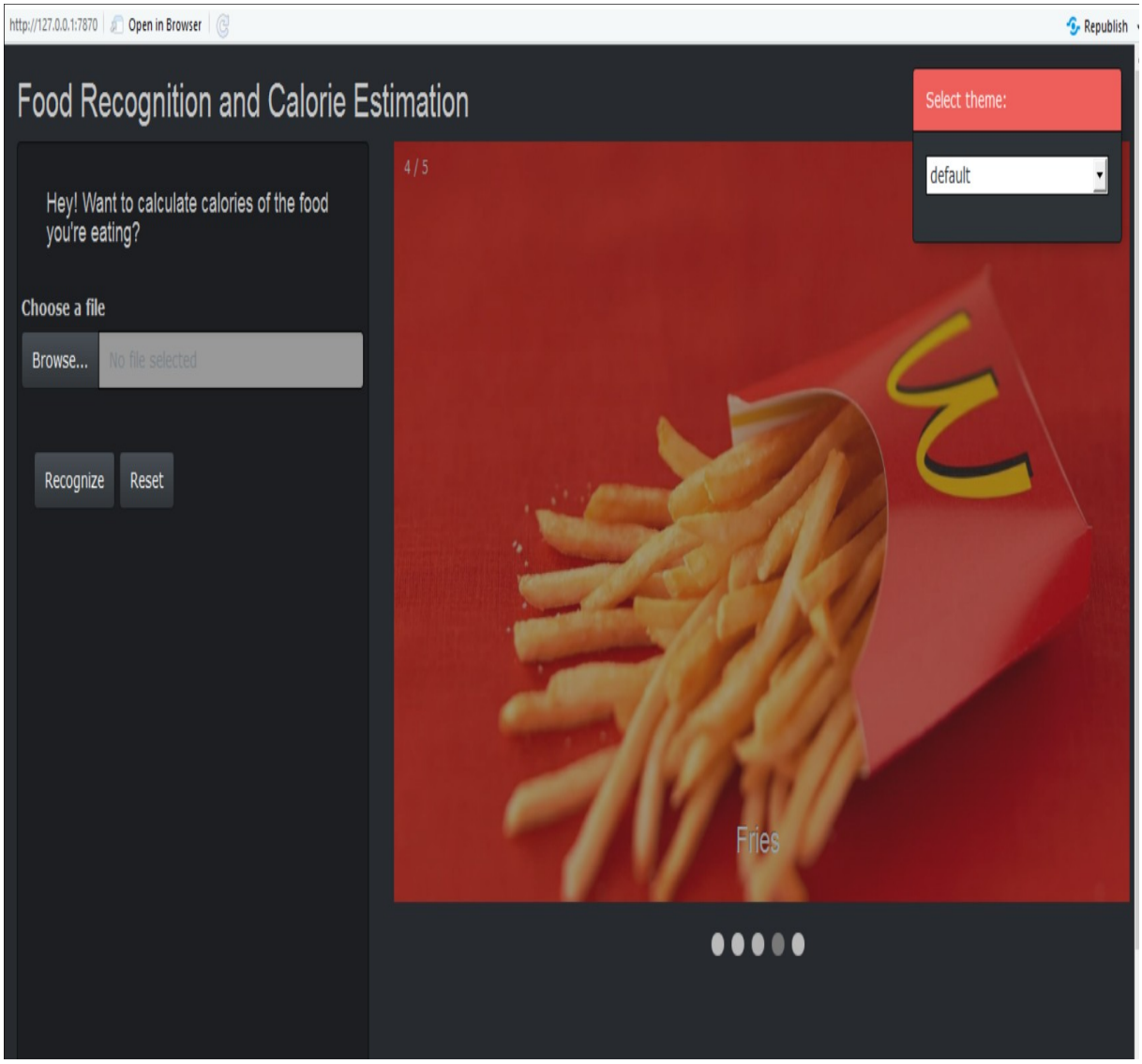
```
observeEvent(input$submit, {

inFile <- input$myFile

if(is.null(inFile))

output$myText<-renderText("No Image Uploaded")

else

{

res<-prediction(paste(getwd(),"/PredictImages/",inFile$name,
sep=""))

#print(is.matrix(res))

#suppressWarnings( file.remove(getwd(),inFile$name))

detected_food<- which.max(res)

switch(

detected_food,

{

output_1<-"French fries"

output_2<-paste(round(runif(1,300,315),1)," Calories ")

output_3<-"French fries, chips, finger chips, or French-fried
potatoes are batonnet or allumette-cut deep-fried potatoes."

},

{

output_1<-"Hamburger"

output_2<-paste(round(runif(1,200,208),1)," Calories ")

output_3<-"A hamburger, beefburger or burger is a sandwich
consisting of one or more cooked patties of ground meat, usually
beef, placed inside a sliced bread roll or bun."

},

{

output_1<-"Pizza"

output_2<-paste(round(runif(1,260,268),1)," Calories ")

output_3<-"Pizza is a traditional Italian dish consisting of a
yeasted flatbread typically topped with tomato sauce and cheese
```

```
and baked in an oven. It can also be topped with additional
vegetables, meats etc."

},

{

output_1<-"Cake"

output_2<-paste(round(runif(1,255,260),1)," Calories ")

output_3<-"Cake is a form of sweet dessert that is typically
baked. In its oldest forms, cakes were modifications of breads,
but cakes now cover a wide range of preparations"



},

{

output_1<-"Waffles"

output_2<-paste(round(runif(1,290,299),1)," Calories ")

output_3<-"A waffle is a dish made from leavened batter or
dough that is cooked between two plates that are patterned to
give a characteristic size, shape and surface impression."

}

)



#res<-
rbind(res,c("french_fries","hamburger","Pizza","cake","waffles")
)

#res<-rbind(res,c("312 calories","204 calories","266
calories","257 calories","291 calories"))

#print(res)

#InsertRow(res,c("Fries","Ice cream","Pizza"),2)

#res[2,]<-c("Fries","Ice cream","Pizza")

output$result1 <- renderText(output_1)

output$result2<- renderText(output_2)

output$result3<-renderText(output_3)

}
```

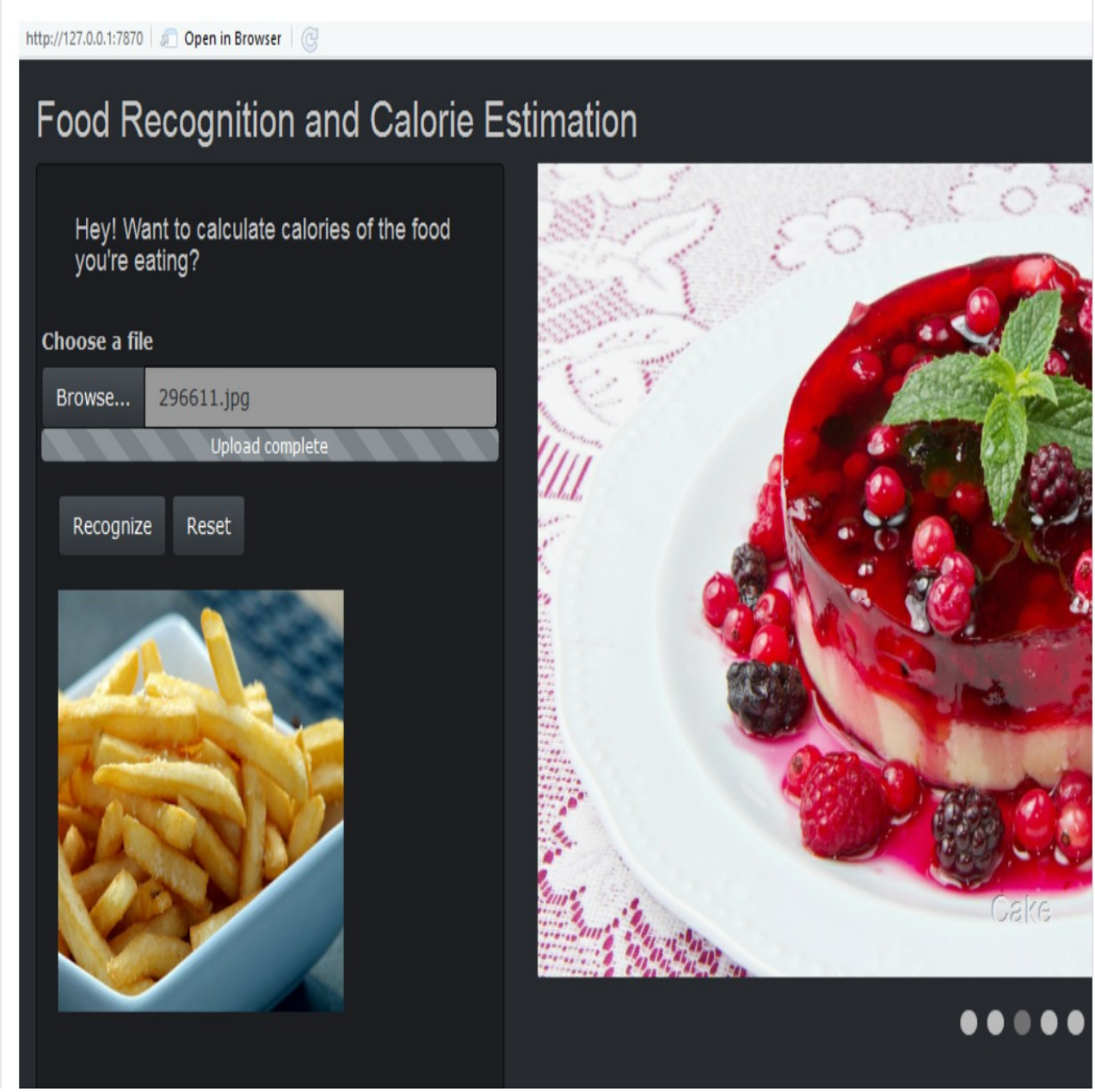
```
#shinyjs::alert("Thank you!")
}
)



observeEvent(input$reset, {
inFile <- input$myFile
output$result1 <- renderText("")
output$result2<-renderText("")
output$result3<-renderText("")
# file.copy(inFile$datapath, file.path( getwd(), inFile$name) )
# output$myImage <-renderImage(list())
})

}
```

APPENDIX-2

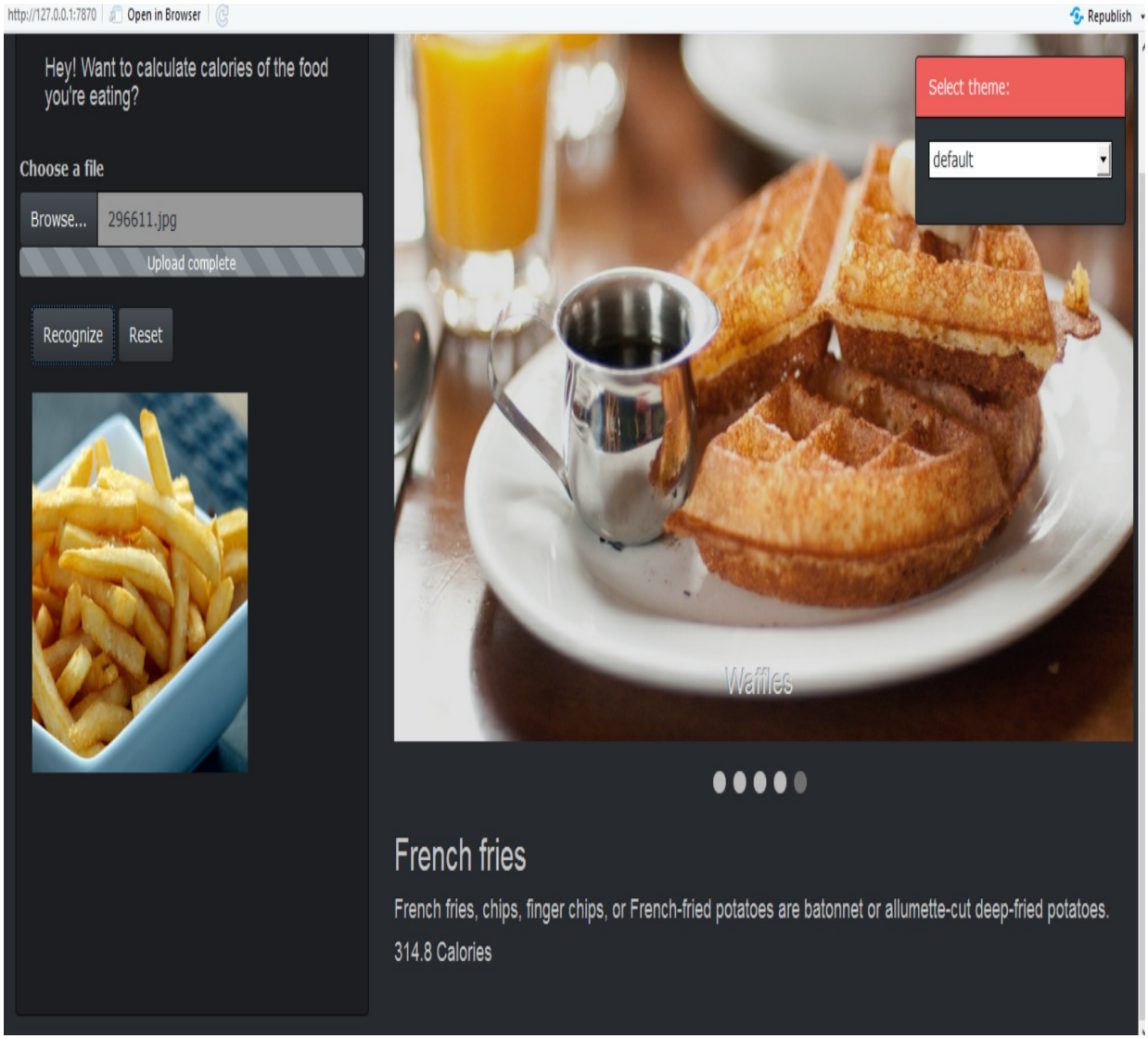
SCREENSHOTS



1. Before Uploading Food Image



2. After Uploading Food Image



3. Predicted Image

REFERENCES

[1] Pouladzadeh Parisa, Shervin Shirmohammadi, and Abdulsalam Yassine. "Using Graph Cut Segmentation for Food Calorie Measurement." 2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA), 2014, DOI:10.1109/memea.2014.6860137, pp 1-6.

[2] Kuhad, Pallavi, Abdulsalam Yassine, and Shervin Shimohammadi. "Using Distance Estimation and Deep Learning to Simplify Calibration in Food Calorie Measurement." 2015 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015, DOI:10.1109/civemsa.2015.7158594, pp 1-6.

[3] Pouladzadeh, Parisa, Pallavi Kuhad, Sri Vijay Bharat Peddi, Abdulsalam Yassine, and Shervin Shirmohammadi. "Food Calorie Measurement Using Deep Learning Neural Network." 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, 2016, DOI:10.1109/ i2mtc.2016.7520547, pp 1-6.

[4] Zhang, Weishan, Dehai Zhao, Wenjuan Gong, Zhongwei Li, Qinghua Lu, and Su Yang. "Food Image Recognition with Convolutional Neural Networks." 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015, DOI:10.1109/uic-atc-scalcom-cbdcom-iop.2015.139, pp 690-693.

[5] Velvizhy P., Pavithra, and A. Kannan. "Automatic Food Recognition System for Diabetic Patients." 2014 Sixth International Conference on Advanced Computing (ICoAC), 2014, DOI:10.1109/icoac.2014.7229735, pp 329- 34. Vol-2 Issue-6 2016 IJARIIE-ISSN(O)-2395-4396 3376 www.ijariie.com 662

[6] Pouladzadeh, Parisa, Shervin Shirmohammadi, and Rana Al-Maghrabi. "Measuring Calorie and Nutrition From Food Image."IEEE Trans. Instrum.

Meas. IEEE Transactions on Instrumentation and Measurement 63, no. 8 (2014), DOI:10.1109/tim.2014.2303533, pp 1947-1956..

[7] Kawano, Yoshiyuki, and Keiji Yanai. "Real-Time Mobile Food Recognition System." 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, DOI:10.1109/cvprw.2013.5, pp 1-7.

[8] Anthimopoulos, Marios, Joachim Dehais, Peter Diem, and Stavroula Mougiakakou. "Segmentation and Recognition of Multi-food Meal Images for Carbohydrate Counting." 13th IEEE International Conference on BioInformatics and BioEngineering, 2013, DOI:10.1109/bibe.2013.6701608, pp 1-4.

[9] Tammachat, Natta, and Natapon Pantuwong. "Calories Analysis of Food Intake Using Image Recognition." 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE), 2014, DOI:10.1109/iciteed.2014.7007901, pp 1-4.