DEEP LEARNING-BASED FOOD CLASSIFICATION AND NUTRITIONAL VALUE ESTIMATION

A PROJECT REPORT

Submitted by

ADITYA KRRISH K S
AJAY ATHITYA R
ARAVIND B

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

AUGUST 2020

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"DEEP LEARNING-BASED FOOD CLASSIFICATION AND NUTRITIONAL VALUE ESTIMATION"** is the bonafide work of **"ADITYA KRRISH KS, AJAY ATHITYA R, ARAVIND B",** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. R. ANITHA	**Ms. G. JANAKASUDHA**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
COMPUTER SCIENCE & ENGG	**COMPUTER SCIENCE & ENGG**

Submitted for the project viva-voce examination held on ……..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The change in our lifestyle has significantly impacted our health and increased obesity at an accelerating rate. This is a reflection to the risks in people's state of health. The elementary step to avoid obesity is to regulate the daily calorie consumption by having nutritious food. Even though, packaged foods come with nutritional and calorie value information it, it's not still very convenient for people to refer. For unpackaged foods the task becomes more complex. Thus, we should start depending on machine learning algorithms in computer vision to help us estimate the calorific value in foods. This project provides a simple yet efficient way of estimating calories from a food image. A CNN which is trained using transfer learning on Inception v3 model using Food 101 dataset with 86.6% training accuracy is used to classify the food. A smartphone-based application is employed for image acquisition. A novel lightweight ratio transfer algorithm is developed to estimate the size of food with reference to the calibration object and compute the total calories present in the food image.

ACKNOWLEDGEMENT

We thank our Principal **Dr. S. Ganesh Vaidyanathan,** Sri Venkateswara College of Engineering for being the source of inspiration throughout our study in this college.

We express our sincere thanks to **Dr. R. Anitha,** Head of the Department, **Computer Science and Engineering** for her encouragement accorded to carry out this project.

With profound respect, we express our deep sense of gratitude and sincere thanks to our guide **Ms. G. Janakasudha**, Assistant Professor, for her valuable guidance and suggestions throughout this project.

We are also thankful to our Project Co-ordinators **Dr. R.Jayabhaduri, Associate Professor, Dr. N.M.Balamurugan, Associate Professor and Ms.V.Rajalakshmi, Assistant Professor** for their continual support and assistance.

We thank our family and friends for their support and encouragement throughout the course of our graduate studies.

ADITYA KRRISH K S

AJAY ATHITYA R

ARAVIND B

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Networks
EXIF	Exchangeable Image File Format
ToF	Time of Flight
FCN	Fully Convolutional Networks
ICT	Information and Communications Technology
IOT	Internet of Things
SURF	Speeded up Robust Features
SPM	Spatial Pyramid Matching
ReLU	Rectified Linear Unit
RGB	Red Blue Green
GPU	Graphics Processing Unit
IP	Internet Protocol
IDE	Integrated Development Environment
OpenCV	Open Source Computer Vision
BSD	Berkeley Software Distribution
ML	Machine Learning
PIL	Python Imaging Library
VGG	Visual Geometry Group

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

High calorie food intake can be harmful and result in obesity, which is a preventable medical condition that causes abnormal accumulation of fat in the body. It can result in numerous diseases such as obesity, diabetes, cholesterol, heart attacks, blood pressure, breast, colon and prostate cancers and other diet-related ailments. In order to deal with such problems, people are inclined towards making a difference in their diet plans by paying more attention to what type of food they are consuming. Diet management is a key concern amongst individuals belonging to different age groups. However, one major challenge in diet management is to maintain a balance between what one eats and how one monitors his/her food consumption.

The immense increase in ailments such as high cholesterol, blood pressure, strokes etc. demand for nutritional and diet management for which people resort to expensive nutrition therapies. It is a known fact that energy balance plays a pivotal role in maintaining a healthy weight and lifestyle. If people become more aware about their food intake and its nutritional value, then the diseases mentioned above can be reduced. This project aims to develop a system that can record real time images of meal and analyse it for nutritional content, so that people can improve their dietary habits and lead a healthy life.

Existing systems assist users in achieving dietary goals such as weight gain/loss or maintaining a healthy diet. However, they require users to manually input the food details along with the portion sizes. This can be very tedious and time consuming, resulting in users to refraining from using these applications for long periods of time. Furthermore, naive users rely on self-reports of calorie intakes which often are misleading.

However, the task of real time food calorie estimation is not carried out. Crowd sourcing is also employed for the nutritional analysis of food items which makes the existing algorithm complex and inhibits widespread application in daily life.

In this study, recognition of food is followed by food volume estimation and calorie computation. The project aims to be a step towards creating awareness based on health and fitness concerns so that people can eat and live in a better way. It helps in determining the nutritional content of food automatically by making it feasible for a person to learn more about it. The inherent theme is to automatically classify food items from an image of a platter and then estimate the respective data such as area, volume, density, mass, total calories present. It consists of two modules. The first module uses a CNN to recognize the food item in an image. The second module estimates food nutrient count. This data is trained on a multi-layer neural network.

In visual object recognition tasks, Convolutional Neural Networks (CNN) have found great success and therefore CNNs are also employed for recognizing food items present in an image. In this work, we employ CNNs to acquire higher recognition accuracy rates of 86.6%. Another challenge is in the accurate computation of nutritional value of the food. Our goal is to minimize the user input and automate this task as much as possible. The project employs simple ratio transfer algorithm for estimating the dimensions of the food.

The output of the proposed system is implemented as a system application, where the mobile phone takes an image of food, sends it to the system file directory which is then given as an input to the model for classification and followed by estimation.

The classifier requires a large dataset containing multiple images against every category of food item for training purposes. This requires assistance from publicly available datasets such as Food-101 dataset.

1.2 RELATED WORK

Mobile devices are evolving rapidly. Every season, new generation of mobile devices are released that are more capable and computationally powerful than the previous generations. Along with the rapid growth of wireless internet technologies that promise high data rate and massive device connectivity, mobile multimedia services and applications can transform the health care sector. Numerous studies have been conducted to study the impact of mobile applications in healthcare processes.

A system that involves capturing an image of the food and processing it through predefined steps, which follow a pipeline architecture. These steps include food image segmentation and food portion recognition. Calorie measurement is done using nutritional fact tables. The system often fails to detect various food portions in mixed food, it also fails to segment them properly. The area measurement technique is based on a depth estimation technique. However, their system uses a dataset that is too simplistic, consisting of food items placed on white plates with smooth texture.

Depth camera such as Kinect is used to estimate the volume of food for calorie measurement. However, dependency of their system on Kinect makes the algorithm unsuitable for normal use. The method consists of three stages i.e. base plane localization, food segmentation and volume estimation. A 2D-3D model to image registration scheme is used for volume estimation. The algorithm does not perform accurately in cases of shadows, reflection, complex food, ingredients and motion blurring.

Similarly, Fang et al. use special fiducial markers placed in the scenes to estimate the food portion size. Im2calorie estimates food categories, ingredients and volume of individual dishes and calories. However, the calorie annotated dataset is not sufficient. The main approach for calorie estimation in the above mentioned methods is to start off by recognising the food category, followed by food portion size estimation and finally calorie estimation using standard nutritional fact tables.

There are other approaches that directly estimate the calories from food images. Some of them directly estimate food calories from photos of food by simultaneously learning about food categories, ingredients and cooking directions. Simultaneous learning of categories, ingredients and calories will boost performance as there exists a correlation between them.

Various approaches have also been proposed for food recognition only. There are two methods which include Speeded up Robust Features (SURF) and Spatial Pyramid Matching (SPM). SURF requires a dictionary of code words, and histograms are generated against those code words using a linear kernel classification scheme.

SPM accounts for spatial information by dividing and subdividing the given image into increasingly smaller sub regions and computing histograms in each. There exists a real time mobile food recognition system, which continuously acquires frames of the image from the camera device. The user draws boxes around the food items on the screen and food recognition is carried out within the boxes.

The graph cut based segmentation algorithm GrabCut is used for accurate food segmentation. Recognitions is performed using the linear kernel SVM (support vector machine). Camera position and viewing direction need to be maintained to obtain more reliable SVM classifications. Convolutional Neural Networks have also employed for the recognition task and as a result the recognition accuracy has improved significantly.

CHAPTER 2

LITERATURE SURVEY

2.1. Smartphone-Based Food Weight and Calorie Estimation Method for Effective Food Journaling

Elder Akpro Hippocrate, Hirohiko, Yutaka, and Keiichi, (2019), proposed that the World health statistics about overweight and obesity show that overweight, obesity and diet-related diseases still remain major health risks. According to the World Health Organization, most of the world's population live in countries where overweight and obesity kill more people than underweight. Recently, many studies have been driven by the motivation of elaborating the "ideal" solutions to prevent and/or monitor overweight, obesity and diet-related diseases, and to encourage healthy diet and lifestyle. In this work, the author presented their idea to automatically measure weight of food and calories, from images using ordinary chopsticks as a measurement reference. The analysis of the obtained results show that the use of near-by eating utensils combined with computer vision techniques is a great and exploitable approach to ubiquitously help in diet assessment and obesity treatment.

In this study, the author proposed a food weight measurement and calorie estimation system based on image processing, which uses a smartphone camera and chopsticks as measurement reference. The system, also, constructs a food journal to keep track of daily consumed meals. The author exploit image-processing techniques and the Exchangeable Image File Format (EXIF) metadata of the food image (camera focal length, sensor size) to measure the food container size, determine the food volume, get the food weight, estimate the calorie by using density and nutrient database information of that particular food. An important aspect of this process is the use of chopsticks, which suppress the obligation of carrying and using calibration objects utilized in others systems. The conducted experiments show tenable results from the system which achieved an average relative error rate of 6.65% for the weight measurement, and 6.70% relative error rate for the calorie estimation. The author believe that this method could be used as helping tool for use in treatment of obesity, overweight and diet-related disease.

2.2 Multisensor-based Food Volume Estimation on Smartphones

Junyi Gai, Weihao Tan, Liantao Ma, Yasha Wang and Wen Tang (2019) – MUSEFood, proposed that **t**he researches have shown diet recording can help people increase awareness of food intake and improve nutrition management, and thereby maintain a healthier life. Recently, researchers have been working on smartphone-based diet recording methods and applications that help users accomplish two tasks: record what they eat and how much they eat. Although the former task has made great progress through adopting image recognition technology, it is still a challenge to estimate the volume of foods accurately and conveniently. The authors proposed a novel method, named MUSEFood, for food volume estimation.

MUSEFood uses the camera to capture photos of the food, but unlike existing volume measurement methods, MUSEFood requires neither training images with volume information nor placing a reference object of known size while taking photos. In addition, considering the impact of different containers on the contour shape of foods, MUSEFood uses a multi-task learning framework to improve the accuracy of food segmentation, and uses a differential model applicable for various containers to further reduce the negative impact of container differences on volume estimation accuracy.

Furthermore, MUSEFood uses the microphone and the speaker to accurately measure the vertical distance from the camera to the food in a noisy environment, thus scaling the size of food in the image to its actual size. The experiments on real foods indicate that MUSEFood outperforms state-of-the-art approaches, and highly improves the speed of food volume estimation.

The authors have introduced MUSEFood to calculate food volume using data collected from multiple sensors on smartphones. MUSEFood uses FCN and utilizes shape information of food containers through multi-task learning structures, resulting in more accurate and fast food image segmentation. The authors use the Maximum Length Sequence (MLS) ranging instead of using reference objects, which improves the convenience of use and achieves higher food volume estimation accuracy.

MUSEFood shows sufficient robustness and versatility in our experiments. The shape of the different food containers does not introduce errors in the results of the food volume estimation. MUSEFood can handle food without regular shapes, which makes the models applicable to more types of foods. Though some smartphones have Time of Flight (ToF) cameras, which can directly obtain the target distance, smartphones with ToF cameras are not very popular. Requiring user to buy the special equipment will increase the cost inevitably. Furthermore, the develop interfaces of these ToF cameras are hardly public to developers.

2.3 An Automatic Calorie Estimation System of Food Images on a Smartphone

Koichi Okamoto, Keiji Yanai (2016), proposed that in recent years, due to a rise in healthy thinking on eating, many people take care of their eating habits, and some people record daily diet regularly. To assist them, many mobile applications for recording everyday meals have been released so far. Some of them employ food image recognition which can estimate not only food names but also food calorie. However, most of such applications have some problems especially on their usability. A novel single-image-based food calorie estimation system which runs on a smartphone as a standalone application without external recognition servers. It is carries out food region segmentation, food region categorization, and calorie estimation automatically.

In this study, the authors proposed an image-based calorie estimation system which runs on a consumer smartphone without external recognition servers. The system estimate food calories automatically by simply taking a meal photo from the top with a pre-registered reference object.

In the step to detect food regions, the author used edge-based dish localization and k-means-based dish bounding box estimation, and finally the author applied GrabCut for accurate food region estimation. To recognize each of the detected food regions, the author adopted the CNN-based food recognition engine which recognizes food photos in around 0.2 seconds. To estimate calorie, simple linear estimation was not used but quadratic curve estimation from the 2D size of foods to their calories was used.

The quadratic curve of each food category is trained based on the training data annotated with real food calories independently. As results, in the experiments, the author achieved the absolute average error, 52.231kcal, and the relative average error, 21.3%, regarding food calorie estimation with 60 test images annotated with real food calorie values. In addition, the authors obtained positive evaluation by the subjects regarding the usability of their system compared to the baseline system.

In this work, relatively simple segmentation methods were used for taking mobile implementation into account. Therefore, it is sometime difficult to treat a meal photo with non-uniform background. For future work, the author wants to introduce more sophisticated segmentation methods, hopefully state-of-the-art CNN-based methods. In addition, planar calibration using a rectangular card as a reference object was used, which relaxes the assumption of taking a meal photo from the overhead.

2.4 Semi Automated System for Predicting Calories in Photographs of Meals

Patrick McAllister, Huiru Zheng, Raymond Bond, Anne Moorhead (2015), proposed that obesity is increasing globally and brings with it many chronic conditions. There has been increasing research in the use of ICT interventions to combat obesity using food logging and image calorie analysis. These interventions allow users to document their calorie intake to help promote healthy living. However using food logs may lead to inaccurate readings as the user may incorrectly calculate portion size when recording nutritional information. This study discusses the use of image nutritional analysis techniques to ascertain a more accurate calorie reading from photographs of food items.

The methods employed involve determining a ground truth data set by correlating weight of a food item with its area in cm^2. This dataset could then be plotted on a regression model and used to determine calorie content of future portions. This system uses a semi-automated approach to allow users to manually draw around the food portion using a polygonal tool. Results show that the application achieved a reasonable accuracy in predicting the calorie content of food item portions with a 11.82% error.

The user initially pours ten grams of sweet corn onto a plate. The $1cm^2$ was placed next to the plate of food and an image was taken using a smartphone. The area is calculated by using a $1cm^2$ reference point and the area is used as input to determine calorie content.

The user was able to draw around a portion of food by using a polygonal tool. The application then calculates the area of the food portion in cm^2. This process is repeated five times for each weight. After the data collection was complete, a regression model was trained using the average areas along with calories per portions.

This application was tested using different portions of the same food item. The portion area size was unknown prior to the testing. The ground truth calories for each weighted portion were determined using food item packaging. The same process described earlier was then completed for each image to determine calorie content. The reason why weight was taken into consideration was that the calorie content could be determined using the packaging contents stating calories per 100g. This could be used to compare the experimental results and ground truth data.

Results show that the application is able to ascertain the calorie content with reasonable efficiency with an average percentage error of 11.82%. The error percentage for each portion accounted for only a small number of calories.

2.5 A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques

Raza Yunus, Omar Arif, Hamad Afzal (2019), proposed that there has been a rapid increase in dietary ailments during the last few decades, caused by unhealthy food routine. Mobile-based dietary assessment systems that can record real-time images of the meal and analyze it for nutritional content can be very handy and improve the dietary habits and, therefore, result in a healthy life. It proposed a novel system to automatically estimate food attributes such as ingredients and nutritional value by classifying the input image of food.

The method employs different deep learning models for accurate food identification. This system is implemented as a mobile app that has its application in the healthcare sector. It presents a system that exploits the extensive use of mobile devices to provide health information about the food we eat. The mobile-based app takes the image of the meal and presents approximate ingredients and nutritional values in food.

A fine-tuned Inception model is employed to recognize food items and a method to estimate attributes of the recognized food item is proposed. The results are improved via data augmentation, evaluation, regularization and other similar techniques. The method proposed by the author for estimating attributes also achieved encouraging results.

2.6 ImageBased Estimation of Real Food Size for Accurate Food Calorie Estimation

Takumi Ege, Yoshikazu Ando, Ryosuke Tanno, Wataru Shimoda and Keiji Yanai (2019), proposed that the Image-based estimation of real size of foods is carried out for accurate food calorie estimation using the following methods:

1. “CalorieCam” which is a system to estimate real food size based on a reference object.
2. Region segmentation based food calorie estimation.
3. “AR DeepCalorieCam V2” which is based on visual inertial odometry built in the iOS ARKit library.
4. “DepthCalorieCam” which employs stereo cameras on iPhone X/XS.
5. “RiceCalorieCam” which uses rice grains as reference objects.

In the last two methods 10% or less estimation error, which was enough for robust food calorie estimation has been achieved. In food image recognition, CNN-based methods have achieved great improvement in recent years and some smartphone applications employ them.

However, in most of the calorie estimation, the estimated calories are just associated with the estimated food categories and these applications often require users to enter information such as size or volume, there are problems that it is a troublesome and subjective evaluation.

The food calorie estimation not only helps people's health a lots, but also is promising as a new problem of image recognition studies. DepthCalorieCam is the most promising approach. However, large-scale calorie annotated 3D food volume data is needed to extend the system into large-scale categories, which is very costly and time-consuming. In addition, the rice grain based method is also promising for meals containing white steamed rice. Especially it is appropriate for Japanese foods.

2.7 Image-based Calorie Content Estimation for Dietary Assessment

Tatsuya Miyazaki, Gamhewage C. de Silva, Kiyoharu Aizawa (2011), proposed that an image-analysis based approach is used for calorie content estimation for dietary assessment. Daily food images are captured and stored by multiple users in a public Web service called FoodLog. The images are taken without any control or markers. A dictionary dataset of 6512 images contained in FoodLog the calorie content of which have been estimated by experts in nutrition is built. An image is compared to the ground truth data from the point of views of multiple image features such as color histograms, color correlograms and SURF features, and the ground truth images are ranked by similarities. Finally, calorie content of the input food image is computed by linear estimation using the top n ranked calories in multiple features.

A web-based dietary management system is used that uses image analysis for extracting nutrition information and estimating calorie content of meals. Image-based input provides an easy way for users to keep a record of their meals. Calendar, graph and map-based visualizations make it easier for them to look at the data. A novel approach to calorie content estimation only based on food images is used. Many pictures in the current dataset show more than one dish. Therefore, image segmentation to identify different dishes and make individual estimates of calorie content can improve the accuracy of calorie content estimation.

2.8 A Survey on Food Computing

Weiqing Min, Shuqiang Jiang, and Linhu Liu (2019), proposed that food-related study may support multifarious applications and services, such as guiding human behavior, improving human health, and understanding the culinary culture. With the rapid development of social networks, mobile networks, and Internet of Things (IoT), people commonly upload, share, and record food images, recipes, cooking videos, and food diaries, leading to large-scale food data. Large-scale food data offers rich knowledge about food and can help tackle many central issues of human society. Therefore, it is time to group several disparate issues related to food computing.

Food computing acquires and analyzes heterogenous food data from different sources for perception, recognition, retrieval, recommendation, and monitoring of food. In food computing, computational approaches are applied to address food-related issues in medicine, biology, gastronomy, and agronomy. Both large-scale food data and recent breakthroughs in computer science are transforming the way we analyze food data. Therefore, a series of works has been conducted in the food area, targeting different food-oriented tasks and applications. However, there are very few systematic reviews that shape this area well and provide a comprehensive and in-depth summary of current efforts or detail open problems in this area. Food computing is formalized and presents such a comprehensive overview of various emerging concepts, methods, and tasks.

This is the first comprehensive survey that targets the study of computing technology for the food area and also offers a collection of research studies and technologies to benefit researchers and practitioners working in different food-related fields.

Food computing is a vibrant interdisciplinary field that aims to utilize computational approaches for acquiring and analyzing heterogeneous food data from disparate sources. With the increasing availability of large-scale food data, more food-oriented computational methods from different fields, such as computer vision and machine learning, will be widely used or quickly developed to enable the prosperity of food computing. Because of its interdisciplinary nature, it can be applied into many applications and services in various fields, from health, culture, agriculture, and medicine to biology. Moving forward, the food computing framework helps researchers understand current research and identify unresolved issues for future research.

CHAPTER 3

PROPOSED WORK

The proposed system presents a novel solution to classify and estimate calorie count in food items. The system is designed to be a lightweight yet a powerful module. The system consists of two major modules:

- Classification Module: Classifying food from images.
- Estimation Module: Estimating the dimensions of the food from input image using a reference object followed by calorie computation.

The publicly available dataset of food images, i.e. Food-101 has been used for training the model. It contains 101 classes of food items with 1000 images in each class. Food-101 is designed specifically for multi-class classification.

There are other datasets as well that have been used for food recognition previously; one such dataset is Food-5k, which is a variant of Food-101. Food-5k contains 5000 images, out of which 2500 are of food and 2500 of non-food. However, this dataset can be used only for binary classification to discriminate food items from non-food items and therefore, is not suitable for our task.

The Food 101 dataset contains images of food, organized by type of food. It does not include food items or classes from the sub-continental cuisine which makes a large portion of the food that people intake in the subcontinental region. Some sub-continental dishes exhibit low inter-class variation and are very similar to each other, so collecting high quality data for accurate classification of different categories is a big challenge.

Owing to the great success of CNNs, top performing pre-trained models are used to train our dataset using transfer learning. Based on their performance in other domains, pre-trained CNN model, Inception-v3 has been used. When a relatively small dataset is used, a super-effective technique is to use **Transfer Learning** where a pre-trained model is used. This model has been trained on an extremely large dataset and we would be able to transfer weights which were learned through hundreds of hours of training on multiple high-powered GPUs.

Many such models are open-sourced such as VGG-19 and Inception-v3. They are trained on millions of images with extremely high computing power which can be very expensive to achieve from scratch. Transfer Learning has become immensely popular because it considerably reduces training time, and requires a lot less data to train on to increase performance.

Transfer Learning has great applications in Computer Vision. Not everyone can afford weeks of training in addition to access to millions of

images to build a high performing CNN model for their custom use case, and this is where Transfer Learning comes in. The power and the flexibility to

retrain some of the lower layers (as per target classes) using a reasonable amount of resources.

Initially the VGG-16 model was used. It fetched satisfactory results closer to Inception v3 but took more time for each epoch when compared to Inception v3 which consumed ~30 mins for each epoch. CNN models based on Inception-v3 are selected as they perform better than the other models tested. In the model, the last fully connected layer is removed and appended with dropout, ReLU activations and softmax layers. This model is loaded to train on Food-101 dataset using 75,750 training images and evaluated on the 25,250 validation images, disregarding data augmentation.

After initial training, Inception-v3 continued to be the base model as the appropriate model based on the performance on the dataset. To improve the validation accuracy of the model on the dataset, various techniques have been employed. First, intensive data augmentation is performed so that the model is robust to affine variations as much as possible and the images are efficiently trained. In every epoch various transformations with random parameters specified by parametric range, are applied on each image of the dataset to produce copies that are transformed from the original image. These include translations, rotations, shearing, zooming and flipping.

Images are also re-scaled considering the large values of RGB coefficients for the model to process. Re-scaling involves multiplication of the RGB value by 1/255 factor, so target values are normalized and lie

between 0 and 1. This makes the image robust to variations in illumination and makes processing data faster.

Other steps for improving the accuracy include batch normalization and regularization to tackle over-fitting and multi-crop evaluation. In multi-crop evaluation, at the time of testing an image, multiple crops of an image are taken from different regions of the image and each crop is tested individually. The most frequently occurring class in the list of resulting predicted labels is considered. In this work, four crops are obtained by equally dividing the image into four squares and a centre crop is also taken. The image is then flipped and the same process is applied.

So in total, we get 10 crops for each image. Moreover, learning rates and decay values are also optimized to get more accurate results. Early stopping saves time as it stops training if the model does not show improvement in the validation accuracy for a set amount of continuous epochs. The last two convolution blocks of the Inception model are also made trainable, along with the final fully connected layer, so that more high-level features specific to our dataset are learned.

A smartphone-based application is used to take an image from the gallery of the phone and send it to the local system directory which is then used by the model in order to perform calorie estimation.

With the predicted label, an image with reference object is required as input to estimate area followed by calorie with few intermediate

computations. The Estimation layer contains a set of simple mathematical operations after the initial estimation of area to derive the total calorie present in the food.

3.1 PROPOSED ARCHITECTURE

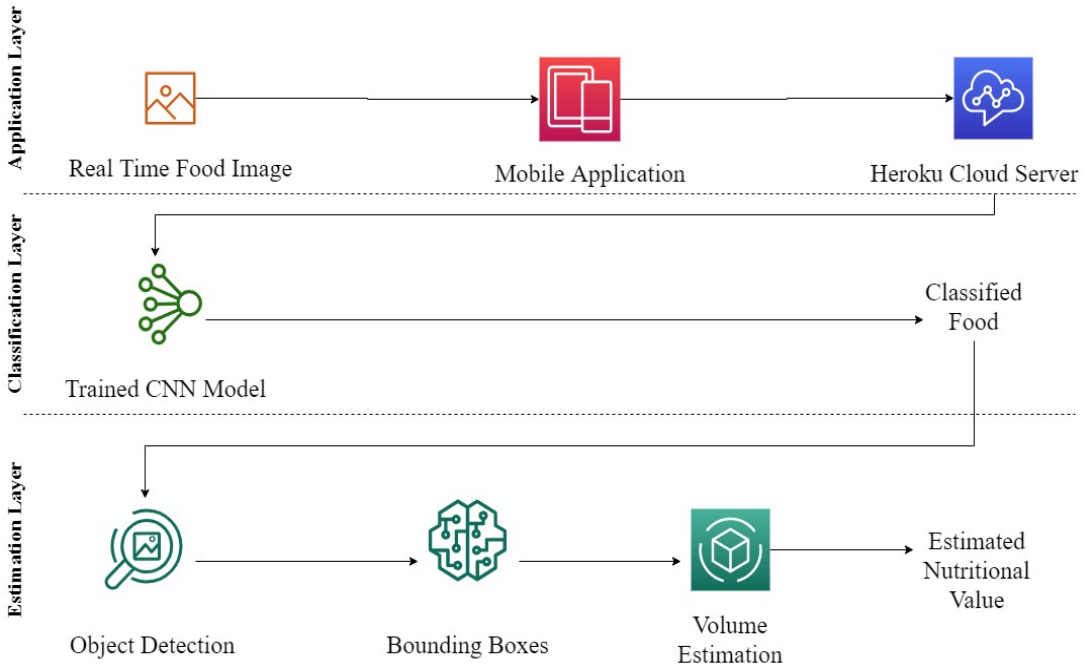

Figure 3.1 System Architecture

The system architecture as shown in Fig 3.1 consists of 3 layers:

1. **Application Layer:** A Real-time food image is selected from mobile gallery and is sent to the server using a mobile application.
2. **Classification Layer:** Upon receiving the image from the mobile application, it is given as the input to the trained CNN model in order to classify the food.
3. **Estimation Layer:** Upon classification, the food item is detected and a bounding box around the food item is drawn. The ratio transfer algorithm computes the dimensions of the food item which leads to calorie estimation.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

PROCESSOR	: Intel Core i3, 2 cores and above
HARD DISK DRIVE	: 10 GB
RAM	: 16 GB
GRAPHICS	: 4GB Integrated Graphics and above
SYSTEM TYPE	: x64 – based PC

4.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM	: Windows 7 and above
LANGUAGES	: Python 3.7, Android 6.0+
LIBRARIES	: Keras, Tensorflow, OpenCV, Pillow, NumPy, Flask, SciPy, Tkinter, Imutils, CSV
IDE & OTHER TOOLS	: Anaconda, Visual Studio Code, Android Studio, Jupyter Lab

4.3 DESCRIPTION OF TOOLS AND LIBRARIES

4.3.1 Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub and community support forums include the GitHub issues page, and a Slack channel.

In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization and pooling.

4.3.2 TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. It is used for easy model building, robust ML production and a powerful experimental tool for research.

4.3.3 OpenCV

Open Source Computer Vision (OpenCV) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and

establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million.

4.3.4 Pillow

Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats. Write support is intentionally restricted to the most commonly used interchange and presentation formats.

4.3.5 NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

4.3.6 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open

authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

4.3.7 SciPy

SciPy is a free and open-source Python library used for scientific computing and technical computing. It builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack.

SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India). Enthought originated the SciPy conference in the United States and continues to sponsor many of the international conferences as well as host the SciPy website.

4.3.8 Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's *de facto* standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python. It is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

4.3.9 Comma-Separated Values (CSV)

A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields.

The CSV file format is not fully standardized. The basic idea of separating fields with a comma is clear, but that idea gets complicated when the field data may also contain commas or even embedded line breaks. CSV implementations may not handle such field data, or they may use quotation marks to surround the field. Quotation does not solve everything: some fields may need embedded quotation marks, so a CSV implementation may include escape characters or escape sequences.

4.3.10 Anaconda

Anaconda is a free and open-source distribution of the python and R Programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analysis, etc.). that aims to simplify package management and deployment. It is developed and maintained by Anaconda Inc. The distribution includes data-science packages suitable for Windows, Linux and MacOS.

Package versions are managed by the package management system *conda*. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called **Miniconda**, which includes only conda, Python, the packages they depend on, and a small number of other packages.

4.3.11 Visual Studio Code

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open-source, released under the permissive MIT License. The compiled binaries are freeware for any use.

4.3.12 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

CHAPTER 5

IMPLEMENTATION MODULES

5.1 FOOD CLASSIFICATION

In this project, Inception-v3 model has been used for Transfer Learning. It is a convolutional neural network that is 48 layers deep. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299. The weights of model trained using ImageNet dataset is loaded before training on Food101.

5.1.1 Data Pre-Processing

The open-source dataset is downloaded from a repository in ETH Zurich website. The entire dataset has images summing up to 5 GB of disk size in compressed format. Since the dataset is noise-free it reduces an intermediate cleaning process. The dataset also provides a train/test split of 75% and 25% respectively. On unzipping the data and copying it into separate folders for train/test, the data is loaded into memory. After loading everything, about 80 GB of memory is allocated. After data pre-processing, the training data contains 75,750 images from 101 classes with each class labeled in a

separate folder. The validation data contains 25,250 images in a separate folder.

5.1.2 Image Augmentation

The Image augmentation configurations used for training comprises of rotation, horizontal and vertical flips, width, height shifts, zoom ranges, channel shifts, rotation etc. After using all these augmentations, the ImageDataGenerator generates the training data. Each image is cropped to have a uniform shape of 299 X 299 which is the input shape of Inception v3 model. Correspondingly, validation data shown in Figure 5.1 is also generated to have a dimension matching Inception v3's input shape.

Figure 5.1 Test Images

5.1.3 Training

The project retrains the Google's Inception - v3 model as shown in Figure 5.2, which is pretrained on ImageNet dataset for improved accuracy and rapid training.

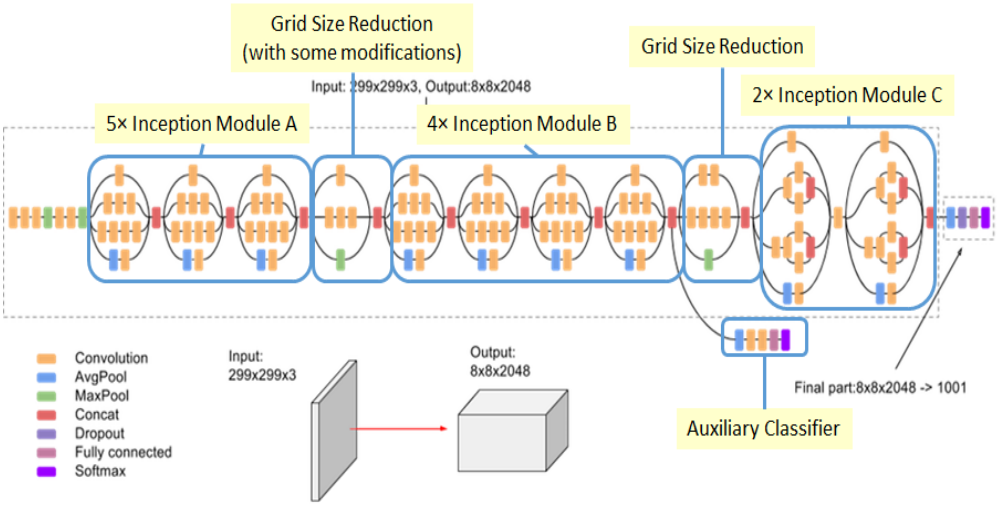

Figure 5.2 Inception v3 Architecture

The base model which has a input shape of 299 X 299 with weights of ImageNet is defined. The developed model uses the input of base model and has the output dense layer with 101 classes. The model is compiled to have categorical cross entropy as loss function. The model is then trained using the images in training data folder and validation data is assigned with the test data folder.

```
Epoch 29/32
Epoch 00028: val_loss improved from 0.69190 to 0.68908, saving model to model4.28-0.69.hdf5
1330s - loss: 0.5983 - acc: 0.8580 - val_loss: 0.6891 - val_acc: 0.8165
Epoch 30/32
Epoch 00029: val_loss improved from 0.68908 to 0.68740, saving model to model4.29-0.69.hdf5
1330s - loss: 0.5817 - acc: 0.8612 - val_loss: 0.6874 - val_acc: 0.8149
Epoch 31/32
Epoch 00030: val_loss did not improve
1328s - loss: 0.5729 - acc: 0.8642 - val_loss: 0.6912 - val_acc: 0.8143
Epoch 32/32
Epoch 00031: val_loss did not improve
1329s - loss: 0.5638 - acc: 0.8663 - val_loss: 0.6895 - val_acc: 0.8159
CPU times: user 8h 49min 20s, sys: 1h 55min 54s, total: 10h 45min 14s
Wall time: 11h 51min 18s
```

Figure 5.3 Model Training Summary

The total number of epochs is 32, In which there is no improvement in validation loss beyond 30th epoch. The training accuracy improved to 86.6% at the end of 32nd epoch. The validation accuracy is 81.59% which is the Top-1 accuracy on the test set as shown in Figure 5.3. The histogram depicting the accuracy by class is shown in Figure 5.4.

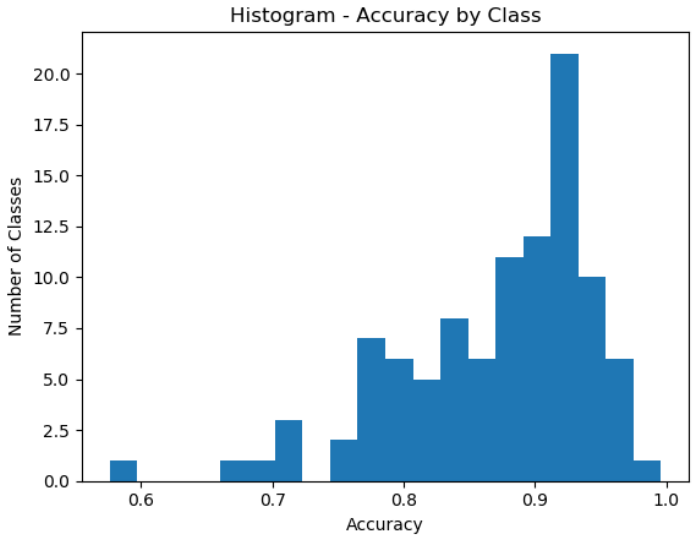

Figure 5.4 Accuracy by Class Histogram

5.1.4 Results

The results from the CNN model as shown in Figure 5.5 are obtained while testing it. The results from the model were impressive and satisfactory. The model was able to predict the classes of all the ten given test images downloaded from the internet correctly.

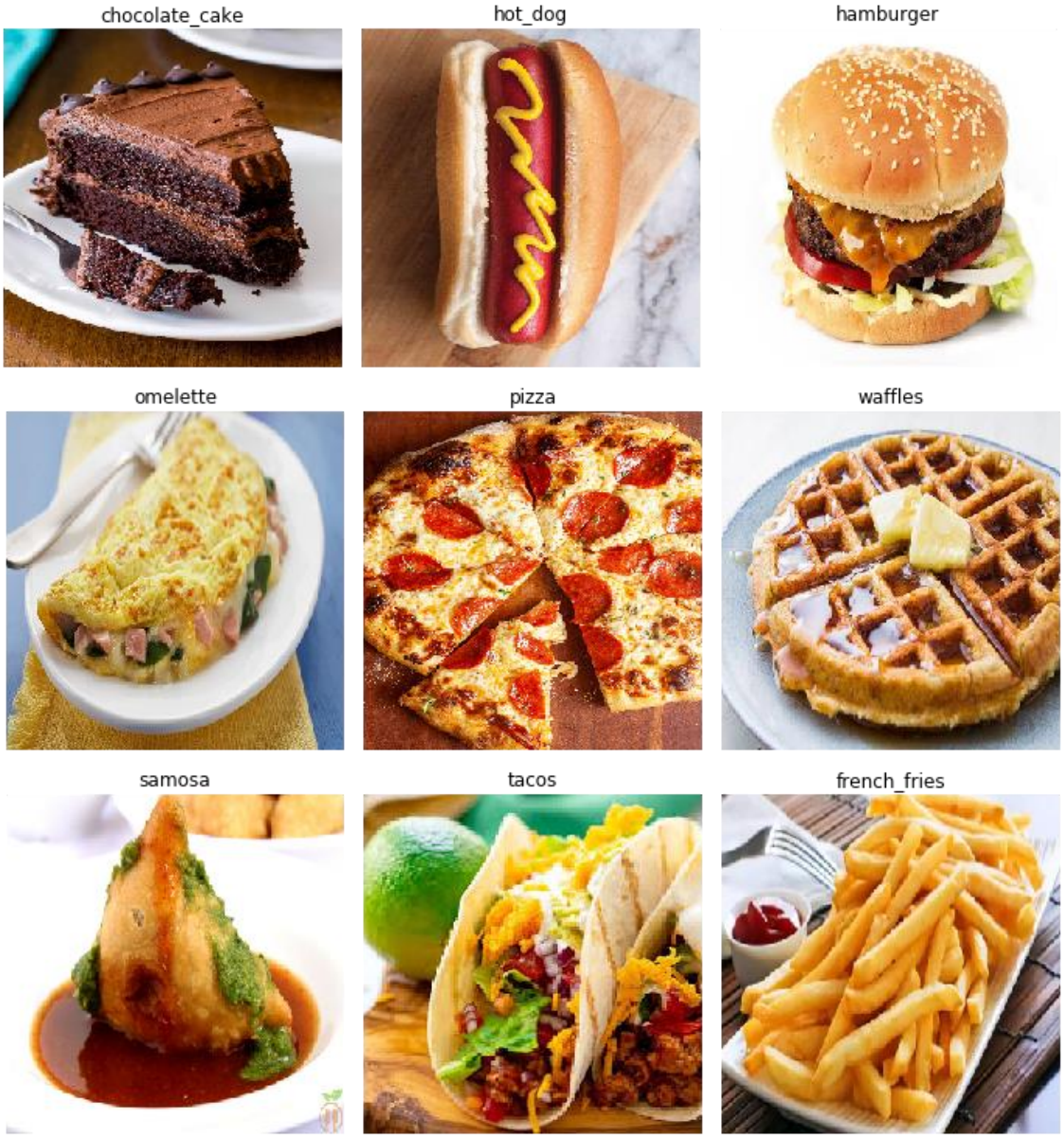

Figure 5.5 Results from CNN for Food Items from 9 Different Classes

5.2 CALORIE ESTIMATION

A novel ratio transfer algorithm is developed to estimate the size of a given food image. The algorithm takes a calibration object and the food image as parameters. Once the size is estimated we multiply by its height to find the volume. The height of the food item, calorie content per 100 grams of the food item and density for corresponding labels that are obtained by web scraping these values are used for calculation of volume and total calories present.

5.2.1 Ratio Transfer Algorithm for Dimension Estimation

The algorithm takes an image containing the food with a reference object used for calibration as shown in Figure 5.6 in the left as input. The dimensions of the calibration object are predefined. This reference object used for calibration can be a coin or a tiny cutout of a paper. While testing an Indian 2 Rupee coin was used as the reference object.

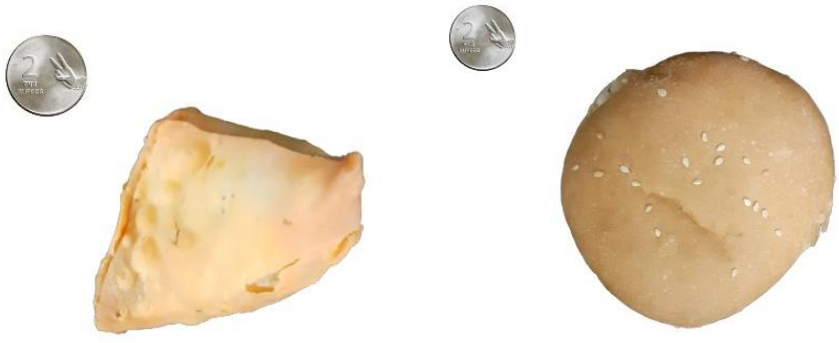

Figure 5.6 Real-time images of a samosa and a burger with a 2 Rupee Coin as Reference Object

The algorithm takes the dimensions of the calibration (left most object) as reference for example 3 cm X 3 cm and calculates the number of pixels present as length and width. Then the algorithm finds the biggest contour present in the image and draws a bounding box around it. The pixel length and width of the bounding box shown in Figure 5.7 is measured and the relative value in cm is computed as the dimensions of food item. The formula used for ratio transfer is

$$\text{Length}_{\text{Food Item}} \text{ (in cm)} = \text{Pixel Length}_{\text{Food Item}} \text{ X Length}_{\text{Reference Object}} \text{ (in cm) / Pixel Length}_{\text{Reference Object}}$$

$$\text{Width}_{\text{Food Item}} \text{ (in cm)} = \text{Pixel Width}_{\text{Food Item}} \text{ X Width}_{\text{Reference Object}} \text{ (in cm) / Pixel Width}_{\text{Reference Object}}$$

Then the area of the Food Item in the Image is computed using the formula,

$$\text{Area}_{\text{Food Item}} \text{ (in cm}^2\text{)} = \text{Length}_{\text{Food Item}} \text{ (in cm) X Width}_{\text{Food Item}} \text{ (in cm)}$$

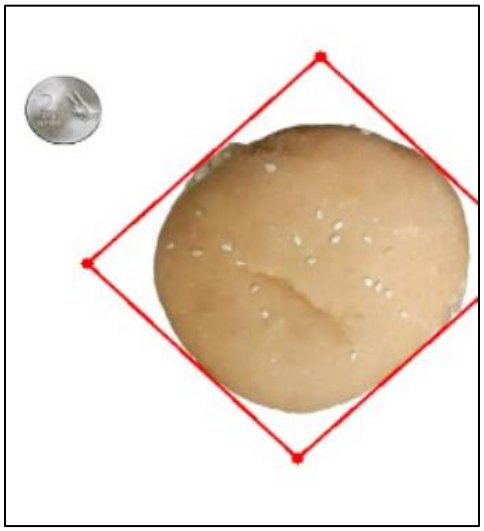

Figure 5.7 Bounding Box of the Biggest Contour in an Image

Once the area is calculated the Height (in cm) of the food is retrieved from the web scrapped dataset in CSV format.

The volume is calculated as

$$\text{Volume}_{\text{Food Item}} \text{ (in cm}^3\text{)} = \text{Area}_{\text{Food Item}} \text{ (in cm}^2\text{) X Height}_{\text{Food Item}} \text{ (in cm)}$$

After estimating the volume, the mass of the food is estimated using the formula:

$$\text{Mass (in g)} = \text{Density (in g/cm}^3\text{)} * \text{Volume (in cm}^3\text{)}$$

The calorie content of the food item is estimated by the following formula.

Total Calories = C (in kcal/100g) X Mass (in g) / 100

Where C (in kcal/100g) represents kilo calories per 100 grams.

Algorithm:

1. Start

2. Read the image from the file

3. Convert the image as a grayscale and store it in gray

4. Apply image smoothing and reduce the noise in the image and store it in gray

5. Find the edges of the object in the image and store it in edges

6. Apply image dilation and store it in edges

7. Apply image erosion and store it in edges

8. Contours of the image is obtained and stored in cnts

9. Iterate with contours stored in a cnts as the iterator

10. If (contour area of the current contour is less than 100)

    go to the next iteration

11. Else Repeat steps

    11.1 Minimum area rectangle is constructed in the image contour

11.2 The coordinates of the rectangle are stored in the box variable

11.3 Construct a circle with the all coordinates of the box

11.4 Construct the mid points of the box edges

11.5 Euclidean distance is calculated between the obtained midpoints

11.6 Pixels with respect to the euclidean distance is calculated.

12. From the obtained boxes, the box with the maximum area is found

13. Contours and circle are constructed for the maximum area rectangle

14. Label, volume, density, calories and mass are calculated and are plotted on the image with maximum area.

15. Stop

The Estimation results using ratio transfer algorithm for a hamburger is shown in Figure 5.8

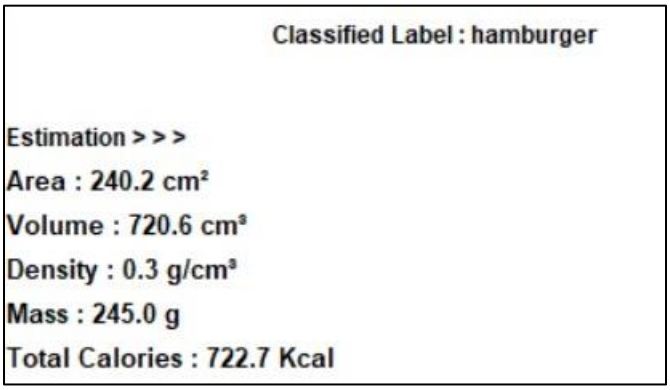

Figure 5.8 Calorie Estimation of a Hamburger

5.3 MOBILE APPLICATION

The user interface consists of following entities:

- Image Description Textbox: It consists of the path of the image on the smartphone.

- Select Image Button: An image from the gallery can be selected upon clicking and adds path to textbox.

- IPv4 Address: This is used to take the IP Address of the working system as input.

- Port Number: The port that is used for carrying out and processing the operation which is 5000 in this case.

- Connect to Server button: Upon clicking the button it sends the image to a local directory across the server to a computer and stores it.

- FLASK: Python Library where the server is up and running in order for the process to complete and send the image to the local system through a localhost server.

- The server upon receiving the image sends a response to the user about the image received and the number of images that is received.

CHAPTER 6

SNAPSHOT OF MODULES

Snapshots in order of application flow from image acquisition to calorie estimation.

6.1 MOBILE APPLICATION

The Home screen of the Mobile App as shown in Figure 6.1 accepts the IP address and Port Number for sending the images.

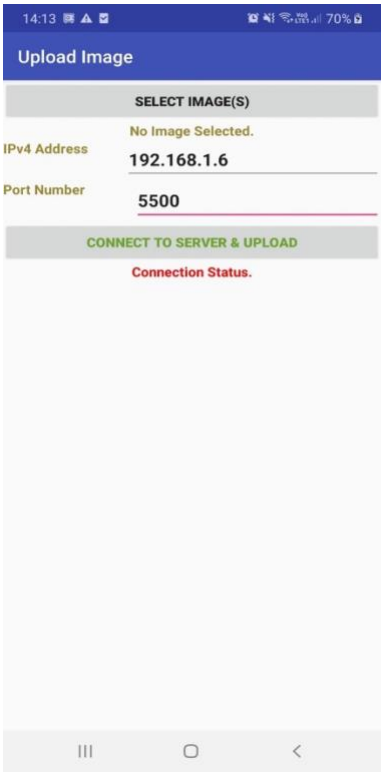

Figure 6.1 Home Page of Application to Select Image

After selecting the “Select Images” option the application takes the user into file directory as shown in Figure 6.2 for choosing the images to be sent.

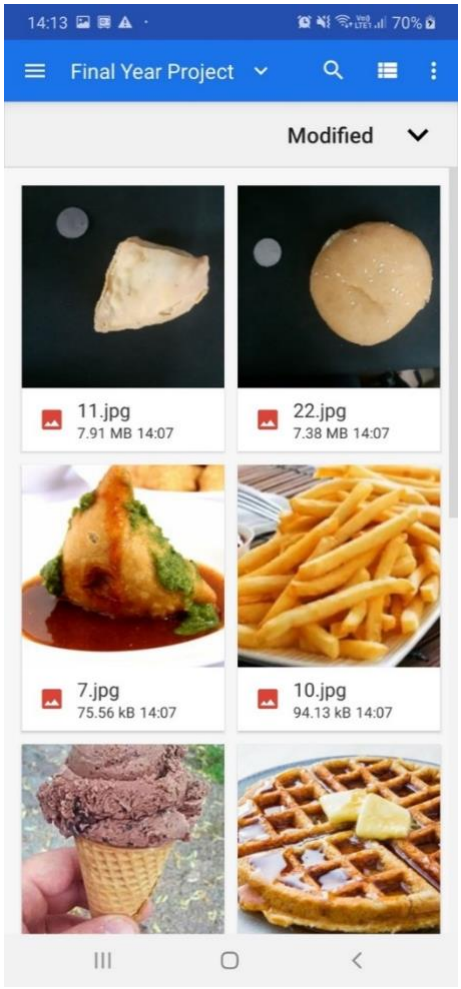

Figure 6.2 Choosing an image from Gallery

After selecting the images, the Application sends the selected ones to the specified IP Address and displays the status below in the Home Screen as shown in Figure 6.3.

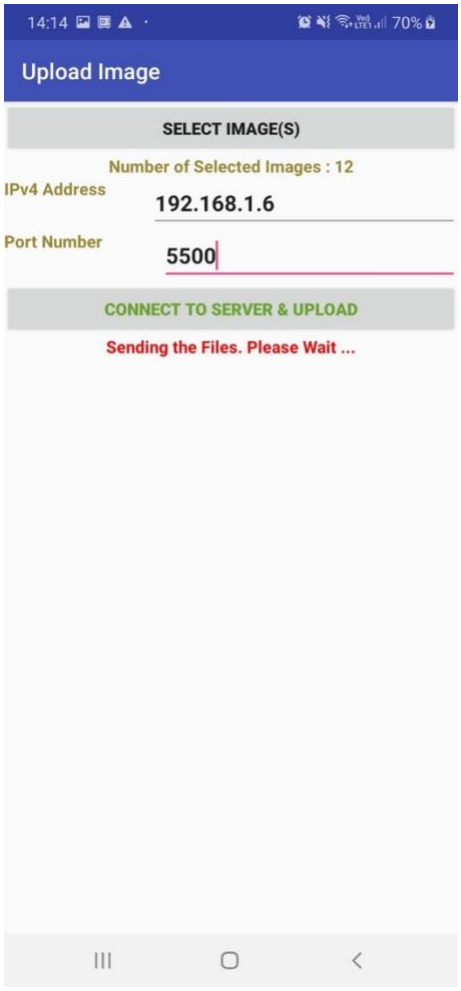

Figure 6.3 Sending the image from Gallery to System Directory

After sending the images the application displays the response received from the server and changes its status to “Successful” as shown in Figure 6.4.

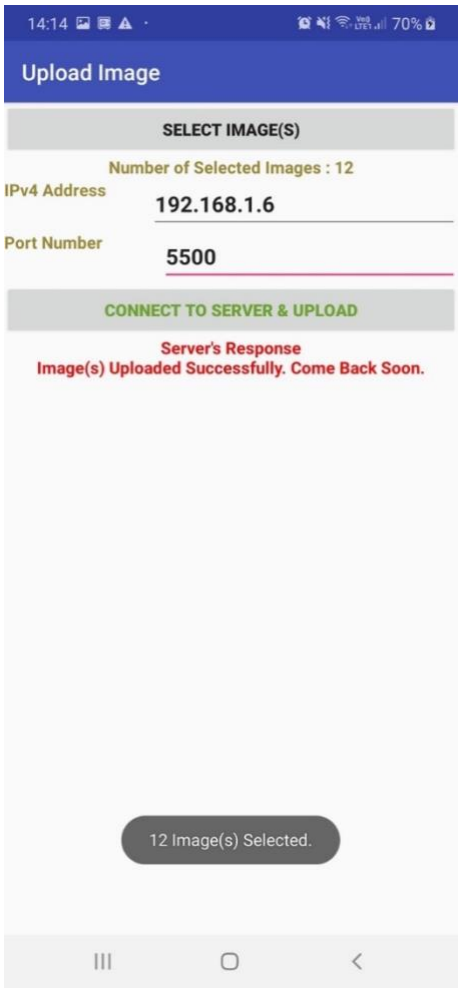

Figure 6.4 Response from Server upon Uploading the image

Figure 6.5 and Figure 6.6 shows the desktop application receiving the images sent from the mobile application.

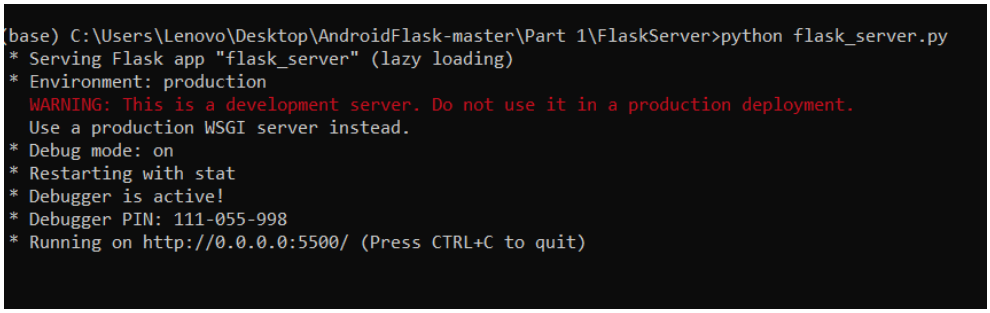

Figure 6.5 Initializing the Desktop Application Server to Receive Images

Figure 6.6 Response from Server upon Receiving Image

Figure 6.7 shows the images received from mobile application being saved in the system file directory.

Figure 6.7 Images received from Application Saved in System Directory

6.2 FOOD CLASSIFICATION

Home screen of the desktop GUI Application shown in Figure 6.8 after loading the trained model in main memory.

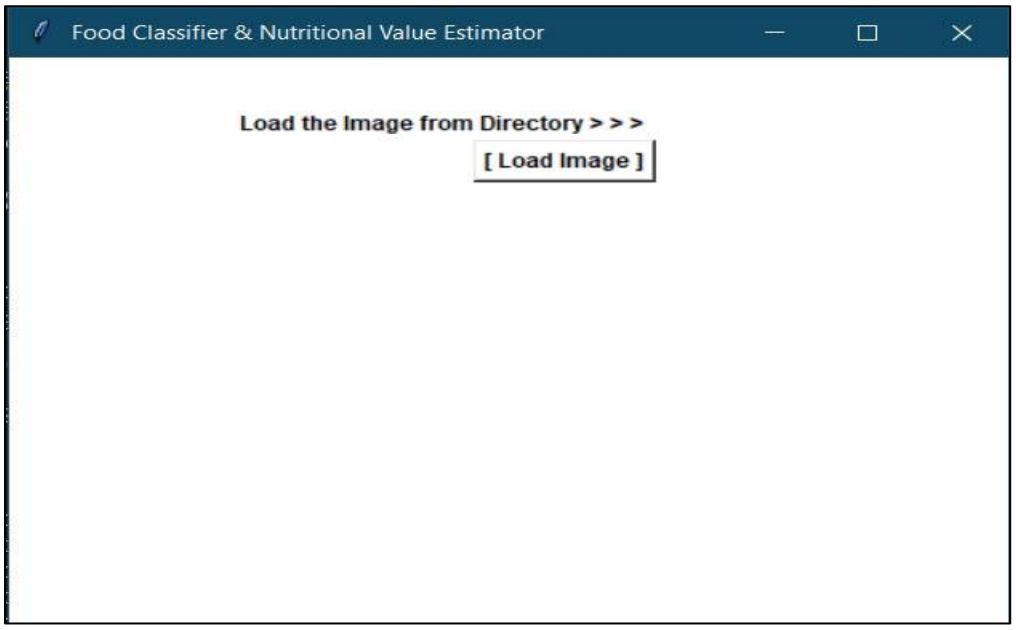

Figure 6.8 Loading an Image from the Directory Sent from the Mobile Application

Upon selecting “Load Image” the application open system dialog for choosing image from directory as shown in Figure 6.9 and displays the selected images as shown in Figure 6.10.

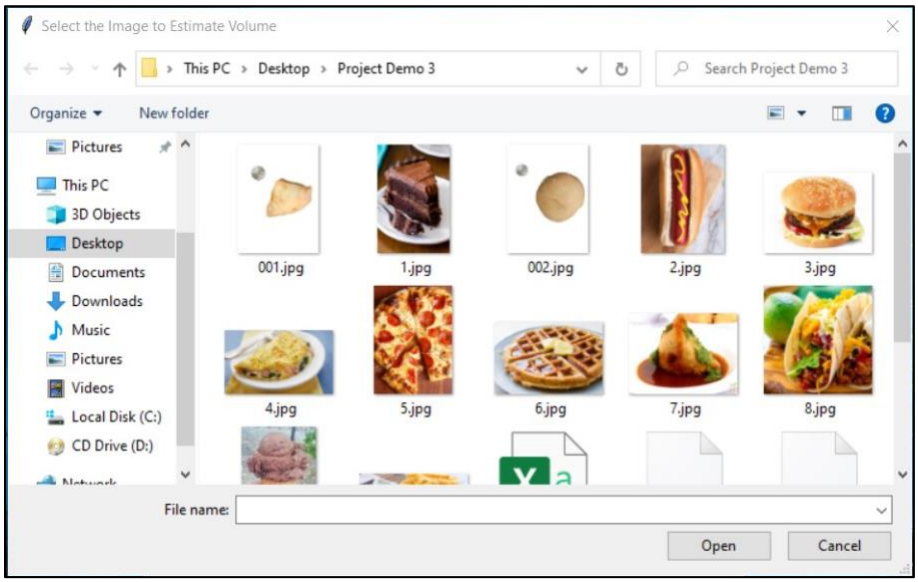

Figure 6.9 Choosing the Image from the Directory

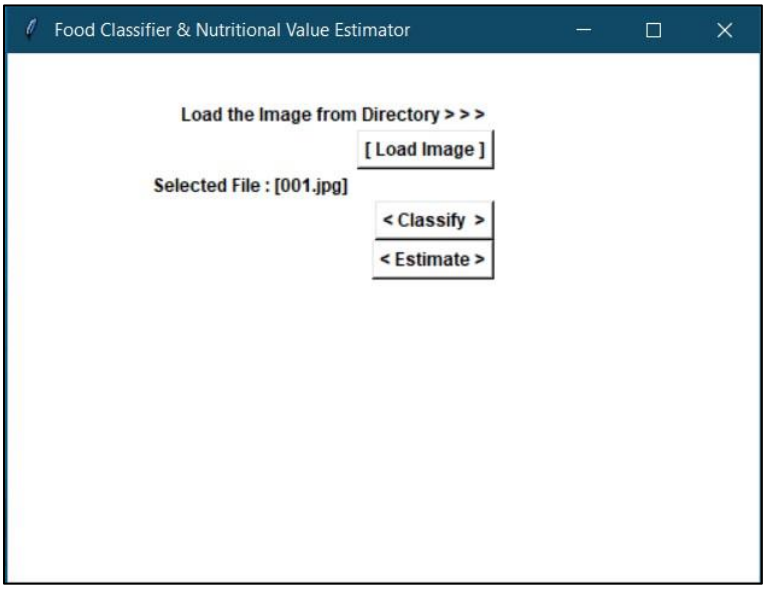

Figure 6.10 Choosing Classification or Estimation of Chosen Image

On clicking “Classify” the image is given as the input to the trained model and the predicted label is displayed in the application as shown in Figure 6.11.

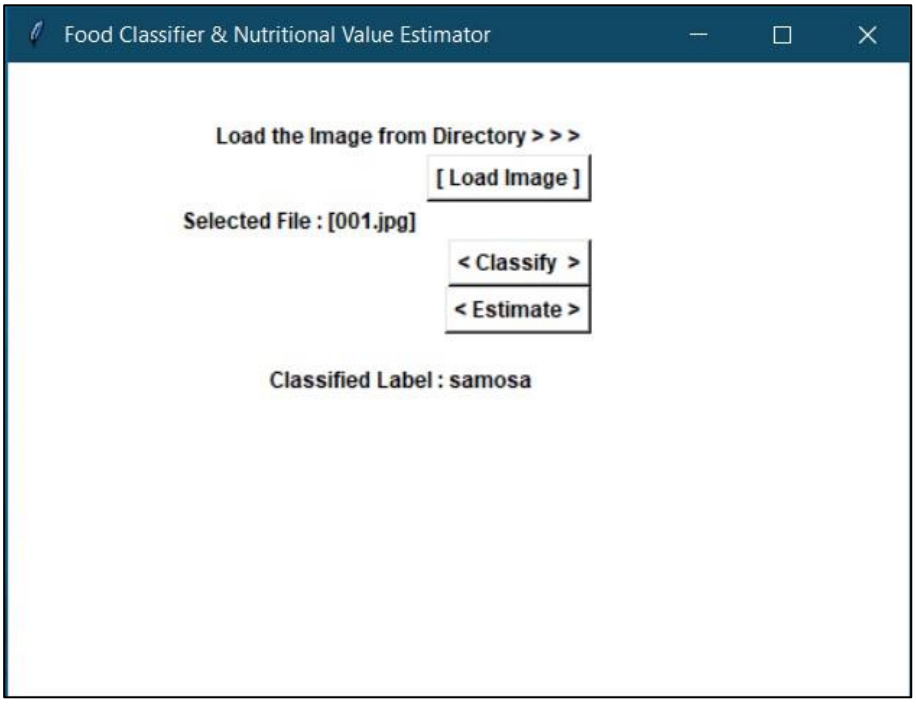

Figure 6.11 Classification of Food Item

6.3 CALORIE ESTIMATION

On selecting “Estimate” a tab with a bounding box around food item is displayed as shown in Figure 6.12 followed by estimation results in home screen as shown in Figure 6.13.

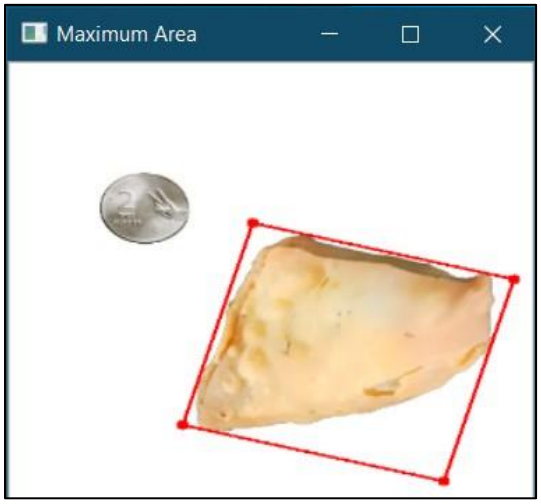

Figure 6.12 Bounding Box to Estimate Dimensions

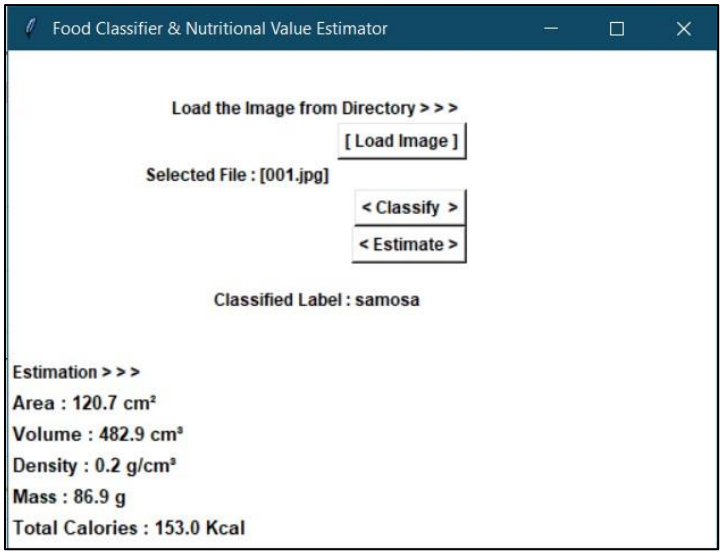

Figure 6.13 Calorie Estimation of the Food Item

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The project's objective to develop a food classifier and a novel system to estimate calories from the food image are accomplished. The model's training accuracy is 86.6% which embraces the confidence over the model. The usage of Food 101 dataset over others helped in employing basic data pre-processing which saved a lot of time.

The Inception v3 model and the image augmentations techniques used, significantly improved the model's accuracy and made it capable of classifying images which are distorted and complex. The mobile application's feature, sending food images to the desktop application is competent. The calorie estimations are lucid and rapid in terms of performance.

This project provides a simple and effective solution to estimate calories and the results obtained from the application are promising. The usage of images taken from smartphones, and the processing techniques used here are coherent. So, this project can be converted or perhaps easily integrated into fitness apps which tends to be minimal without any heavy computation models in backend.

7.2 FUTURE WORK

The Bounding Box takes extra area if the food item's shape is anything other than a rectangle. This increase in area, results in increased calorie count. To balance that, the calories/100 g is modified manually to provide accurate value. To overcome this issue, more sophisticated algorithm needs to be employed for accurate detection of any given shape.

Training with custom image data comprising top view of the food can reduce the hassle of using two different photos in selected labels for classification and estimation.

Dataset comprising regional food items can increase the usage of the application. Also, liquid food items such as milkshakes, yogurt, may be included.

Improved mobile application capable of discarding the usage of desktop application and including the entire stack in the project may be developed.

REFERENCES

[1] Elder Akpro Hippocrate, Hirohiko, Yutaka, and Keiichi (2017), "Smartphone-Based Food Weight and Calorie Estimation Method for Effective Food Journaling", SICE Journal of Control, Measurement, and System Integration, Vol. 10, No. 5, pp. 360–369.

[2] Junyi Gai, Weihao Tan, Liantao Ma, Yasha Wang and Wen Tang (2019), "MUSEFood: Multisensor-based Food Volume Estimation on Smartphones", ArXiv.

[3] Koichi Okamoto, Keiji Yanai (2016), "An Automatic Calorie Estimation System of Food Images on a Smartphone", Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management.

[4] Patrick McAllister, Huiru Zheng, Raymond Bond, Anne Moorhead (2015), "Semi Automated System for Predicting Calories in Photographs of Meals", IEEE International Conference on Engineering, Technology and Innovation/ International Technology Management Conference (ICE/ITMC).

[5] Raza Yunus, Omar Arif, Hamad Afzal (2019), "A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques", IEEE Access Vol No.7, pp. 2643-2652.

[6] Takumi Ege, Yoshikazu Ando, Ryosuke Tanno, Wataru Shimoda and Keiji Yanai (2019), "ImageBased Estimation of Real Food Size for Accurate Food Calorie Estimation", IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).

[7] Tatsuya Miyazaki, Gamhewage C. de Silva, Kiyoharu Aizawa (2011), “Image-based Calorie Content Estimation for Dietary Assessment”, IEEE International Symposium on Multimedia.

[8] Weiqing Min, Shuqiang Jiang, and Linhu Liu (2019), “A Survey on Food Computing”, ACM Computing Surveys.