International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

STANLEY COLLEGE OF ENGINEERING AND TECHNOLOGY FOR WOMEN
Chapel Road, Abids, Hyderabad.

Institute Vision:

Empowering girl students through professional education integrated with values and character to make an impact in the World.

Institute Mission:

M1: *Enabling quality engineering education for girl students to make them competent and confident to succeed in professional practice and advanced learning.*
M2: *Providing state-of-art-facilities and resources towards world class education.*
M3: *Integrating qualities like humanity, social values, ethics, leadership towards their contribution to society.*

Department Vision:

Empowering girl students with the contemporary knowledge in Computer Science and Engineering for their success in life.

Department Mission:

M1: *To impart quality education for girl students to learn and practice various hardware and software platforms prevalent in industry.*

M2: *To achieve self-sustainability and overall development through research and development activities.*

M3: *To provide education for life by focusing on the inculcation of human & moral values through and honest and scientific approach*

M4: *To groom students with good attitude, team work and personality skills.*

Programme Educational Objectives

PEO1: Our graduates shall have enhanced skills and contemporary knowledge in software and hardware technologies for professional excellence, towards successful employment, advanced learning and research.

PEO2: Our graduates shall have life-long learning attitude, innovation and creativity to master latest technologies, devise solutions for realistic and social issues in the society.

PEO3: Our graduates shall have good attitude and personality skills, ethical values, teamwork and leadership skill towards professionalism and ethical practices within the organization and the society.

Program Outcomes:

PO1: Engineering Knowledge: Apply knowledge of mathematics and science, with fundamentals of Computer Science & Engineering to be able to solve complex engineering problems related to CSE.

PO2: Problem Analysis: Identify, Formulate, review research literature and analyze complex engineering problems related to CSE and reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3: Design/Development of solutions: Design solutions for complex engineering problems related to CSE and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural societal and environmental considerations.

PO4: Conduct Investigations of Complex problems: Use research–based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, Select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to computer science related complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply Reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the CSE professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the CSE professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply Ethical Principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary Settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large such as able to comprehend and with write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering management principles and apply these to one‘s own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.

PO12: Life-Long Learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning the broadest context of technological change.

Program Specific outcomes:

PSO1: Problem-Solving Skills: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for the benefit of students.

PSO2: Design, Implement, Test and Evaluate a computer system, component or algorithm to meet desired needs and to solve a computational problem.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Major Project Expected Program Outcomes

PO2: PROBLEM ANALYSIS

PO3:DESIGN/DEVELOPMENT OF SOLUTION
PO5: MODERN TOOLS USAGE
PO6: THE ENGINEER AND SOCIETY
PO9: INDIVIDUAL AND TEAM WORK
PO11: PROJECT MANAGEMENT
PO12: LIFELONG LEARNING

Program Specific outcomes
PSO1: PROBLEM-SOLVING SKILLS
PSO2: DESIGN, IMPLEMENT, TEST

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

FOOD RECOGNITION AND CALORIE ESTIMATION USING CNN

By

M. Samyuktha 160618733155

M. Sreeja 160618733156

P. Srinidhi 160618733161

Department of Computer Science and Engineering

Stanley College of Engineering & Technology for Women (Autonomous)

Chapel Road, Abids, Hyderabad – 500001

(Affiliated to Osmania University, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC with _A' Grade)

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ABSTRACT

Obesity and overweight have traditionally been linked to intake of high calorie food. 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 major 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 many other reasons that lead to monitoring of food calorie estimation. Proposed system will be trained with many images in each category, which then classifies the food image and then estimates nutrition / calorie. Proposed system uses CNN as a classifier for food recognition and based on the food weight in grams the calorie of the food items are determined. This deep convolutional neural network will configure to detect and recognize local food images. Data about most consumed local food items were collected from internet. The main aim of this project is to improve the accuracy of the pre-training model using CNN.

Keywords: Food Recognition, Convolution Neural Network, Calorie Estimation, Volume, food images.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CHAPTER 1
INTRODUCTION

1.1 About Project

Food is the key of human‘s body. So, a diet plan always needs to take into consideration the total number of calories to be consumed to maintain a fit and healthy life. But, in most cases, unfortunately people face difficulties in estimating and measuring the amount of food intake due to the mainly lack of nutritional information, which includes manual process of writing down this information, and other reasons. As such, it will be useful if there is a system to keep track and maintain the calorie intake. Hence accurate prediction of food calorie is equally important in such cases. In the last three years, object classification and detection capabilities have dramatically improved due to advances in deep learning and convolutional neural networks. Harnessing this technology to accurately classify and detect food objects is significantly essential for a healthy and fit life. But to always refer to the nutritional content in each food item is an extremely tedious task. Food image recognition provides an easy means to estimate the dietary caloric intake and evaluate people‘s eating habits, by using cameras to stay track of their food consumption. An accurate estimation of daily nutritional intake provides a useful solution for keeping healthy and to prevent diseases.

Problem statement :

A system will be developed using a computer vision approach that can be used to calculate the number of food calories automatically based on the size of the food volume using the Deep Learning Convolutional Neural Network algorithm.

1.2 Objectives of Project

- Designed to accurately estimate food calories from still images could help users and health professionals identify dietary patterns and food choices associated with health and health risks more .
- To create a layout where people can upload a picture that needs to be estimated.
- To detect the food image by using image processing technique.
- To develop a CNN (Convolution Neural Network), to estimate the number of calories in the food item.

1.3 Scope of Project

- These image classification techniques will help the user to select the desired food item by displaying the calories of food item in online food delivery purpose. It is particularly useful when the users may not know the details about the food items.
- Easy and accurate calorie measurement.

1.4 Advantages

- Edibles are detected in images.
- Our application will display the results immediately after the user inserts the food image thereby decreasing the navigation time of the user. Hence our application is user-friendly.

1.5 Disadvantages

- We have used Food 101 dataset. It consists of 3000 images which is very large. So we have taken 25 classes.
- Recognizing the food item with the help of single picture.

1.6 Applications

- Tracking of calories taken in a day.
- Simple guidelines for gaining or losing weight.
- Motivates to stay healthier.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

1.7 Hardware & Software Requirements

Hardware Requirements

- **System Processor** – Intel Core i3 2.20 Ghz
- **RAM** – 4 GB
- **Hard Disk** - 500GB
- **Key Board** - Standard Keyboard
- **Monitor** – 15 VGA Colour

Software Requirements

- **Operating System:** Windows
- **Coding Language**: Python 3.7
- **IDE**: Jupyter

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

To identify what‘s on the plate, we need to instance-segment the given food image into the possible food categories. Instance Segmentation classifies individual pixel in the given picture into possible classes i.e. foods in our case. Given the problem of instance segmentation, the architecture of CNN would be a matching solution. CNN takes an image and spits out three outputs, masks of the identified items, bounding boxes and classes for each mask detected. Masks are the binary coded single-channel matrices of the size of the input image which denote the boundaries of the identified object. Accuracy is 85%. [3]

Food is the key of human‘s body. So, a diet plan always needs to take into consideration the total number of calories to be consumed to maintain a fit and healthy life. But, in most cases, unfortunately people face difficulties in estimating and measuring the amount of food intake due to the mainly lack of nutritional information, which includes manual process of writing down this information, and other reasons. As such, it will be useful if there is a system to keep track and maintain the calorie intake. mAP is only 75.9% [2]

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. Obesity in childhood and adolescents is associated a number of complications like, psychological stress, cardiovascular disease, orthopedic problems, etc. Obesity is usually preventable through a mixture of social changes and private choices.

Improved the functionality and flexibility of the recognition system by adding shape and size features as well as colour and texture. It gives the accuracy is 89%.[12]

- Calculating the number of calories digitally from food requires the parameters of area, volume

and mass of food.[13] Some previous studies in the field of Computer Vision (CV) have been carried out to get a constant number of calories based on food types and not based on actual food volume measurements.

2.2 Proposed System

People often take pictures of their food before they eat and put it on social media, well the solution lies in that very process. We propose to estimate the calorie content in the user-provided image by identifying the food and estimating the quantity using deep learning. To give an estimation of the calories we need accurate object detection combined with accurate IoU (intersection over the union). An impressive amount of IoU can be achieved using Single Shot Detections which are also faster than their counterparts but the problem is with the segmentation. We cannot approximate the amount with the output of Bounding Box, we need more precision. So, the solution would be to use Instance Segmentation.

Advantages of proposed system:

Food images are detected.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CHAPTER 3

PROPOSED ARCHITECTURE

SOFTWARE TECHNOLOGIES

What is Python:-

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like -- Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

Machine Learning

- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

What is Machine Learning: -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush[11]. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Categories of Machine Learning:-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Applications of Machines Learning:-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Customer segmentation
- Object recognition
- Recommendation of products to customer in online shopping

Purpose:-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Modules Used in Project :-

Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine

learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits..

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes

properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

3.1 Detailed Design

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it‘s one of the most popular business process modeling techniques.

It is based on **diagrammatic representations** of software components. As the old proverb says: ‒a picture is worth a thousand words‖. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Support higher level development concepts such as collaborations, frameworks, patterns & components.
6. Integrate best practices.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

i. USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

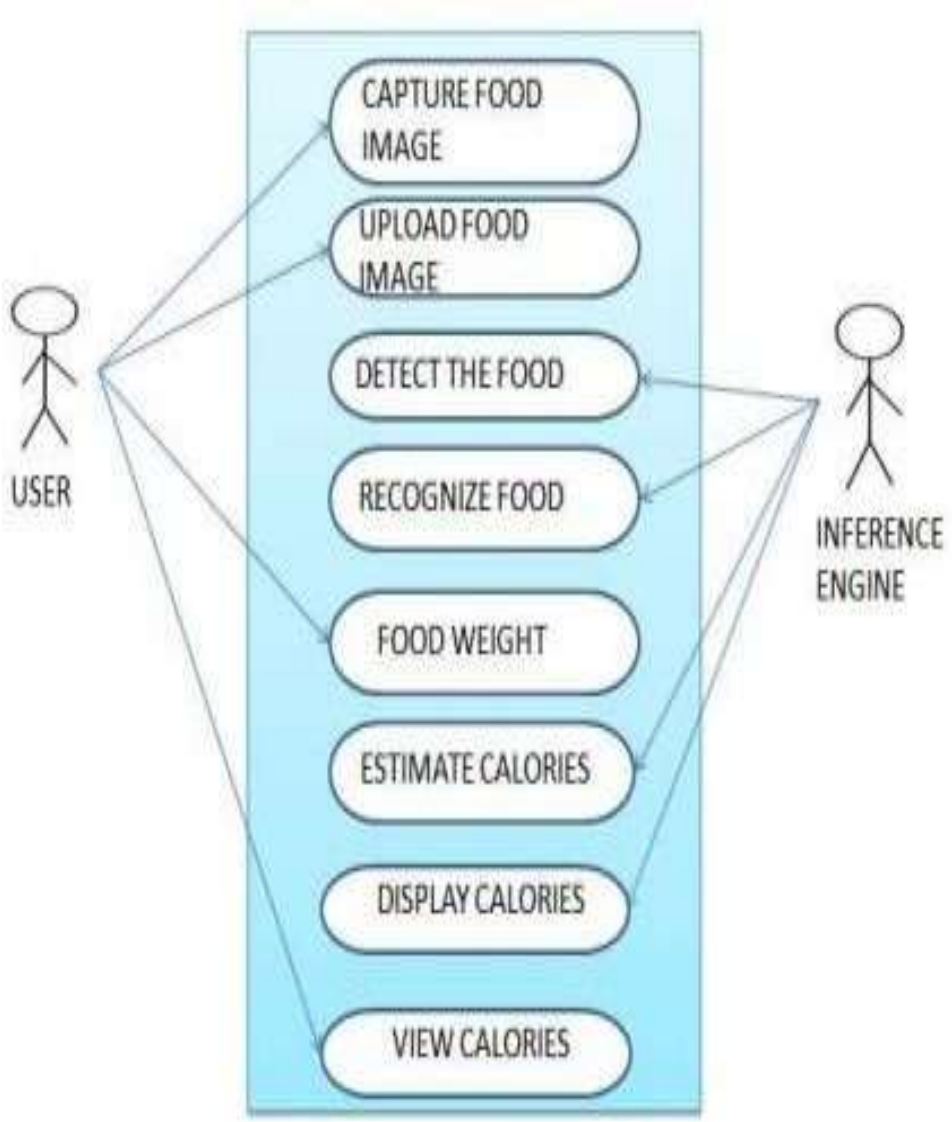


Fig 3.1.1: Use case diagram

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ii. SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

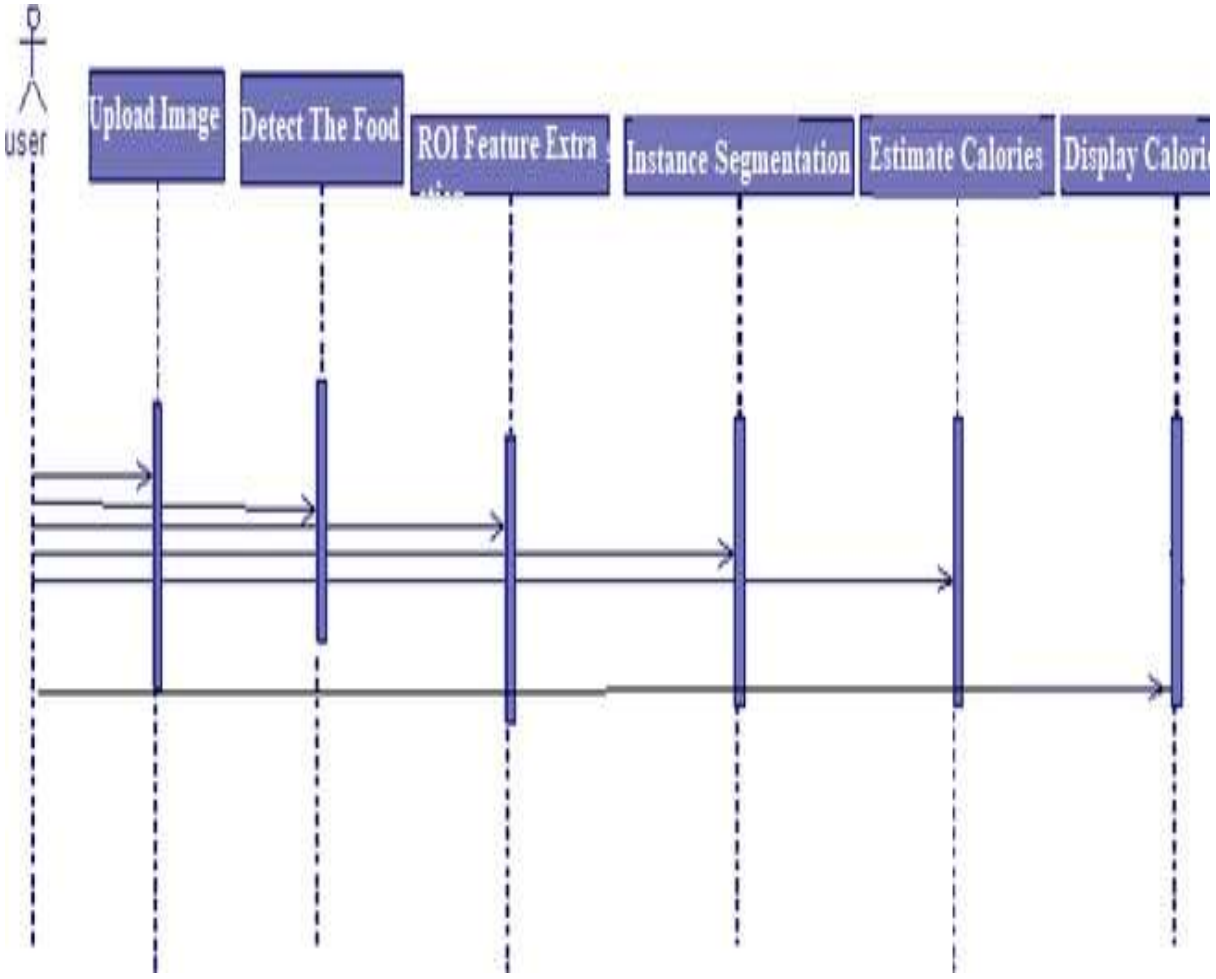


Fig 3.1.2: Sequence diagram

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

iii. CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

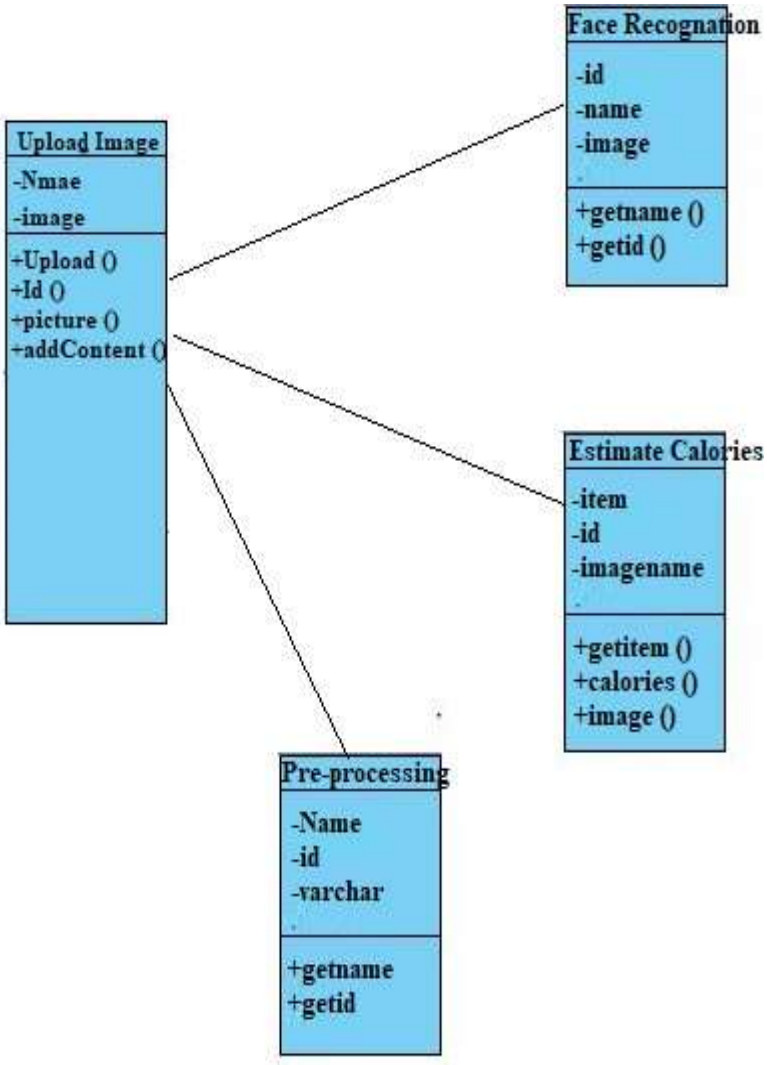
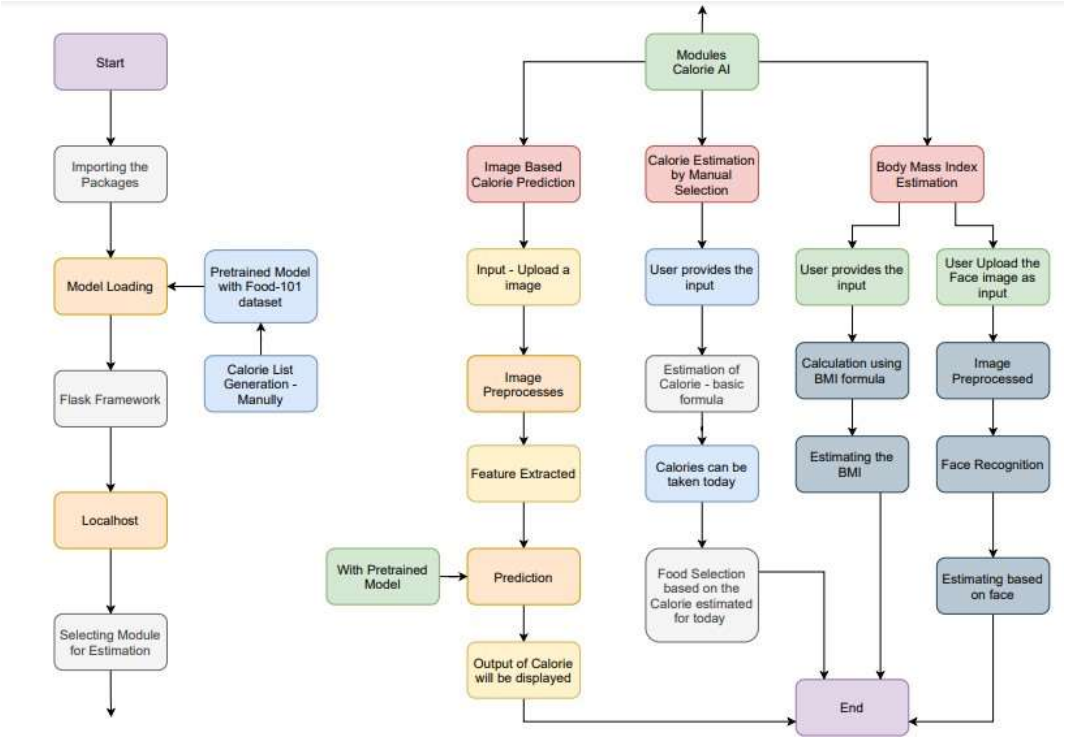


Fig 3.1.3: Class Diagram

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

iv. DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Fig 3.1.4: Data flow Diagram

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

v Component Diagram :-

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

UML Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

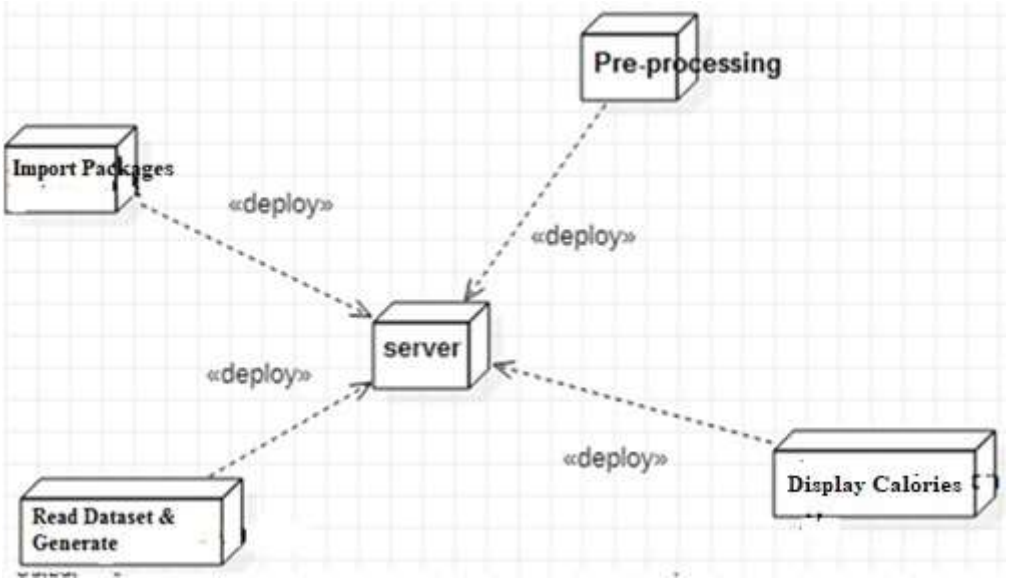


Fig 3.1.5: Component Diagram

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

vi ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

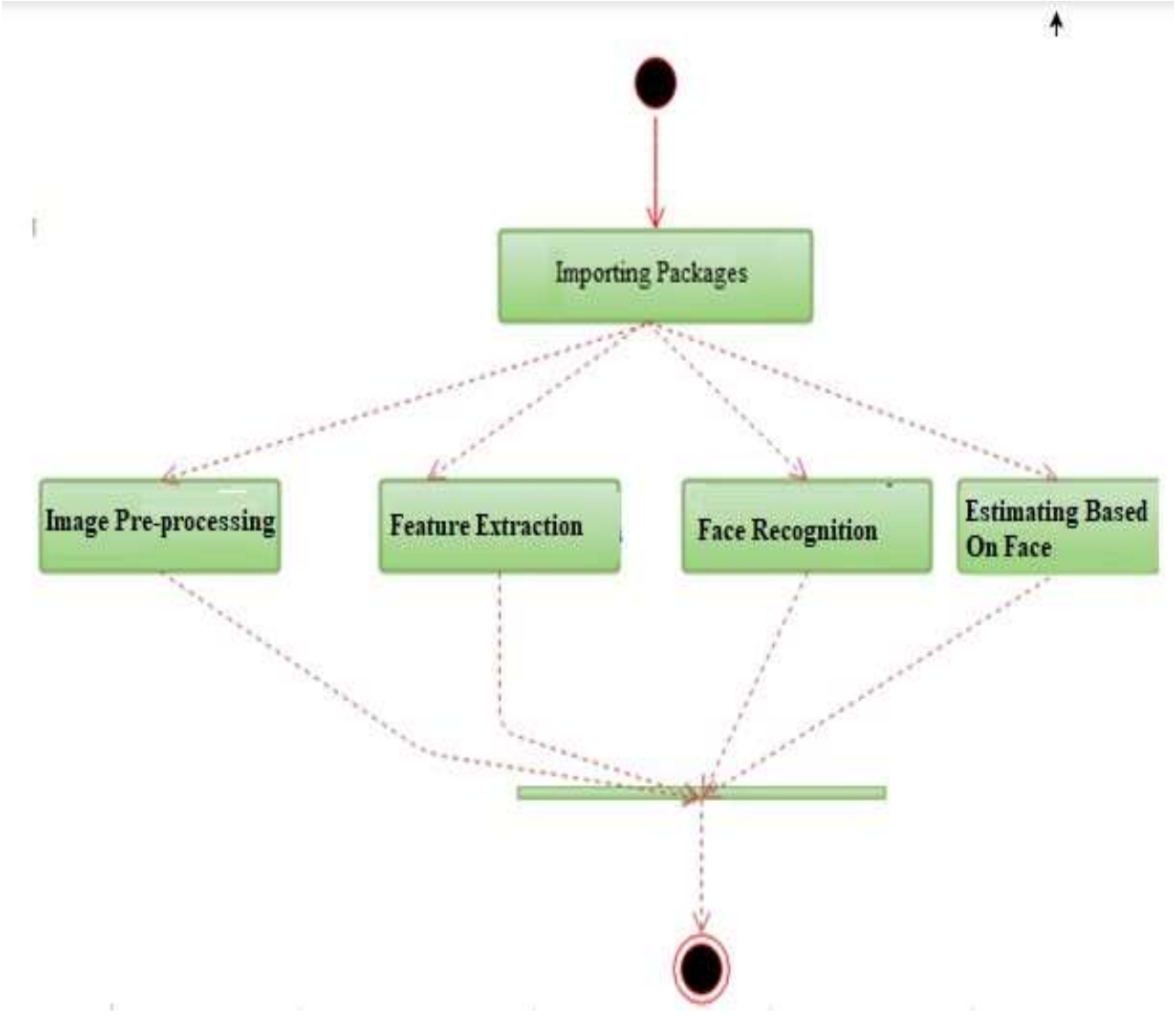


Fig 3.1.6: Activity Diagram

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CHAPTER 4

IMPLEMENTATION

4.1 Algorithm

Algorithm used in this project :-

CONVOLUTION NEURAL NETWORK

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[8]. 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.

CNN Specifications: The most popular readily available dataset for image classification is the ImageNet database, which has been used to train the CNN. It also has multiple existing classification categories. To generalize the system, another 25 categories are added to this model by training it on the Food-101 dataset. The input sensor has a size of 299x299x3, with a Max pooling downscale of 2 in each spatial dimension with a dropout rate of 0.4 and the softmax activation function[5]. The optimization function used for this task is stochastic gradient descent, which basically finds the maxima and minima through optimal number of iterations.

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, donuts, waffles, sushi, nachos.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

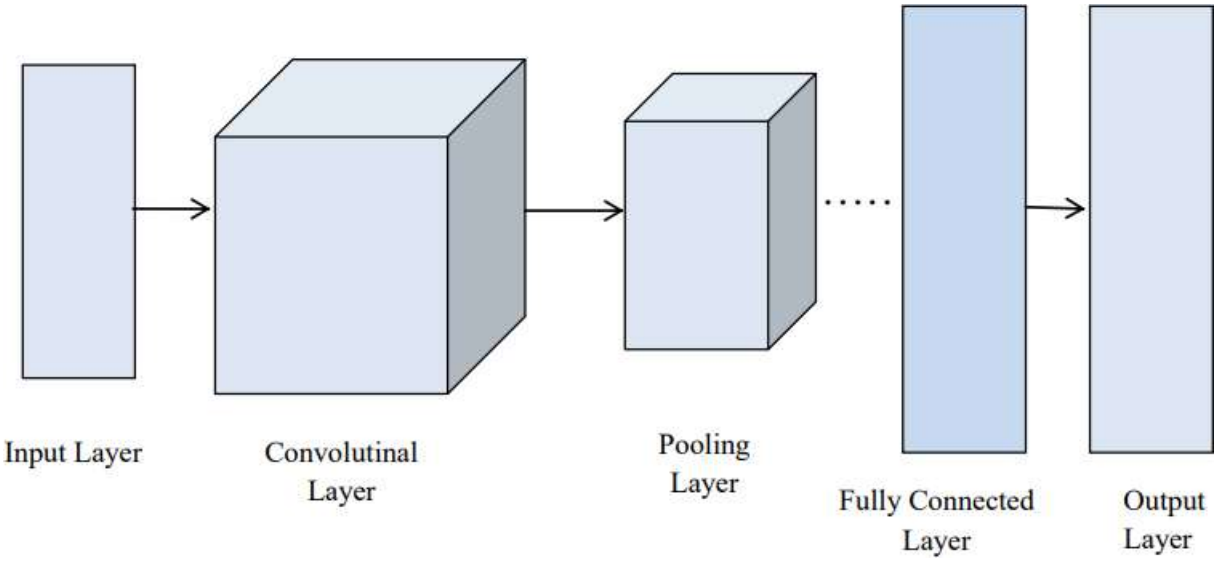


Fig 4.1.1 Layers of a Convolutional Neural Network

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[7]. 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.

The convolutional Neural Network (CNN) is used to recognize the food item. Further to estimate the calories we have given the standard calorie value for one gram of each food item. The weight of the food item is given as an input and based on the standard calorie value the accurate calorie value of the food item is calculated.[1]

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take in an input image, assign importance to various considerations in the image, and be able to distinguish one from the other. It is a multilayer neural network, whose neurons take small shifts and rotations. CNN‘s are generally a configuration of three types of layers. Convolutional layer, fully connected neural network, apply a number of convolution filter with specific weight (n x n) to the input image. For each section of the image, a set of mathematical operations is applied to produce a single value in the output. Each input convolves these

filters. Each layer has many filters that generate several outputs. The filter is called a kernel. The second type of layer is the pooling layer, which produces a down sample of the resulting image produce by the convolution layer to reduce the size of the feature map for faster processing time.

There are several algorithms such as maximum polling and average pooling. A widely used algorithm is maximum pooling. This makes the CNN output more invariant with respect to position Fully connected layer, perform classification on the extracted feature after down sampling by a pooling layer. Each unit of the final layer represents the class probability. This layer is used to enumerate the score classes.

Calorific Value Estimation: The remaining task after the process of classification is mapping the food names to a calorific value. This can be achieved relatively easily by scraping the web for the average calorie value of food items per unit weight. The average calorie values are considered for the different classes food, per 100g of serving.[10]

4.2 Code Implementation

```
from__future__import absolute_import
from__future__import division
from__future__import print_function

import argparse
import hashlib
import os.path
import random
import re
import sys

import numpy as np
from six.moves import urllib
import tensorflow as tf

from tensorflow.python.framework import graph_util
from tensorflow.python.framework import tensor_shape
from tensorflow.python.platform import gfile
from tensorflow.python.util import compat
```

```
FLAGS = None
MAX_NUM_IMAGES_PER_CLASS = 2 ** 27 - 1 # ~134M
```

```
def create_image_lists(image_dir, testing_percentage, validation_percentage):
  if not gfile.Exists(image_dir):
    tf.logging.error("Image directory '" + image_dir + "' not found.")
    return None
  result = collections.OrderedDict()
  sub_dirs = [
  os.path.join(image_dir,item)
    for item in gfile.ListDirectory(image_dir)]
  sub_dirs = sorted(item for item in sub_dirs
                    if gfile.IsDirectory(item))
  for sub_dir in sub_dirs:
    extensions = ['jpg', 'jpeg', 'JPG', 'JPEG']
    file_list = []
    dir_name = os.path.basename(sub_dir)
    if dir_name == image_dir:
      continue
    tf.logging.info("Looking for images in '" + dir_name + "'")
    for extension in extensions:
      file_glob = os.path.join(image_dir, dir_name, '*.' + extension)
      file_list.extend(gfile.Glob(file_glob))
    if not file_list:
      tf.logging.warning('No files found')
      continue
    if len(file_list) < 20:
      if percentage_hash < validation_percentage:
        validation_images.append(base_name)
      elif percentage_hash < (testing_percentage + validation_percentage):
        testing_images.append(base_name)
      else:
        training_images.append(base_name)
    result[label_name] = {
```

'dir': dir_name,

'training': training_images,

'testing': testing_images,

```
      'validation': validation_images,
    }
  return result
def get_image_path(image_lists, label_name, index, image_dir, category):
  if label_name not in image_lists:
    tf.logging.fatal('Label does not exist %s.', label_name)
  label_lists = image_lists[label_name]
  if category not in label_lists:
    tf.logging.fatal('Category does not exist %s.', category)
  category_list = label_lists[category]
  if not category_list:
    tf.logging.fatal('Label %s has no images in the category %s.',
                     label_name, category)
  mod_index = index % len(category_list)
  base_name = category_list[mod_index]
  sub_dir = label_lists['dir']
  full_path = os.path.join(image_dir, sub_dir, base_name)
  return full_path
  return graph, bottleneck_tensor, resized_input_tensor
def run_bottleneck_on_image(sess, image_data, image_data_tensor,
                            decoded_image_tensor, resized_input_tensor,
                            bottleneck_tensor):
  resized_input_values = sess.run(decoded_image_tensor,
                                  {image_data_tensor: image_data})
  bottleneck_values = sess.run(bottleneck_tensor,
                               {resized_input_tensor: resized_input_values})
  bottleneck_values = np.squeeze(bottleneck_values)
  return bottleneck_values
def maybe_download_and_extract(data_url):
def variable_summaries(var):
  with tf.name_scope('summaries'):
```

```
mean = tf.reduce_mean(var)
tf.summary.scalar('mean', mean)
with tf.name_scope('stddev'):
```

```
    stddev = tf.sqrt(tf.reduce_mean(tf.square(var - mean)))
  tf.summary.scalar('stddev', stddev)
  tf.summary.scalar('max', tf.reduce_max(var))
  tf.summary.scalar('min', tf.reduce_min(var))
  tf.summary.histogram('histogram', var)

def add_final_training_ops(class_count, final_tensor_name, bottleneck_tensor,
                           bottleneck_tensor_size):
 with tf.name_scope('input'):
  bottleneck_input = tf.placeholder_with_default(
      bottleneck_tensor,
      shape=[None, bottleneck_tensor_size],
      name='BottleneckInputPlaceholder')

  ground_truth_input = tf.placeholder(tf.float32,
                                      [None, class_count],
                                      name='GroundTruthInput')
 layer_name = 'final_training_ops'
 with tf.name_scope(layer_name):
  with tf.name_scope('weights'):
   initial_value = tf.truncated_normal(
       [bottleneck_tensor_size, class_count], stddev=0.001)

   layer_weights = tf.Variable(initial_value, name='final_weights')

   variable_summaries(layer_weights)
  with tf.name_scope('biases'):
   layer_biases = tf.Variable(tf.zeros([class_count]), name='final_biases')
   variable_summaries(layer_biases)
  with tf.name_scope('Wx_plus_b'):
   logits = tf.matmul(bottleneck_input, layer_weights) + layer_biases
```

```
    tf.summary.histogram('pre_activations', logits)

final_tensor = tf.nn.softmax(logits, name=final_tensor_name)
```

```
  tf.summary.histogram('activations', final_tensor)
  with tf.name_scope('cross_entropy'):
    cross_entropy = tf.nn.softmax_cross_entropy_with_logits(
        labels=ground_truth_input, logits=logits)
    with tf.name_scope('total'):
      cross_entropy_mean = tf.reduce_mean(cross_entropy)
def add_evaluation_step(result_tensor, ground_truth_tensor):
with tf.name_scope('accuracy'):
    with tf.name_scope('correct_prediction'):
      prediction = tf.argmax(result_tensor, 1)
      correct_prediction = tf.equal(
          prediction, tf.argmax(ground_truth_tensor, 1))
    with tf.name_scope('accuracy'):
      evaluation_step = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
  tf.summary.scalar('accuracy', evaluation_step)
  return evaluation_step, prediction
def prepare_file_system():
  if tf.gfile.Exists(FLAGS.summaries_dir):
  tf.gfile.DeleteRecursively(FLAGS.summaries_dir)
  tf.gfile.MakeDirs(FLAGS.summaries_dir)
  if FLAGS.intermediate_store_frequency > 0:
  ensure_dir_exists(FLAGS.intermediate_output_graphs_dir)
  return
    if (size_string != '224' and size_string != '192' and
      size_string != '160' and size_string != '128'):
      tf.logging.error(
          """The Mobilenet input size should be '224', '192', '160', or '128',
 but found '%s' for architecture '%s'""",
          size_string, architecture)
      return None
    if len(parts) == 3:
```

```
is_quantized = False
else:
  if parts[3] != 'quantized':
```

```
tf.logging.error(
    "Couldn't understand architecture suffix '%s' for '%s'", parts[3],
    architecture)
return None
```

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CHAPTER – 5

RESULT

Fig 5.1.1: Before uploading the food image

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

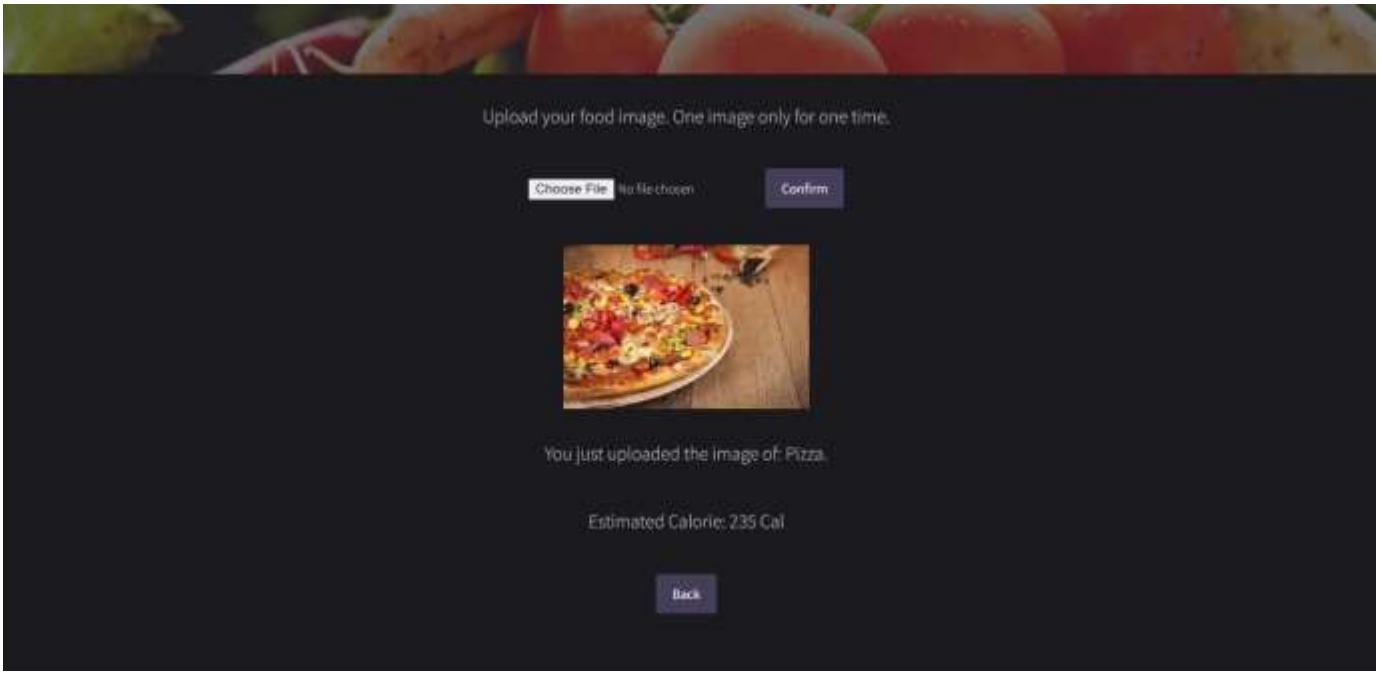


Fig 5..1.2: After uploading the food image

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

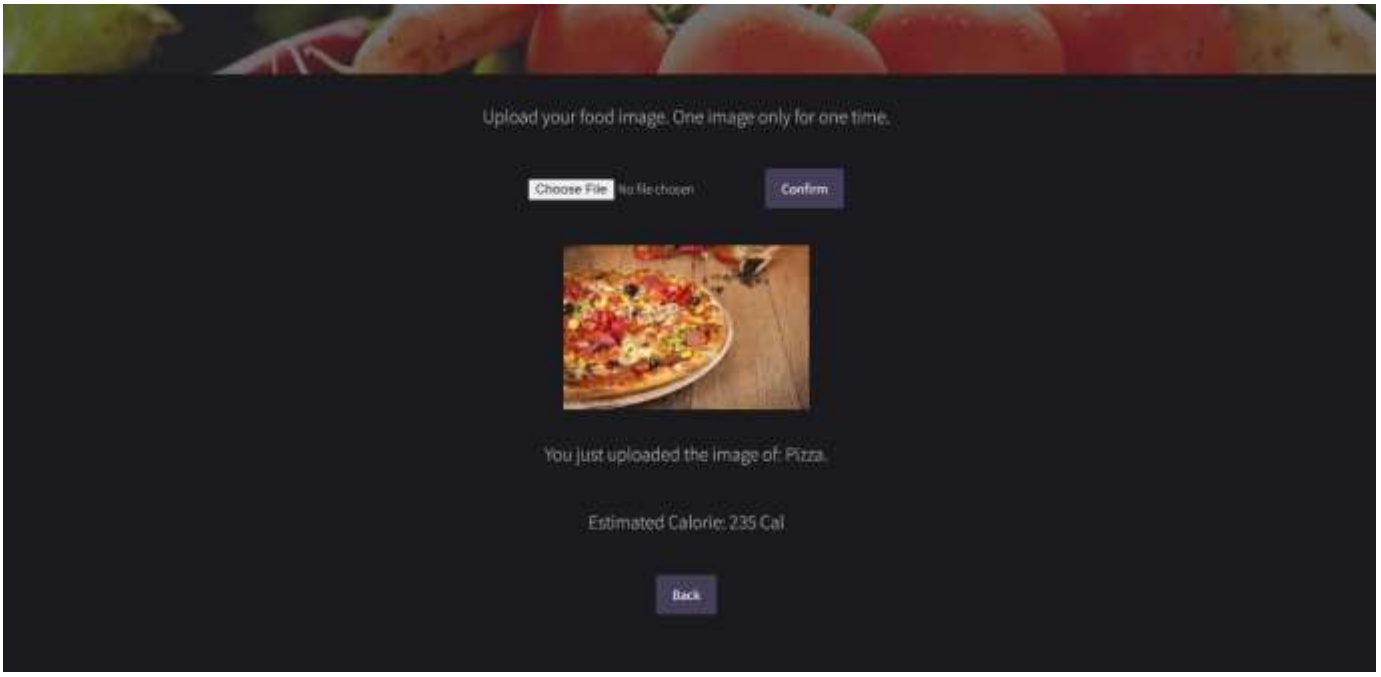


Fig 5.1.3: After uploading the image with estimated calories in it

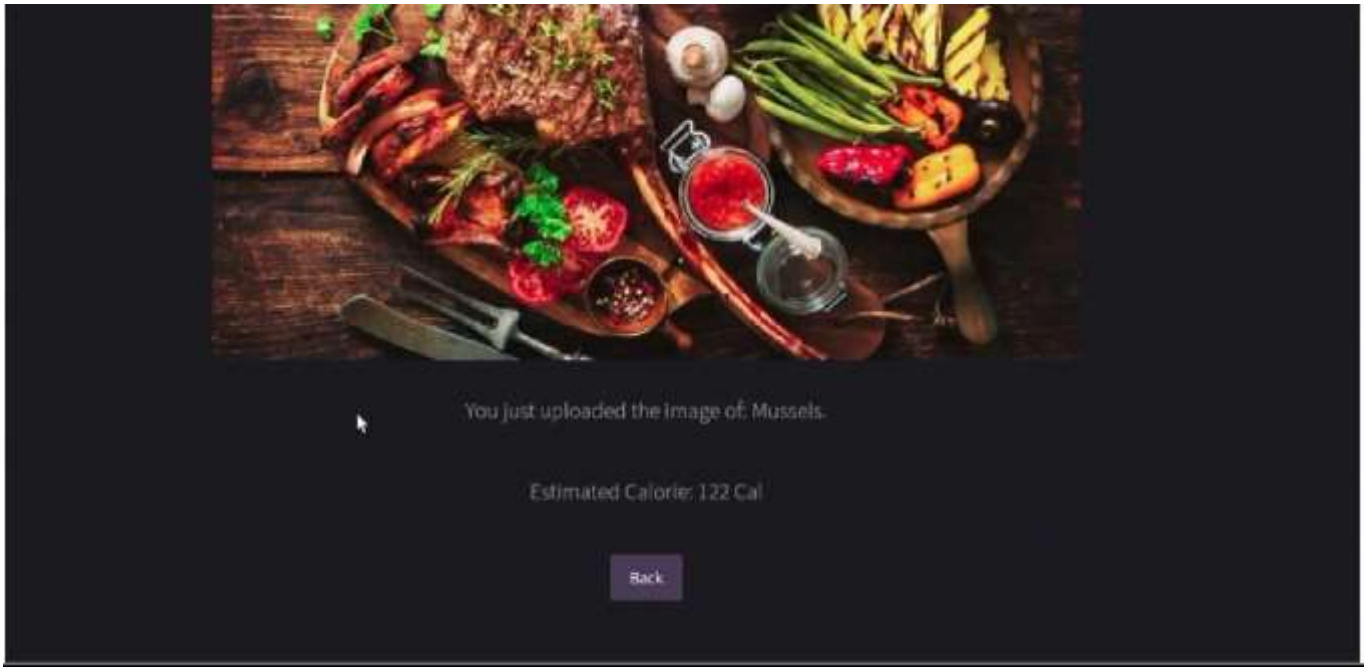


Fig 5.1.4: after uploading other image with estimated calories in it

TEST RESULTS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application
.it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from

a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot –see into it. The test provides inputs and responds to outputs without

considering how the software works.

5.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 6 CONCLUSION

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.

CHAPTER 7 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 .With this brief testing of food images dataset with CNN, we can deduce that it is quite possible to achieve an application that is capable of estimating calories from food images. The application will have a tremendous impact on how people perceive a plate of food and will also impact the weight-loss and weight-management market.

CHAPTER-8

REFERENS

1. Deep Learning-Based Food Calorie Estimation Method in Dietary Assessment, Yanchao Lianga, Jianhua Lia. a School of Information Science and Engineering, East China University of Science and Technology, China.
2. Y. Kawano and K. Yanai, "Food Image Recognition with Deep Convolutional Features," 13 September 2014. [Online]. Available: http://ubicomp.org/ubicomp2014/proceedings/ubicomp_adjunct/work shops/CEA/p589-kawano.pdf.
3. Manal Chokr, Shady Elbassuoni Calories Prediction from Food Images, Department of Computer Science, American University of Beirut, Beirut, Lebanon, mmc35@mail.aub.edu - se58@aub.edu.lb
4. Interactive Foreground Extraction using GrabCut Algorithm, https://docs.opencv.org/trunk/d8/d83/tutorial_py_grabcut.html
5. Boykov, Y., and Jolly, M.-P. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In Proc. IEEE Int. Conf. on Computer Vision.
6. https://grabcut.weebly.com/background--algorithm.html [7]ijariie.com/AdminUploadPdf/Food_Recognition_And_Calorie_Estimation_Using_Image_Processing
7. _ijariie11454
8. V Balaji Kasyap School of Engineering and Technology CHRIST (Deemed to be University), India
9. Meghana M Reddy, –Calorie-estimation-from-food-images-open cv‖, Git repo , May 2016
10. https://www.semanticscholar.org/paper/Food-101-Mining-Discriminative-Components-with-
11. Bossard-Guillaumin/8e3f12804882b60ad5f59aad92755c5edb34860e
12. https://jakevdp.github.io/PythonDataScienceHandbook/05.01-what-is-machine-learning.html
13. Ms. Ankita –Food-Recognition-System-for-Calorie-Measurement‖