ABSTRACT

The importance of calorie tracking is gaining wide acceptance not only in the fitness sector but also in the daily lives of people. It becomes important to ensure that the process is simple, fast, and effective. The existing fitness applications require manual entry of food items to calculate calories intake. The advancement in the field of computation and computer vision has led to the rise of deep learning algorithms like CNN and platforms such as TensorFlow and Keras for effective and accurate image classification. In this paper, a fitness application that uses image processing to detect the food name and calculate the calories has been implemented. We have developed a mobile application that possesses the ability to track calories based on food images using CNN trained neural network model. The mobile application serves ingredient based suggestive predication, which allows the user to select their choice of ingredient and then view the calorie count. The application also provides feature to calculate BMI. We found out about the advantage CNN has over traditional ML approach, in CNN we did not have to hard-code the feature extractors, CNN algorithm automates the process. We found out that features like corner, texture, shape, and edge were extracted by neural network and that these features are automatically selected and updated.

C O N T E N T S

List of Figures

List of Tables

Chapter 1: INTRODUCTION

1.1. Importance of the project

The importance of calorie tracking is gaining wide acceptance in health and fitness sector. It becomes important to ensure that process is simple, fast, and effective. Estimation of right nutrient intake through food consumption has become increasingly significant to maintain the proper health of the wellbeing. The existing fitness applications require manual entry of food items to calculate calorie intake.

The growing need of diet tracking in the world has evolved many applications that shows user the calories intake a food product can provide, this project does it in a different manner. It will provide the option of taking the input as an image. The application will take the image of the food product as the input from the user and process it to show the output i.e., the calories and other nutrition contents like proteins, fats, and carbohydrates in the food item. The most significant component of this project is image classification model designed using Deep learning's Convolutional Neural Network model trained on relevant dataset. The calorie tracking application has been designed using Flutter SDK along with TensorFlow Lite framework's support to load CNN model and predict food name using user input image.

1.2. Literature Survey

Ch. Kavya, R. Priyadarsini and B. Madhavi in their paper titled "Calories and Nutrition Measurement from the Image of Food" designed a system to extract features like color, shape, size, and texture [1]. These were given to the K-nearest neighbor (KNN) for recognizing the food and then the calorie value was measured with the help of nutrition dataset. Hemalatha Reddy V., Kumari S., Muralidharan V., Gigoo K., Thakare B.S. talked about a similar system in their paper- "Food Recognition and Calorie Measurement Using Image Processing and Machine Learning Techniques". They implemented with help of K-Means clustering [2]. I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek's paper "A brief introduction to OpenCV," discussed about OpenCV as an effective tool for image processing. OpenCV is an open-source library for image and video analysis, originally introduced more than decade ago by Intel. It is important to note that

OpenCV is considered by many to be side by side with many commercial image processing packages, and yet it is an open-source tool [3].

The conventional approach of image classification has been pre-processing, feature extraction and classification by using classifiers like SVM, KNN, ANN, K-Means etc. These models have certain limitations as they are based on Machine Learning algorithms. The existing projects have managed to detect the food item and calories of naturally available food items such as fruits and vegetables, but these models fail in case of processed food. The concept of food detection is purely dependent upon features such as shape and color that are extracted from food image, in case of change in recipe the model may not give accurate results. There are numerous fitness applications that are available in market but none of them offer a mechanism of calorie tracking through food image. The existing work have tackled relatively small range of food items. In real life scenario, the variety

of food available is huge hence algorithms like SVM, K means clustering and KNN cannot handle the large volume of dataset. This can lead to model failure or loss in accuracy of detection. Machine learning uses a set of algorithms to analyze and interpret data, learn from it, and based on the learnings, make best possible decisions.

On the other hand, Deep learning structures the algorithms into multiple layers to create an “artificial neural network. The most important difference between deep learning and traditional machine learning is its performance as the scale of data increases. When the data is small, deep learning algorithms do not perform that well. This is because deep learning algorithms need a large amount of data to understand it perfectly [4].

Hokuto Kagaya, Kiyoharu Aizawa, Makoto Ogawa implemented a CNN model in their paper " Image-based Calorie Content Estimation for Dietary Assessment,". They applied a convolutional neural network (CNN) to the tasks of detecting and recognizing food images [5]. Because of the wide diversity of types of food, image recognition of food items is generally very difficult. However, deep learning has been shown recently to be a very powerful image recognition technique, and CNN is a state-of-the-art approach to deep learning. [6]

1.3. Motivation

In recent times with easy access to internet, food is delivered at our doorsteps just on the click of a button due to which people have started to consume higher amount of fast food. The consumption of ready-made food items has been on the rise and are being sold in large numbers, these products have a profound effect on health as they are constituted with a lot of preservatives and chemical substances to increase their shelf life, they are not suitable for daily consumption and are

dangerous. This has accelerated the chances of suffering from a chronic disease known as obesity which creates problems in the basic functioning of the human body. Since obesity has become such a widespread disease, various mobile e-health applications have been developed for assistive calorie measurement to help people fight against health-related problems, hence increasing the demand of diet tracker applications. Estimation of right nutrient intake through food consumption has become increasingly significant to maintain the proper health of the wellbeing. If people are aware of the amount of food that they consume and the number of calories they consume, it can help them to check their intake and help stay fit and healthy.

1.4. Scope of the Project

The project caters to the fitness and health industry that is looking for strategies to make calorie tracking an efficient and simpler process. The project offers an add on feature to the existing technology, as well as scope to be scaled up as a separate product. The project can be used by gym trainers as well as nutritionists to check the daily calorie intake of their clients which can help them determine the type of training, exercises, or diet one needs to follow to keep themselves fit.

A person cannot rely on memory to recall the food they consumed during the day. Even if they can recall what they ate, it is nearly impossible to track every meal. Hence, they are not likely to get a precise calorie number if their food journal is incomplete. Calorie counters and nutrient tracking-based smartphone applications are incredibly useful if an individual is trying to lose, maintain or even gain weight. These types of projects can prove useful for such an audience if it is made available as an application on Google Play store and Apple's App Store for download.

1.5. Organization of the project report

The material presented in this book is organized into six chapters. After this introductory chapter, chapter 2 describes the "Proposed Work" for solving the Image Classification problem and developing a mobile application to measure calories from food images. Chapter 3 summarizes requirement gathering, planning and analysis performed in development of project. It contains information about project roadmap, technology used, and methodology adopted. It also provides an account of System analysis performed on CNN model.

Chapter 4 provides an account of the design phase in development of mobile application. It consists of information about algorithm, database, user interface etc. Chapter 5 presents brief a brief conclusion about work undertaken and future scope. It also consists of list of references used.

Chapter 2: PROPOSED WORK

2.1. Problem Definition

To develop an image processing-based model to classify and calculate calories from food images. The model should be able to identify various classes of Indian food images and estimate the calories

2.2. Data Flow Diagram

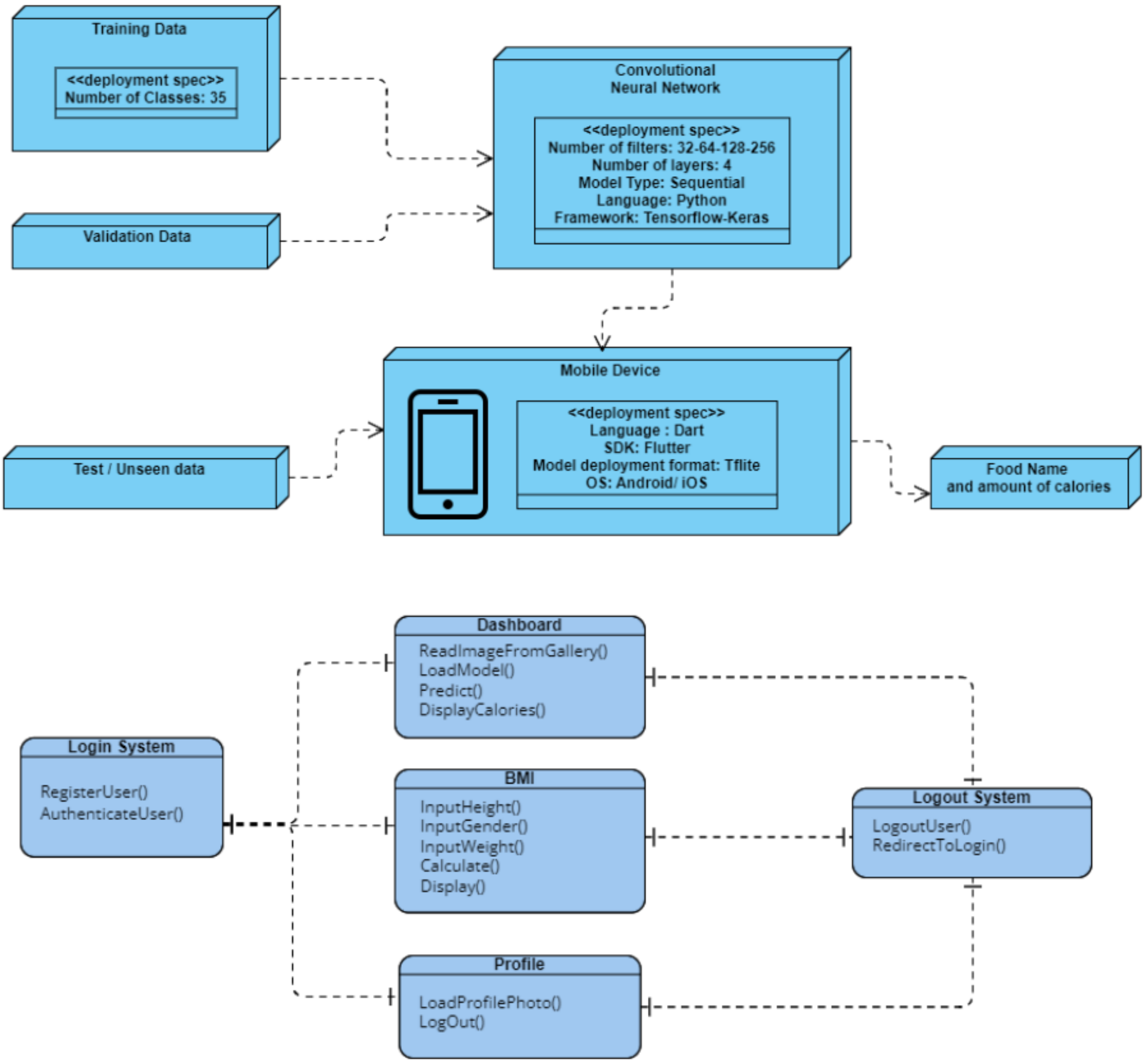


Figure – 2.1: Data Flow Diagram

The project was divided into two phases viz. Image classification and Application Development. We used CNN algorithm to create a image classification model. A training data of 35 food labels was used to train the algorithm. The model was validated on validation data set of 35 food labels to calculate accuracy and loss of model. Tensorflow library of Python provides Keras module for training CNN model. The weights and bias of model was stored in Keras compatible (.h5 extension) file and eventually converted to Tensor Flow Lite (.tflite extension) for deployment in Android application. The mobile application developed in Flutter- Dart language consists of gallery image reader functionality that allows an user to test input food image from phone's gallery. The image is displayed on dashboard along with calories and other nutritional information. The mobile application consists of several screens like authentication, BMI, profile, and logout.

Chapter 3: ANALYSIS AND PLANNING

3.1. Feasibility Study

The project will cater to the audience who are fitness and health conscious. Our project will help make such an audience that is looking for strategies to make calorie tracking an efficient and simpler process. The project offers a database which contains a variety of Indian food items which are frequently consumed, it will act as a unique feature and as well as lend scope to be scaled up as a separate product. The project can be used by gym trainers as well as nutritionists to check the daily calorie intake of their clients which can help them determine the type of training, exercises, or diet one needs to follow to keep themselves fit.

If an application allows the user to be notified with the number of calories from a food image, it might help the user to track the calorie intake easily. A person cannot rely on memory to recall the food they consumed during the day. Even if they can recall what they ate, it is nearly impossible to track every meal. Hence, they are not likely to get a precise calorie number if their food journal is incomplete. Calorie counters and nutrient tracking-based smartphone applications will be extremely useful for an individual who is trying to lose, maintain or even gain weight, basically such an application will help track the calorie intake. This kind of mobile application can turn out to be quite useful to society as it provides an ease to measure dietary intake.

3.2. Project Planning

To successfully complete a project of this magnitude, a well-sorted and efficient plan was to be put into action. Firstly, it was important to decide what will be our final goal and what steps will be followed to achieve it.

The first approach considered was Machine Learning, image recognition using machine learning was taken up and research was done on how it could be implemented for the project. Other alternatives were brought up to notice while researching, Deep Learning algorithm was one which could make our work easier and thus we finally selected it for our image classification problem.

A reliable and a generic dataset was required which will provide us with a wide range of various food items over which the model could be trained. To make the training set more precise, some images were needed to be manually added so that the model would be well aware of the kind of day-to-day images the general audience might use while. Pre-processing was to be done over the final dataset as the images present could differ in dimensions, to maintain uniformity, the images were reshaped and resized. Data augmentation was then to be done over the dataset so that the model could be fed with images with different orientations, this would make the overall efficiency of the model better.

The project would use Convolutional Neural Network for image classification. Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign weights and biases to various aspects/objects in the image and then is able to differentiate among them. The Convolutional Neural Network model was created using Python's Keras framework and training and validation data were fitted on the model. To decide upon the correct selection of parameters hit and trail approach was to be done to select appropriate values for model. Various combinations of number of filters, activation functions and number of epochs were to do, and changes would be observed in validation accuracy and loss. Training Accuracy and Training Loss are metrics associated with model's performance with training dataset i.e., seen data, whereas Validation accuracy and validation loss refers to performance of model w.r.t. unseen data. The output of CNN model is an integer corresponding to label defined in training. A map/dictionary of indices as key and food name as value was created. The trained model along with weights would be saved in h5 extension, which could be further converted into tflite – TensorFlow Lite, an android/iOS suitable extension which can used in developing mobile application.

A mobile application was to be built to enhance the reach of the project and make it much more usable. The mobile application would have several screens, including login, home page, profile, and so on. The user input option to upload an image from the mobile gallery will be available on the main page. The image processing model is loaded, and the food index is predicted as soon as the user uploads an image. Within the program, the projected meal index is utilized as a key to find the correct calories and nutrition information, such as proteins and carbs, from a text file. In mobile applications, the concept of file handling will be used extensively.

3.3. Scheduling

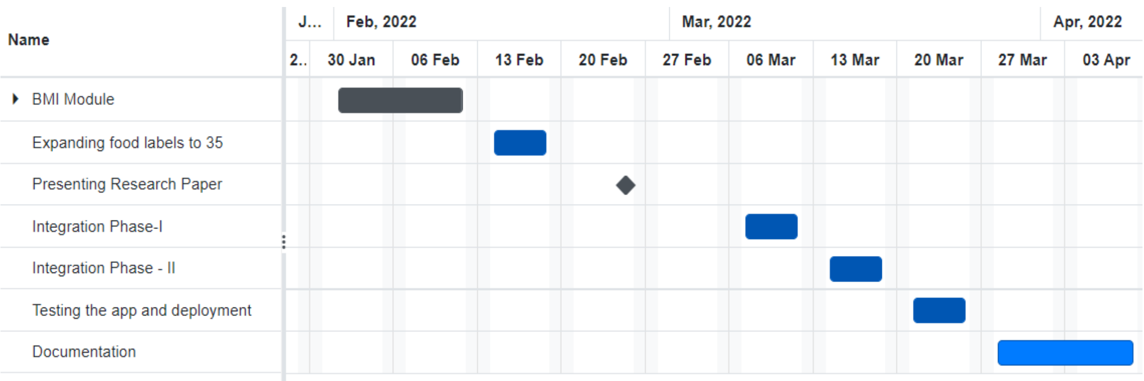


Figure – 3.1: Scheduling

3.4. Hyper-tuning the parameters of CNN

At the initial stage of our project, the developed h5 model consisted of only 17 food items, which were further increased to 35 items. But on increasing the number of items, there was a significant drop in validation and training accuracy. This was solved by increasing the number filters in CNN architecture and by increasing the number of epochs while training on the advice given by our guide.

Chapter 4: DESIGN PHASE

4.1. Technology

Table – 4.1: Technology used for Image Classification

Sr. No.	Tools	Specifications
1	IDE	Google Colaboratory
2	Frameworks / Libraries	Tensorflow – Keras 2.6.0 NumPy Matplotlib
3	Hardware (Cloud based GPU)	NVIDIA-SMI 470.63.01 Tesla K80
4	Language	Python 3.8.1

Table – 4.2: Technology used for Mobile Application development

Sr. No.	Tools	Specification
1	Language	Dart Programming
2	Framework	Flutter SDK - 1.0.0+1 for Android and iOS application development.
3	Libraries for various widgets	cupertino_icons: 0.1.2 flutter_svg: 0.19.1 image_picker: 0.8.1+3 tflite: 1.1.2 meta: 1.1.8

4.2. Model Development

The Image processing model was implemented using Convolutional Neural Network model through Keras and TensorFlow in python. The basic idea of CNN is to extract low level and high-level features from training dataset, assign random weights and biases and eventually fine-tune and learn these values. These weights and biases along with suitable activation functions are stored in the model which is generated by batchwise training over train dataset. The model can classify for an unseen input data, generally referred as test data because of these values.

We created our own CNN architecture using Keras. The model type was Sequential, which allows to build a model layer by layer. Four convolutional layers along with pooling, flattening and dense layer. Gradient, color, dimensions, edges are few low-level features that are extracted by convolutional layer. The addition of more layers enables the model to capture high level features and hence a proper understanding of each image is established. Convolutional layer is accompanied by Pooling layer. There are two types of Pooling operations – increment in dimension and decrement in dimensions. A holistic and effective understanding of image dataset is achieved because pooling layer has the capability to learn high level rotational features.[8]

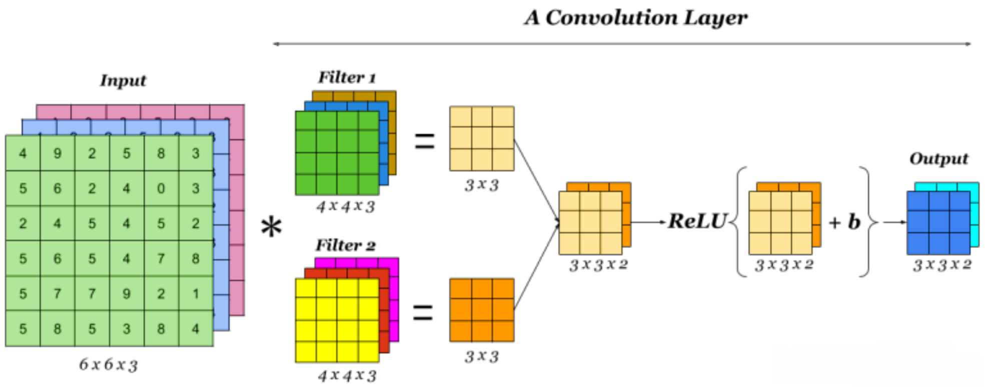


Figure – 4.1: Convolution Operation

The max pooling operations results in maximum value from the product of filter and area of image covered by the filter. In contrast to this, average pooling method returns the mean of all the values obtained by convolution operation in each portion of image. The above process ensures that model has successfully learnt all features of training dataset. Now, the feature map is flattened and feed to a regular Artificial Neural Network (ANN) to perform classification operation. The ANN employs backpropagation technique for each epoch. Eventually, the algorithm is capable enough to distinguish between features of input images and hence perform classification task for unseen data. The next process involved compilation of model using loss function, metrics, and optimizer. We selected ‘adam’ optimizer. The purpose of optimizer is to constantly adjust the learning rate throughout the training. The learning rate of model impacts the values of weights and biases, which eventually impacts the accuracy of model.

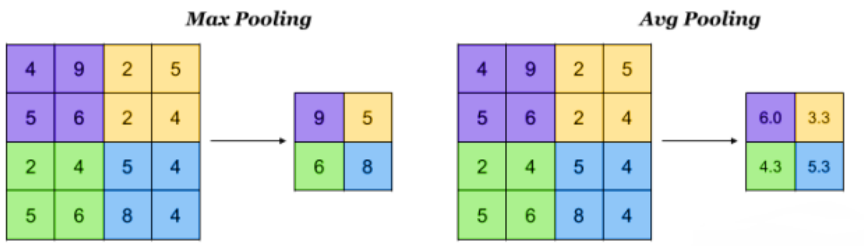


Figure – 4.2: Pooling Operation

Loss functions are widely used in machine learning to quantify model's performance. There are several loss functions such as L1 loss, L2 loss, tanmito loss, logistic, hinge, Chebyshev, cross entropy etc. We used 'categorical cross entropy'. It denotes a number which model aims to minimize by learning over training dataset [9]. We can understand loss function as optimization parameter.

The image dataset that has been used for training the CNN model consisted of 35 labels of 2670 images divided both in training image count of 1875 and validation image count of 795. It was created by subsetting datasets like Food 101 and ECUSTFD, with a special focus on incorporating Indian food items.

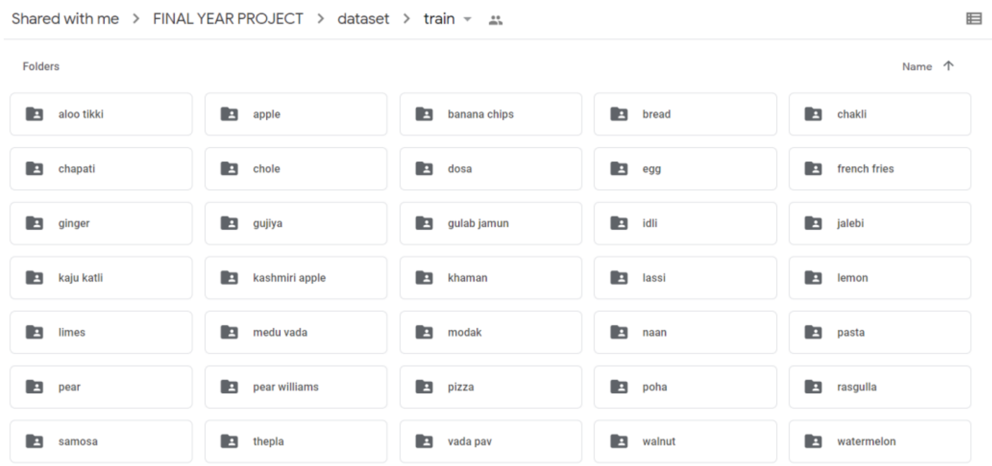


Figure – 4.3: Training Dataset

The dataset consisted of multiple images with different angle, colour, texture, etc. We also implemented data augmentation methods such as mirroring, rotation, random cropping, and color shifting to create a robust dataset for training. The model was trained over 90 epochs. Epochs stands for a counter, which denotes number of times algorithm will iterate over training dataset. It denotes number of times algorithm will learn the features and adjust its weights and biases by going through whole training dataset [10]. It determines the number of times the model will iterate over the complete dataset. We found that there was drastic increase in training accuracy and validation accuracy of model with increase in number of epochs, till a certain limit. After that negligible increase in accuracy is observed after increasing epochs.

We also created a database consisting of calories and nutrition values associated with various food items. The database was created through consultation with a nutritionist, gym trainer and credible online website viz. FatSecret API (https://www.fatsecret.com/) and NutritionX (https://www.nutritionix.com/food/).

Table – 4.3: A sample of nutrition dataset

Sr. No.	Food Name	Calories	Protein (gms)	Carbs (gms)	Fat (gms)
1	Apple	52	0.3	13.8	0.2
2	Chapati	70	3	15	0.4
3	Dabeli	278	7.23	42.35	9.4
4	Sev Puri	59	1.73	7.93	2.25
5	Vadapav	304	9.71	40.17	11.91
6	Samosa	308	4.67	32.21	17.86
7	Banana	105	1.1	23	0.3
8	Oranges	62	1.23	15.39	0.16
9	Modak	153	1.63	27.99	4.08
10	Rice	130	2.36	28.7	0.9
11	Dal	222	14	34	4.2
12	French Fries	323	3.4	43	15

We created an interface in form of Android and iOS mobile application that provided feature to read an image from gallery. The mobile application was written in Dart programming language using Flutter SDK. It consists of various screens – welcome screen, login, home page and profile. The mobile application leverages TensorFlow lite technology to store the pre-trained neural network model and predict for new data.

The home page contains menu widget that enables user to upload image from gallery, whenever a user inputs the food image model is loaded and prediction of food name is made. The application's source code contains a text file that stores calories, proteins and carbohydrate content associated with food item. These values are also fetched and shown as output to the user.

4.3. Ingredient based suggestive Prediction

One of the limitations of the application which was pointed out by the external examiner was, if a same food item which can be made of different ingredients the application should acknowledge that and estimate the calories of such food items. This issue was solved by updating the application and database of food items and their calorie information.

Whenever the model classifies food items that might have variations, our application will display calories of all the variations of that food item. For example, when a user inputs Gujiya, the application displays the calories associated with Gujiya as well as suggest the calories inside Mawa Gujiya. It isn’t possible for image processing to learn the features of Mawa Gujiya because the filling inside the shell makes the difference.

Chapter 5: RESULTS & DISCUSSIONS

The methodology of CNN is quite like that of traditional ML algorithms – input, feature extraction, feature map, and classification. However, unlike ML algorithms the features are learned automatically by CNN network. The initial convolutional layers are responsible for feature extraction. The size of filter provided to Keras functions decided the dimensions of filter matrix in each convolution layer. The filter performs convolution operation by dragging the kernel matrix over same size portion of image and calculates product of matrix multiplication. The advantage lies in the fact that unlike traditional ML approach, we don't have to hard-code the feature extractors. They are automatically assigned by the CNN algorithm We found out that features like corner, texture, shape, and edge were extracted by neural network. These features are automatically selected and updated. We also found that CNN used canny edge detection method to detect edges of food images.

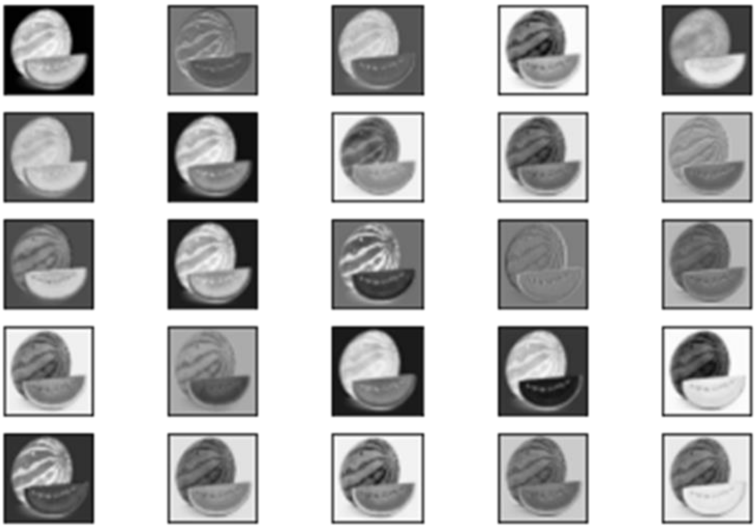


Figure – 5.1: A Sample map feature created by CNN

Although CNN algorithm has capability to tackle highly complex problems, but it becomes important that parameters are selected appropriately. We performed hit and trail approach to select appropriate values for model. Various combination of number of filters in each layer, number of epochs and different activation functions were used in hyper-tuning the model to achieve an accurate and effective model.

Training Accuracy and Training Loss are metrics associated with model's performance with training dataset i.e., seen data, whereas Validation accuracy and validation loss refers to performance of model w.r.t. unseen data. Hence, we give more priority to validation accuracy and validation loss while deciding value of any parameter. The results were recorded in below tables.

Table – 5.1: Performance of Different activation functions

Sr. No.	Activation Functions	No of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	Sigmoid	10	0.7543	0.8978	0.778	0.8035
2	ReLu	10	0.8425	0.4626	0.9618	0.2190
3	TanH	10	0.6071	0.5144	0.851	0.5865

Table – 5.2: Number of filters in Hidden Layers of Convolutional Neural Network

Sr. No.	No of filters in Hidden Layers of Neural Network				Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
	1	2	3	4				
1	8	8	8	8	0.5654	1.1468	0.6125	0.9834
2	16	16	16	16	0.5743	1.0226	0.6912	0.9541
3	8	16	32	64	0.7339	0.7960	0.8576	0.4392
4	32	64	128	256	0.8315	0.5117	0.9349	0.3176

Table – 5.3: Performance of model with different epochs size

Sr. No.	Number of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	10	0.8425	0.4626	0.9618	0.2190
2	20	0.9071	0.2780	0.9101	0.1210
3	30	0.9134	0.2202	0.9067	0.15494
4	40	0.9449	0.1888	0.9234	0.1204

Activation functions play significant role in neural networks by introducing the non-linearity property. It's absence simply equates to weighted sum of input and bias. They play pivotal role in decision making regarding activation of neurons in the architecture. Sigmoid, ReLu, TanH etc. are most widely used activation functions for CNN paradigm. We found that ReLu is suitable for this training dataset as it gave high validation accuracy and low validation loss. It also reduced computation time.

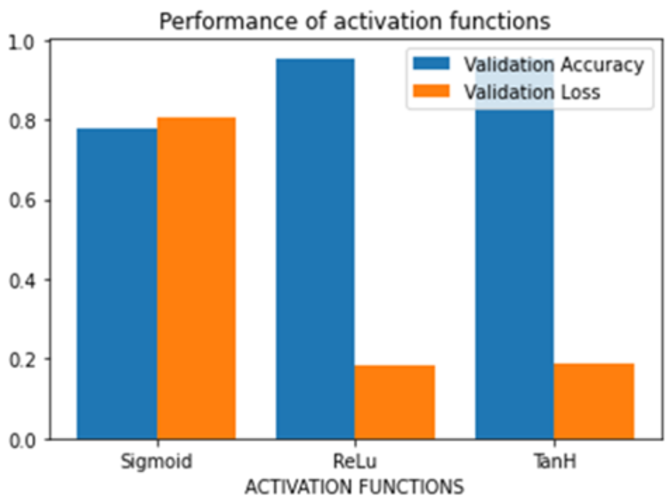


Figure – 5.2: Performance of activation functions

A CNN model with four convolution layer was created with various filter size. The reptation of same kernel over input image in a map of activations gives rise to feature map. We found that providing different number of filters in each hidden layers gives better result than same number of filters in every layer. We decided to use configuration of 32 – 64 – 128 – 256 for four hidden layers.

An accuracy metric provides a quantitative approach to measure the performance of model. The accuracy of a model is usually determined after the model parameters and is calculated in the form of a percentage. It denotes a comparative measure of model's predicted output against actual label. The work of loss function is to optimize the weights and algorithm. A model's loss is calculated using training and validation set and denotes how model is performing on these two sets. We aim for minimum loss model. We obtained an accuracy of 92.34% and loss of 12.04% with validation data.

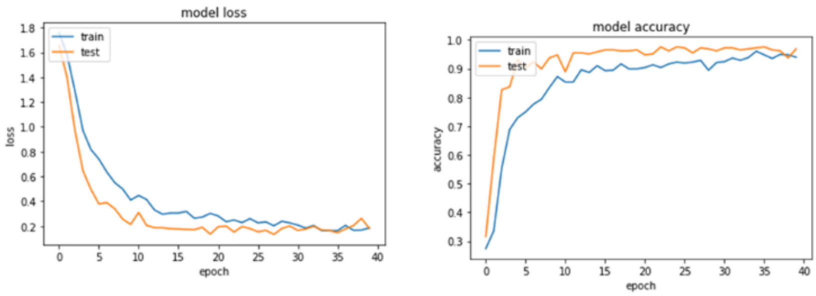


Figure – 5.3: Model loss and accuracy against epochs

We created an interface in form of Android and iOS mobile application that provided feature to read an image from gallery. The mobile application was written in Dart programming language using Flutter SDK. It consists of various screens – welcome screen, login, home page and profile. The mobile application leverages TensorFlow lite technology to store the pre-trained neural network model and predict for new data. The home page contains menu widget that enables user to upload image from gallery, whenever a user inputs the food image model is loaded and prediction of food name is made. The application's source code contains a text file that stores calories, proteins and carbohydrate content associated with food item. These values are also fetched and shown as output to the user.

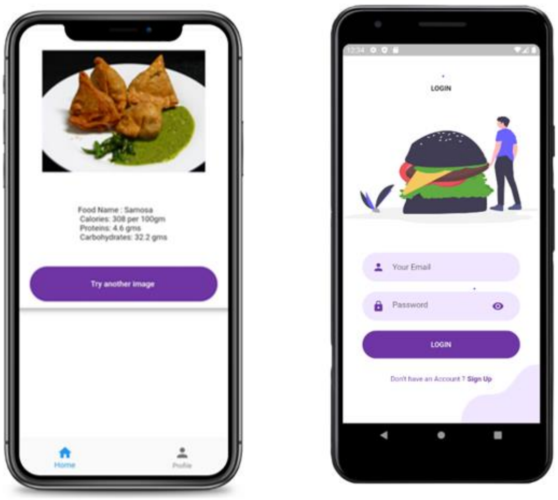


Figure – 5.4: User Interface

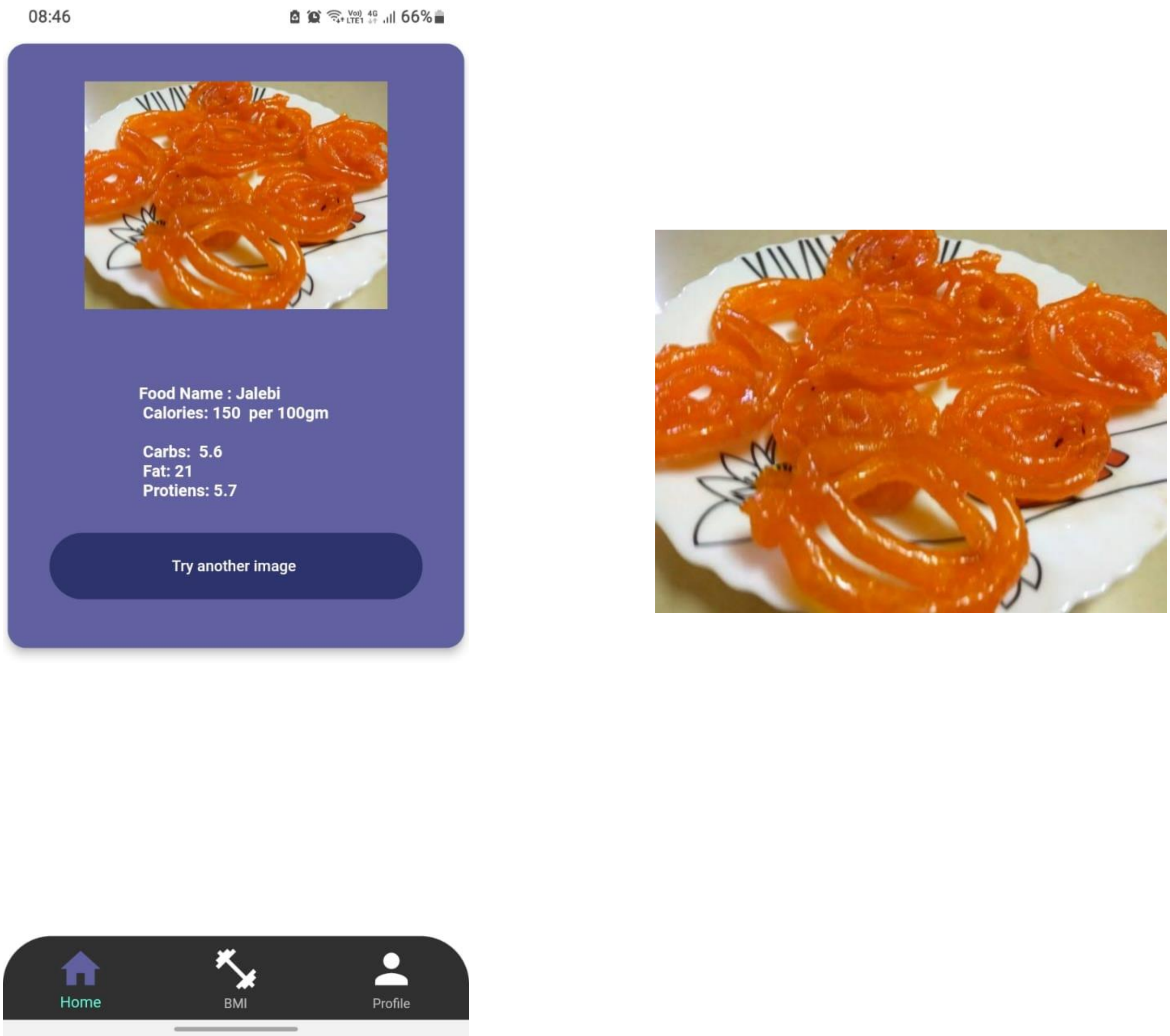


Figure 5.5: Testing of product in real environment

We have provided a real time image of Jalebi. The mobile application contains deployed neural network model which has capability to classify the food from trained label set. The output of classification is used as search query to extract nutritional content like calories, fat, protein and carbs associated with the food item. We can see that mobile app is able to identify the food item as Jalebi and appropriate calories are displayed as output.

Chapter 6: CONCLUSION & FUTURE SCOPE OF WORK

We have developed a prototype app which can classify 35 classes of different food items and display its class name along with its calories which is the main motive of our project. For the classifying the image, CNN algorithm is used, we have also compared different activation layers in CNN and found out that ReLu gave the best results. Along with this, comparisons were also made based on number of epochs, and we concluded that 40 epochs are the best option for keeping the computational efficiency as well as the achieving high training and validation accuracy.

Activation functions play pivotal role in decision making regarding activation of neurons in the architecture. Sigmoid, ReLu, TanH etc. are most widely used activation functions for CNN paradigm. We found that ReLu is suitable for this training dataset as it gave high validation accuracy and low validation loss. It also reduced computation time.

We found that providing different number of filters in each hidden layers gives better result than same number of filters in every layer. We decided to use configuration of 32 – 64 – 128 – 256 for four hidden layers. Performance of the model across different epochs size were taken in consideration. The more the epochs more the amount of time is required to process but as we increased the number of epochs the validation accuracy and validation loss numbers became much more considerable. For a training dataset of 17 labels, till 40 epochs we could observe a significant change, when we increased it to 50 there was no significant change and amount of time consumed was higher hence, our model was set for 40 epochs. Eventually it was set to 90 for 35 food labels.

In addition to displaying calories, the app also displays nutritional values such as carbohydrates, proteins, and fats of that food item. The app also has login and logout functionalities which makes user authentication must for using this app. The app also has 'My profile' where a user profile picture is displayed along with the 'My account' button and the logout button. Since this is the first version of our app, the user authentication is hard coded. Database connectivity is the next step in our path to convert this app from a prototype to a finished product. The primary focus now is to further increase the number of classes of items our app can classify and display their respective calorie value. Based on the plan to include these features in the future, we can say that the later versions of our applications will be useful to fitness enthusiasts, Dieticians and the general audience in their day-to-day life.

Measuring Calories and Nutrition from Food Images

Atul Kumar Tiwari
Department of E&TC
T.C.E.T.
Mumbai, India
atulktiwari310@gmail.com

Punit Savlesha
Department of E&TC
T.C.E.T.
Mumbai, India
p.savlesha@gmail.com

Yash Shetiya
Department of E&TC
T.C.E.T.
Mumbai, India
yashxlency0410@gmail.com

***Abstract*— The importance of calorie tracking is gaining wide acceptance not only in fitness sector but also in daily lives of people. It becomes important to ensure that process is simple, fast, and effective. The existing fitness applications require manual entry of food items to calculate calories intake. The advancement in field of computation and computer vision has led to rise of deep learning algorithms like CNN and platforms such as TensorFlow and Keras for effective and accurate image classification. In this paper, a fitness application that uses image processing to detect the food name and calculate the calories has been implemented. We have developed a mobile application that possess the ability to track calories on basis of food images using CNN trained neural network model.**

***Keywords*— Activation Function, Convolutional Neural Network (CNN), Deep Learning**

Introduction

Nutrition and calories tracking plays a significant role in health and fitness industry. The growing need of diet tracking in the world has evolved many applications that show you the calories intake a food product can provide, our application does it in a different manner. The application will take the image of the food product as the input from the user and process it to show the output i.e. the calories and nutrients in the food item. The mobile application has been developed in Dart programming and Flutter SDK. The core part of application is image processing model developed using Python's Keras and Tensorflow model. Convolutional Neural Network (CNN), a powerful deep learning algorithm for image processing and machine vision has been used to solve the image classification. The CNN model provides the capability to classify an input image into one of the pre-trained labels according to the features extracted by initial layers from input image.

LITERATURE SURVEY

Ch. Kavya, R. Priyadarsini and B. Madhavi in their paper titled "Calories and Nutrition Measurement from the Image of Food" designed a system to extract features like color, shape, size, and texture[1]. They used K-Nearest Neighbor (KNN) to tackle the image classification problem by training on food database. Hemalatha Reddy V., Kumari S., talked about a similar system in their paper- "Food Recognition and Calorie Measurement Using Image Processing and Machine Learning Techniques". They implemented with help of K-Means clustering [2]. I. Culjak, D. Abram's paper "A brief introduction to OpenCV," explained about practical usage of OpenCV library in image processing and machine vision . Intel introduced OpenCV library decades ago, it is an opensource library that helps users to deal with images and videos. The most peculiar thing about OpenCV is the fact that despite being an open-source tool, it is capable to giving results as good as commercial libraries developed by organizations [3].

The conventional approach of image classification has been pre-processing, feature extraction and classification by using classifiers like SVM, KNN, ANN, K-Means etc. These models have certain limitations as they are based on Machine Learning (ML) algorithms. The existing projects have managed to detect the food item and calories of naturally available food items such as fruits and vegetables, but these models fail in case of processed food. The main crux of food classification lies in combination of low-level and high-level features extracted by the neural network or ML algorithms. There are numerous fitness applications that are available in market but none of them offer a mechanism of calorie tracking through food image. The existing work have tackled the classification problem using Machine Learning approach. The ML approach suffers from limitation of feature extraction. The features to be extracted are generally hand-picked and hard coded by developer as per their understanding of domain. The process of feature extraction becomes cumbersome and relatively difficult as size of training dataset increases.

In contrast to Machine Learning, a stack of neural network layers is generally created by deep learning. The most important difference between deep learning and traditional machine learning is its performance as the scale of data increases. When the data is small, deep learning algorithms do not perform that well. This is because deep learning algorithms need a large amount of data to understand it perfectly [4].

Hokuto Kagaya's research titled "Image-based Calorie Content Estimation for Dietary Assessment," used multiple layers of CNN architecture to create a powerful model which was capable of detecting and classifying food items[5]. CNN showed tremendous success in handling complex and diverse training dataset.

Convolutional neural networks (CNN) are one of the most popular and powerful paradigms available in deep learning for solving image classification and other machine vision related problems. They make use of huge amount of low-level and high-level features to create a feature map and perform complex classification task [6]. Convolutional neural networks consist of two major layers, namely convolutional layers, and pooling layers. [7].

IMPLEMENTATION

The Image processing model was implemented using Convolutional Neural Network model through Keras and TensorFlow in python. The basic idea of CNN is to extract low level and high-level features from training dataset, assign random weights and biases and eventually fine-tune and learn these values. These weights and biases along with suitable activation functions are stored in the model which is generated by batchwise training over train dataset. The model can classify for an unseen input data, generally referred as test data because of these values.

We created our own CNN architecture using Keras. The model type was Sequential, which allows to build a model layer by layer. Four convolutional layers along with pooling, flattening and dense layer. Gradient, color, dimensions, edges are few low-level features that are extracted by convolutional layer. The addition of more layers enables the model to capture high level features and hence a proper understanding of each image is established. Convolutional layer is accompanied by Pooling layer. There are two types of Pooling operations – increment in dimension and decrement in dimensions. A holistic and effective understanding of image dataset is achieved because pooling layer has the capability to learn high level rotational features [8].

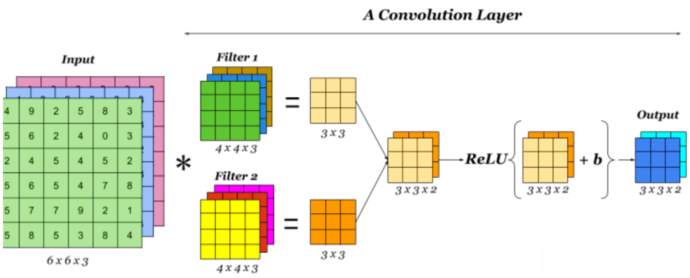


Fig. 1: Convolution Operation

The max pooling operations results in maximum value from the product of filter and area of image covered by the filter. In contrast to this, average pooling method returns the mean of all the values obtained by convolution operation in each portion of image. The above process ensures that model has successfully learnt all features of training dataset. Now, the feature map is flattened and feed to a regular Artificial

Neural Network (ANN) to perform classification operation. The ANN employs backpropagation technique for each epoch. Eventually, the algorithm is capable enough to distinguish between features of input images and hence perform classification task for unseen data. The next process involved compilation of model using loss function, metrics, and optimizer. We selected ‘adam’ optimizer. The purpose of optimizer is to constantly adjust the learning rate throughout the training. The learning rate of model impacts the values of weights and biases, which eventually impacts the accuracy of model.

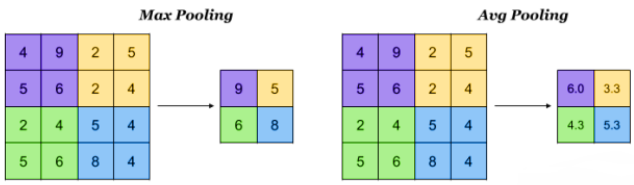


Fig. 2: Pooling Operation

Loss functions are widely used in machine learning to quantify model’s performance. There are several loss functions such as L1 loss, L2 loss, tanmito loss, logistic, hinge, Chebyshev, cross entropy etc. We used ‘categorical cross entropy’. It denotes a number which model aims to minimize by learning over training dataset [9]. We can understand loss function as optimization parameter.

The model was trained on 1276 data images of 17 classes. The dataset consisted of multiple images with different angle, color, texture, etc. We also implemented data augmentation methods such as mirroring, rotation, random cropping, and color shifting to create a robust dataset for training. The model was trained over 40 epochs. Epochs stands for a counter, which denotes number of times algorithm will iterate over training dataset. It denotes number of times algorithm will learn the features and adjust its weights and biases by going through whole training dataset [10]. It determines the number of times the model will iterate over the complete dataset. We found that there was drastic increase in training accuracy and validation accuracy of model with increase in number of epochs, till a certain limit. After that negligible increase in accuracy is observed after increasing epochs.

The model was tested on 565 images belonging to same classes and model was saved in h5 format. We created an interface in form of Android and iOS mobile application that provided feature to read an image from gallery. The mobile application was written in Dart programming language using Flutter SDK. It consists of various screens – welcome screen, login, home page and profile. The mobile application leverages TensorFlow lite technology to store the pre-trained neural network model and predict for new data. The home page contains menu widget that enables user to upload image from gallery, whenever a user inputs the food image model is loaded and prediction of food name is made. The application’s source code contains a text file that stores calories, proteins and carbohydrate content associated with food item. These values are also fetched and shown as output to the user.

RESULTS AND DISCUSSION

The methodology of CNN is quite like that of traditional ML algorithms – input, feature extraction, feature map, and classification. However, unlike ML algorithms the features are learned automatically by CNN network. The initial convolutional layers are responsible for feature extraction. The size of filter provided to Keras functions decided the dimensions of filter matrix in each convolution layer. The filter performs convolution operation by dragging the kernel matrix over same size portion of image and calculates product of matrix multiplication. The advantage lies in the fact that unlike traditional ML approach, we don’t have to hard-code the feature extractors. They are automatically assigned by the CNN algorithm We found out that features like corner, texture, shape, and edge were extracted by neural network. These features are automatically selected and updated. We also found that CNN used canny edge detection method to detect edges of food images.

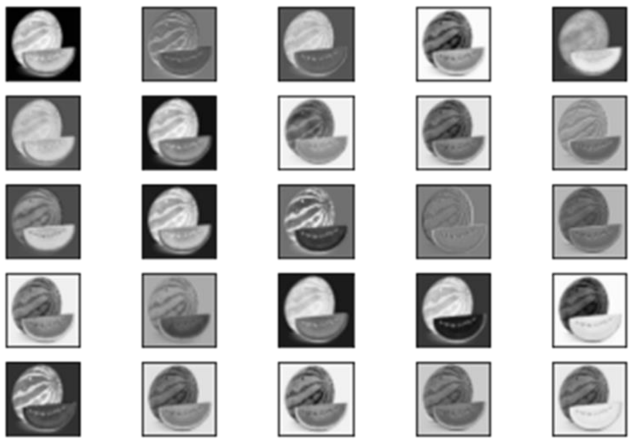


Fig. 3: A sample feature map created by CNN

Although CNN algorithm has capability to tackle highly complex problems, but it becomes important that parameters are selected appropriately. We performed hit and trail approach to select appropriate values for model. Various combination of number of filters in each layer, number of epochs and different activation functions were used in hyper-tuning the model in order to achieve an accurate and effective model.

Training Accuracy and Training Loss are metrics associated with model's performance with training dataset i.e., seen data, whereas Validation accuracy and validation loss refers to performance of model w.r.t. unseen data. Hence, we give more priority to validation accuracy and validation loss while deciding value of any parameter. The results were recorded in below tables.

Table 1: Performance of different activation functions

Sr. No.	Activation Functions	No of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	Sigmoid	10	0.7543	0.8978	0.778	0.8035
2	ReLu	10	0.8425	0.4626	0.9618	0.2190
3	TanH	10	0.6071	0.5144	0.851	0.5865

Table 2: Number of filters in Hidden Layers of Convolutional Neural Network

Sr. No.	No of filters in Hidden Layers of Neural Network				Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
	1	2	3	4				
1	8	8	8	8	0.5654	1.1468	0.6125	0.9834
2	16	16	16	16	0.5743	1.0226	0.6912	0.9541
3	8	16	32	64	0.7339	0.7960	0.8576	0.4392
4	32	64	128	256	0.8315	0.5117	0.9349	0.3176

Table 3: Performance of model with different epochs size

Sr. No.	Number of Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	10	0.8425	0.4626	0.9618	0.2190
2	20	0.9071	0.2780	0.9101	0.1210
3	30	0.9134	0.2202	0.9067	0.15494
4	40	0.9449	0.1888	0.9234	0.1204

Activation functions play significant role in neural networks by introducing the non-linearity property. It's absence simply equates to weighted sum of input and bias. They play pivotal role in decision making regarding activation of neurons in the architecture. Sigmoid, ReLu, TanH etc. are most widely used activation functions for CNN paradigm. We found that ReLu is suitable for this training dataset as it gave high validation accuracy and low validation loss. It also reduced computation time.

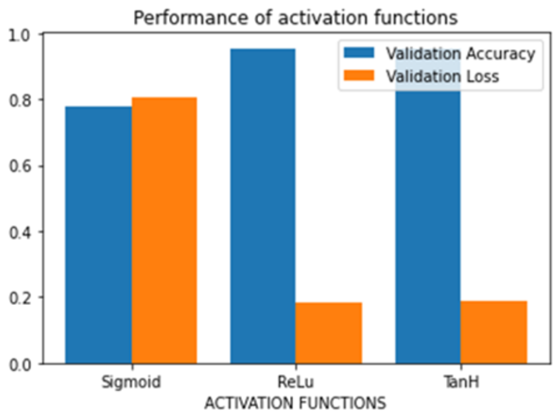


Fig. 4: Performance of activation functions

A CNN model with four convolution layer was created with various filter size. The reptation of same kernel over input image in a map of activations gives rise to feature map. We found that providing different number of filters in each hidden layers gives better result than same number of filters in every layer. We decided to use configuration of 32 – 64 – 128 – 256 for four hidden layers.

An accuracy metric provides a quantitative approach to measure the performance of model. It denotes the ratio of sum of true positive and true negative to sum of all predictions. Machine Learning makes use of variety of loss functions to learn optimum and effective weights and biases for particular dataset. It gives a quantitative metric that also denotes the performance of trained model on test/validation dataset. We aim for minimum loss model. We obtained an accuracy of 92.34% and loss of 12.04% with validation data.

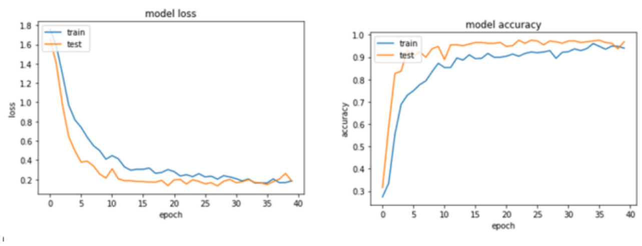


Fig. 5: Model loss and accuracy against epochs

We created an interface in form of Android and iOS mobile application that provided feature to read an image from gallery. The mobile application was written in Dart programming language using Flutter SDK. It consists of various screens – welcome screen, login, home page and profile. The mobile application leverages TensorFlow lite technology to store the pre-trained neural network model and predict for new data. The home page contains menu widget that enables user to upload image from gallery, whenever a user inputs the food image model is loaded and prediction of food name is made. The application's source code contains a text file that stores calories, proteins and carbohydrate content associated with food item. These values are also fetched and shown as output to the user.

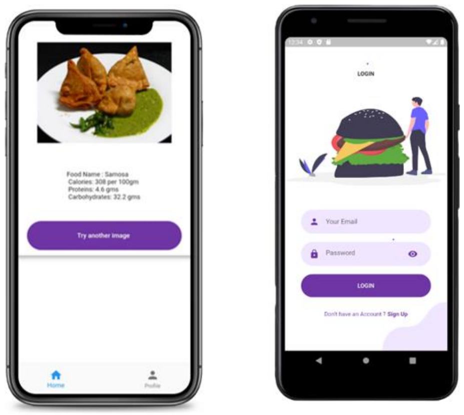


Fig. 6: User Interface of mobile app

CONCLUSION

We have developed an app which can classify 17 classes of different food items and display its class name along with its calories. The image processing model was developed using CNN algorithm. We also performed a comparative study of various activation functions, impact of filter size. Along with these, comparisons were also made based on number of epochs, and we concluded that 40 epochs are the best option for keeping the computational efficiency as well as the achieving high training and validation accuracy. In addition to displaying calories, the app also displays nutritional values such as carbohydrates, proteins, and fats of that food item.

The application has been developed on Model View Template (MVC) pattern of software engineering hence separation of concerns can be achieved. The application can be further integrated with backend database and APIs for development of industry ready product. We received suggestions for development of internal features like macro-based nutrient tracking, water intake tracker and workout tracker to enhance the application into a holistic fitness related product. We intend to expand the number of food labels on which neural network has been trained in future.

Acknowledgment

It would be unfair if we do not acknowledge the help and support given by Professors, students, friends etc. We want to express our sincere gratitude towards Ms. Purnima Chandrasekhar, our project guide for her valuable feedback and support at every stage. We want to thank Electronic and Telecommunication department of T.C.E.T., Mumbai for giving us the opportunity to work on this project.

References

[1] Ch. Kavya, R. Priyadarsini and B. Madhavi, "Calories and Nutrition Measurement from the Image of Food" International Journal of Advance Research, Ideas and Innovations in Technology, https://www.ijariit.com/manuscripts/v3i3/V3I3-1205.pdf

[2] Hemalatha Reddy V., Kumari S., Muralidharan V., Gigoo K., Thakare B.S. (2020) Literature Survey—Food Recognition and Calorie Measurement Using Image Processing and Machine Learning Techniques. In: Kumar A., Mozar S. (eds) ICCCE 2019. Lecture Notes in Electrical Engineering, vol 570. Springer, Singapore

[3] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012, pp. 1725-1730

[4] P. Pouladzadeh, S. Shirmohammadi and R. Al-Maghrabi, "Measuring Calorie and Nutrition From Food Image," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 8, pp. 1947-1956, Aug. 2014, doi: 10.1109/TIM.2014.2303533.

[5] Hokuto Kagaya, Kiyoharu Aizawa, Makoto Ogawa ," Image-based Calorie Content Estimation for Dietary Assessment," "MM '14: Proceedings of the 22nd ACM international conference on Multimedia November 2014," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 171-175, doi: 10.1109/ICIP.2018.8451422.

[6] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu and S. Yang, "Food Image Recognition with Convolutional Neural Networks," 2015 IEEE 12th and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, 2015, pp. 690-693, doi: 10.1109/UIC-ATCScalCom-CBDCom-IoP.2015.139.

[7] Ms. Ankita A. Podutwar , Prof. Pragati D. Pawar , Prof. Abhijeet V. Shinde," A Food Recognition System for Calorie Measurement", International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 1, January 2017, ISSN (Online) 2278-1021 ISSN (Print) 2319 5940.

[8] K Scott Mader, "Food Images (Food 101)', Available: https://www.www.kaggle.com/kmader/food41

[9] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[10] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

Appendix A

Abbreviation and symbols

1. CNN: Convolutional Neural Network
2. ML: Machine Learning
3. DL: Deep Learning
4. SVM: Support Vector Machine
5. KNN: K Nearest Neighbor
6. ANN: Artificial Neural Network
7. SDK: Software Development Kit

Plagiarism Report

ORIGINALITY REPORT

15%	14%	7%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	docplayer.net Internet Source	1%
2	www.coursehero.com Internet Source	1%
3	link.springer.com Internet Source	1%

Appendix B

Definitions

1. Model: A model in data science refers to a file (saved or trained) on a given dataset using pre-defined algorithms to understand and learn certain type of pattern.

2. Accuracy: It's a measurement of correct predictions that are made by model. It is defined as ratio of sum of True Positive (TP) and True Negative (TN) to sum of True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

   Accuracy = $\frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}}$

3. Loss: It a number denoting measure of bad predictions. An ideal model should have zero loss. It refers to penalty for incorrect predictions.

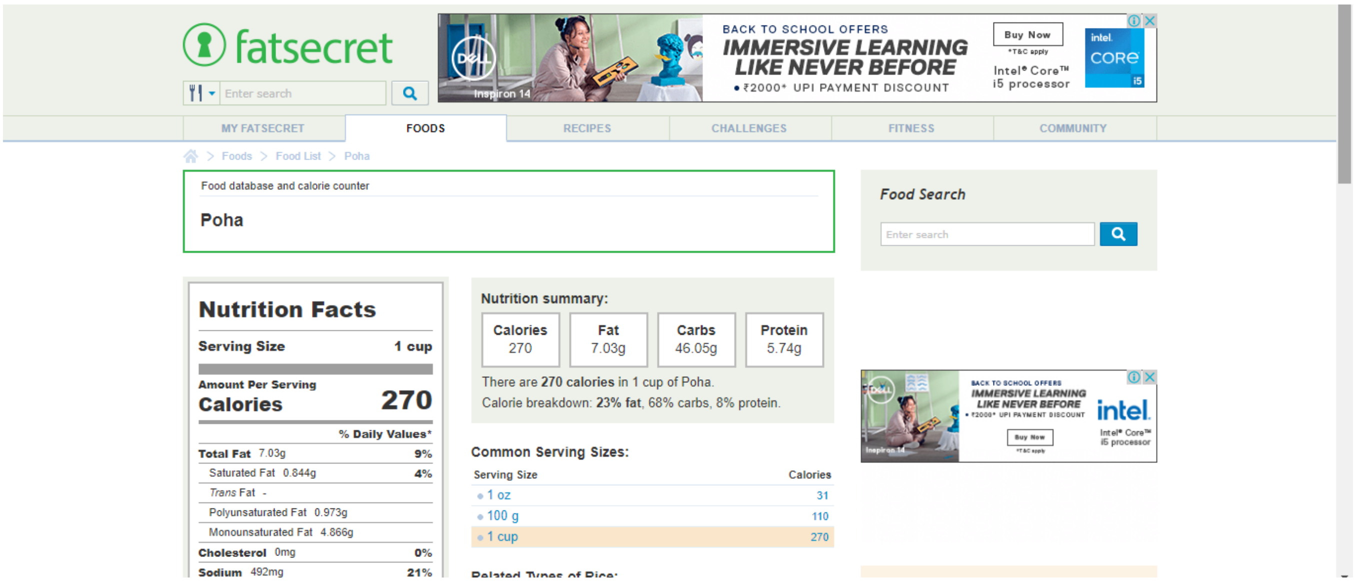
4. Training Accuracy: The value of accuracy calculated for model over seen dataset i.e., train dataset is called training accuracy.

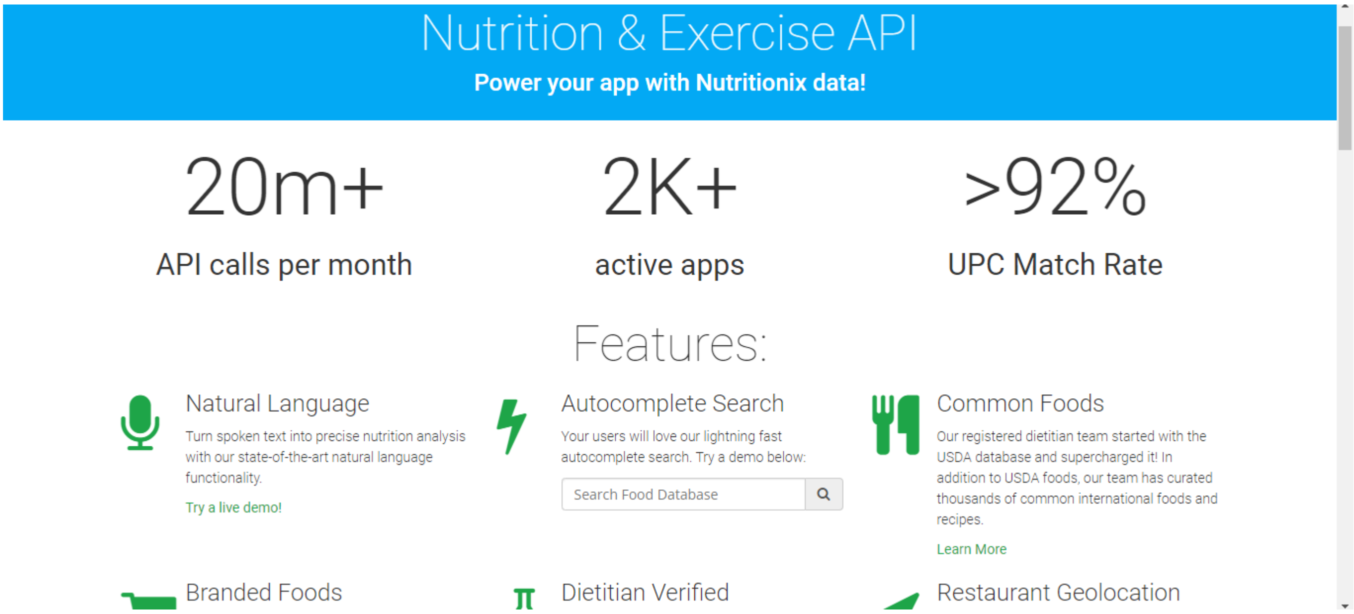
5. Validation Accuracy: The value of accuracy calculated for model over unseen dataset i.e., validation dataset is called validation accuracy.

6. Kernel/Filter: Kernel refers to a matrix which is used to convolute (dot product) with patch of an image (local receptive field) of similar dimension.

7. Epochs: The hyperparameter that decides the number of times an algorithm will iterate and learn on a particular dataset is called as epochs.

Nutrition Database Reference





REFERENCES

[1] Ch. Kavya, R. Priyadarsini and B. Madhavi, “Calories and Nutrition Measurement from the Image of Food” International Journal of Advance Research, Ideas and Innovations in Technology, Available on: https://www.ijariit.com/manuscripts/v3i3/V3I3-1205.pdf

[2] Hemalatha Reddy V., Kumari S., Muralidharan V., Gigoo K., Thakare B.S. (2020) Literature Survey—Food Recognition and Calorie Measurement Using Image Processing and Machine Learning Techniques. In: Kumar A., Mozar S. (eds) ICCCE 2019. Lecture Notes in Electrical Engineering, vol 570. Springer, Singapore

[3] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012, pp. 1725-1730

[4] P. Pouladzadeh, S. Shirmohammadi and R. Al-Maghrabi, "Measuring Calorie and Nutrition From Food Image," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 8, pp. 1947-1956, Aug. 2014, doi: 10.1109/TIM.2014.2303533.

[5] Hokuto Kagaya, Kiyoharu Aizawa, Makoto Ogawa ," Image-based Calorie Content Estimation for Dietary Assessment," "MM '14: Proceedings of the 22nd ACM international conference on Multimedia November 2014," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 171-175, doi: 10.1109/ICIP.2018.8451422.

[6] G. S. Jayalakshmi and V. S. Kumar, "Performance analysis of Convolutional Neural Network (CNN) based Cancerous Skin Lesion Detection System," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862143.

[7] S. Mhalagi, “The quest of higher Accuracy for CNN models,” Towards Data Science, 22-May-2019. [Online]. Available: https://towardsdatascience.com/the-quest-of-higher-accuracy-for-cnn-models-42df5d731faf. [Accessed: 14-Oct-2021].

[8] S. Saha, “A comprehensive guide to convolutional neural networks — the ELI5 way,” Towards Data Science, 15-Dec-2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 14-Oct-2021].

[9] J. Brownlee, “A gentle introduction to the rectified linear unit (ReLU),” Machinelearningmastery.com, 08-Jan-2019. [Online]. Available:

https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. [Accessed: 14-Oct-2021].

[10] D. Scherer, A. Muller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in Proc. of the Intl.Conf. on Artificial Neural Networks, 2010, pp. 92–101