Computer Vision Based Food Recognition with Nutrition Analysis

SHALINI K GOWDA, G S SUCHITHA, LAKSHMI S, DARSHAN GOWDA T S

Department of Computer Science and Engineering

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Udayapura, Kanakapura Road, Bangalore-560082

Email:shalinikg32@gmail.com

ABSTRACT

Tracking food intake can give insights into eating habits; hence it is useful to confront the rising public health threat of obesity and overweight. Because of high pace of modern life, people usually do not have time for manually recording their daily meals. Therefore, it is necessary to have an application which can accelerate the tracking process by automatically detecting foods from input image, collecting nutrition facts and recording. Deep learning methods have already proved promising in the food detection challenge. Among these methods, Tensor Flow Object Detection API is an open-source framework, consists of many pre-trained models that can be used to train and deploy a custom object detection model. Furthermore, there are some pre-trained models uses the Mobile Net which is an efficient con- volitional neural network for mobile vision applications

INTRODUCTION

Overweight and obesity are seen to increase sharply, according to research of World Health Organization (WHO) (World Health Organization, 2020), there were more than 1.9 billion adults, 340 million children and adolescents aged 5-19 overweight in 2016 and this number was tripled in comparison to 1975. The condition that body weight is greater than normal or healthy level for a certain height is considered as overweight and obesity. Overweight and especially obesity can lead to many other serious medical conditions such as high blood pressure, heart disease, and metabolic syndrome. Furthermore, it may cause negative effects on mental and emotional of obesity people. In theory, beside extra muscle or bone, the increase in dimension and amount of fat cell in body are reasons for overweight. There are many factors that lead to condition which lose control of fat cell in the body such as stress, lack of exercise lifestyles, insomnia or sleeping difficulties and eating behaviors which is considered as dominant reason (Jiang, Qui, Liu, Huang & Lin, 2020).

OBJECTIVE

Features learned by deep Convolutional Neural Networks (CNNs) have been recognized to be more robust and expressive than hand-crafted ones. They have been successfully used in different computer vision tasks such as object detection, pattern recognition and image understanding. Given a CNN architecture and a training procedure, the efficacy of the learned features depends on the domain-representativeness of the training examples. In this paper we investigate the use of CNN-based features for the purpose of food recognition and retrieval. Helping people lose weight, maintain a healthy weight, and prevent chronic disease by improving dietary habits requires providing education about food and nutrition, assuring access to healthier food options, and promoting the desire and ability to become physically active.

• Improves taste and food quality.

• Improves digestibility.

• Adds variety in the diet.

• Increases consumption of food.

• Increases availability of nutrients.

• Increases antioxidant value.

• Cooking concentrates nutrients.

• To promote the physical and mental growth and development of human beings.

MOTIVATION

Consumer perception of food products is a very complex phenomenon that is influenced by a wide range of characteristics. The major motivation for food science and nutrition should be sensual features, cost/price balance, and consumer health (sufficient/balanced nutrition). However, there are important differences between theory and reality. Food choice is a complex process influenced by a number of factors related to the product, the consumer, and the consumption context. The role of the consumer in determining the market success of a product is of maximum relevance. Consumer perceptions and preferences are in motion and in change. Understanding and analyzing consumers' motivation factors, perception and preferences are important both food industry and also governments. In this study, some of these factors were discussed and aimed to identify them with reasons.

Applications

• Nutrition tracking Apps
• Food Labeling
• Food Safety Inspection
• Personalized Nutrition
• Healthcare

METHODOLOGY AND PROPOSED SYSTEM

Proposed System The proposed method will use CNN based Computer vision algorithm to recognize the Food images. Then nutrients quantity will be analyzed with ML algorithm. Here are the following steps: 1. CV Model will be built (training) by different images of food with labels 2. CV Model will be validated with the testing data 3. ML model will be built for nutrients quantity analysis 4. Final model will be deployed for the actual use.

Advantages

• Efficiency: These systems can quickly process a large number of food images and provide nutritional analysis.
• Convenience: Users can simply take a photo of their food and receive an estimate of its nutritional value, making it easier to track their dietary intake.
• Personalization: These systems can be personalized to the user's dietary needs and preferences, allowing for more targeted and effective nutritional tracking.

• Health benefits: By tracking their dietary intake, users can make more informed choices and improve their overall health and wellness.

Disadvantages:

• Limited accuracy: While computer vision-based systems can be very accurate, they are not perfect and may still make mistakes in recognizing food items and estimating their nutritional value.
• Limited scope: These systems are only able to analyze food items that have been previously trained on and may not be able to recognize all types of food.
• Cost: Developing and implementing a computer vision-based system can be costly, especially for smaller companies or individuals.
• Privacy concerns: Users may be uncomfortable with sharing their dietary information with a third-party system, raising privacy concerns.
• User error: Users may take photos incorrectly, or the lighting or angle may be wrong, leading to inaccurate recognition and nutritional analysis.

Problem Statement

• The mobile cloud computing invites an unprecedented opportunity to discover early predictors and novel biomarkers to support and enable smart care decision making in connection with health scenarios, including that of dietary assessment.
• There are thousands of mobile cloud health software (e.g., mobile health Apps available for iPhone, iPad, and Android) and many mobile health hardware options (e.g., activity tracker, wireless heart rate monitors).
• While these Apps have features to track food intake, exercise, and save data in the cloud, the user still has to manually enter everything they ate. Several apps have an improved level of automation.
• It is plausible to train large and complex CNN models for accurate recognition, which surpassed most of the methods adopted using hand-crafted features.
• The machine-learned features with deep learning based method, rather than the hand engineered features, to achieve a much higher accuracy

Proposed Methodology

Conventional dietary assessment programs require maintaining a daily record of consumed food, manual identification of its contents, and an estimation of its volume. However, these methods pose a challenge for elders especially when it involves an accurate estimation of the amount and time of the food intake. For these reasons, the need for a sophisticated system to automatically carry out all the tasks of food intake, such as detection, food type classification, and volume estimation, has been the main focus in many recent research efforts. The advancements in machine learning and computer vision based applications have paved the way for more robust dietary assessment tools. The general purpose of vision-based methods is to recognize the food, estimate its volume, and assess the related nutrient information. With the development of deep learning algorithms, food detection and recognition accuracy have been drastically improved. However, the performance and effectiveness of such solutions depend on several factors. First, optimal classification accuracy can be attained by training the image classifier with a large number of food images for each class. Additionally, a proper segmentation approach must be chosen and implemented to identify all food segments within a single image. Computer Vision Based Food Recognition with Nutrition Analysis Dept. Of CSE, DSATM 2022-23 Page 17 In addition to the extraction of these segments from the image back- ground. Finally, after identifying the food, volume estimation of each food item must take place to assess the corresponding weight and nutrient information. A typical procedure of vision-based dietary assessment system is shown in Fig. 1

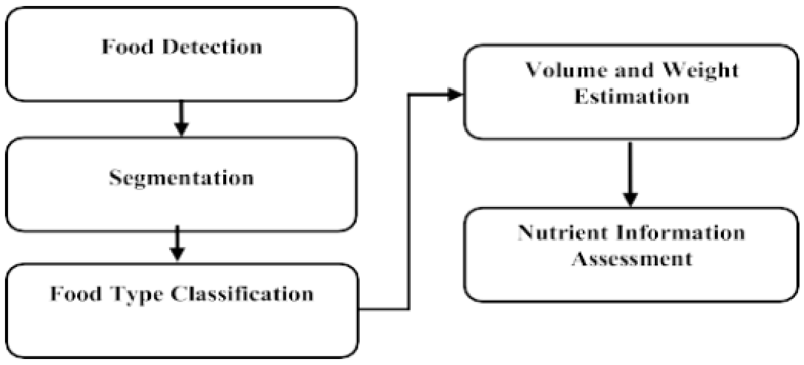


Fig.1: A typical procedure of vision-based dietary assessment system

Food Image Classification:

A basic automatic dietary assessment system is required to identify and recognize the food contained in a meal. The image classification, a machine learning technique, is used to identify a set of unknown objects that belong to a subset (class), which has been learned by the classifier in the training phase. In this step, food images are used as input data to train the classifier. An ideal classifier must be able to recognize any food type that has been included in the learning process. Practically, multiple variations exist in digital images, including rotation, distortion, color distribution, lighting conditions, and so forth, which may affect the overall accuracy. The training process itself is a tedious task that consumes a considerable amount of time to reach its intended accuracy goals. The classifier accuracy is affected mainly by the quantity and quality of images used in the training process as well as the proper selection of visual features. The extraction of image features used in the learning process splits a typical image classifier implementation into two strategies: traditional classifiers with handcrafted features and deep learning approaches as shown in Fig. 2

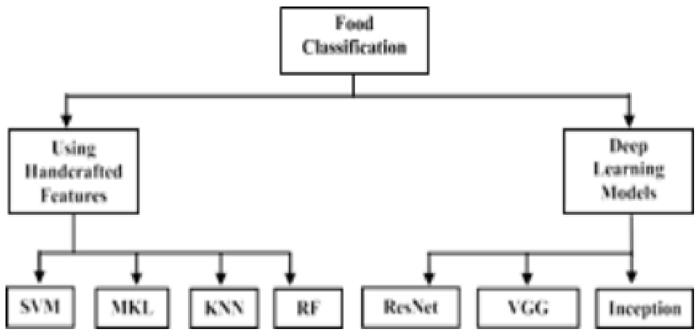


Fig.2: Common Classification Approaches For Food Images

Modules Identified

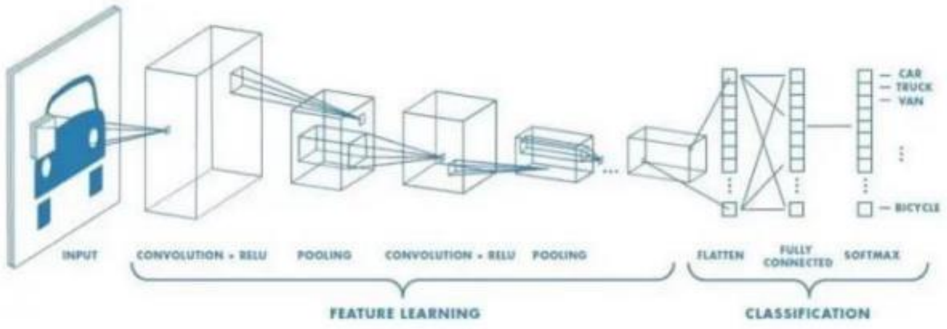


Figure 4.1: Architecture Diagram

Convolutional Neural Network Convolutional Neural Network (CNN) is a neural network with many layers, each network has aspeci- fied architecture which depend on input, output, and design of developer. In general, CNNtries to extract complex features of input data through each layer and give result as the latest layerwhich is differently designed for reaching defined target (Cioca, et al., 2018). Computer Vision Based Food Recognition with Nutrition Analysis.

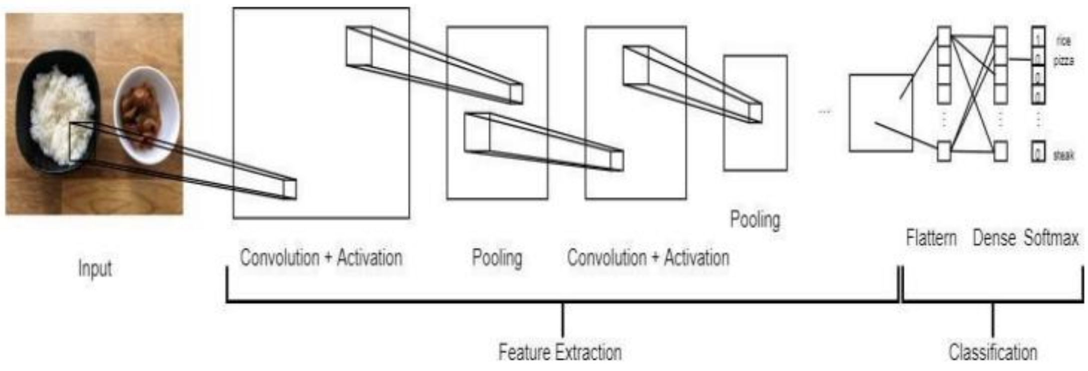


Figure 4 : An example of CNN architecture (Kiourt, Pavlidis & Markantonatou, 2020)

Figure 4 shows a simple CNN which has 3 parts: input, feature extraction and classification. The input of CNN is usually an image or a static frame that extracted from video. During preprocessing, the image is transformed to square by cropped or squeezed techniques. After that, it is fed into the network as a tensor which is a multidimensional array (Kiourt, et al., 2020). For example, an image with 32 x 32 pixels and 3 colour channels RGB will present as a tensor in form (32, 32, 3). Each element in the array keeps a value in range from 0 to 255, then taking corresponding positions in 3 colour channels such as (1, 1, 1), (1, 1, 2) and (1, 1, 3) can represent the correct colour at that point in the image. It depends on batch size then the network receives (32, 32, 32, 3) in case a batch of 32 images or (1, 32, 32, 3) if there is only 1 image inputted (Koirala, 2018). In the mentioned CNN architecture, feature extraction level has two combined convolution and activation layers beside 2 pooling layers, however, in real-world architecture, the feature extraction level is usually a set of many blocks constituted of convolutional and pooling layers. Convolution layer is a place that part of the input image is convolved with hidden layer which is also called as filters designed by developer to shaping features of image. There are 2 techniques that usually go along with convolution which are padding and stride convolution. Padding is a technique of adding extra values around input array to avoid losing information from the edge of image. Because convolution always results a smaller dimension array, consequently, there is an imbalance information among parts of image after several times if padding is not used.

In addition, padding is also useful to keep the output matrix as the same size as the input. Zeropadding isthe most common method by adding number of zeroesto each side input array (Jiang, et al., 2020). Conversely, if stride is defined, during convolution, matrix would move by defined value instead of one stride at a time. In figure , convolution layer is combined with activation function, which are mathematical equations that determine output of layer or neural network. There are many kinds of activation functions such as Sigmoid, Hyperbolic Tangent (TanH). The most common activation in CNN is Rectified Linear Unit (ReLU) which always returns greater value between 0 and input value. One another function is Softmax which is often used as the last activation function in a neural network to normalize result to a probability distribution over predicted output classes . The rest layer of one CNN block is pooling layer, which is used to reduce the tensor dimensions in height and width. There are average pooling and max pooling, the latter is used with higher frequency in real-world projects. If the pooling size is 2x2 and the tensor size is 4x4, the output will be 2x2, the pooling window checks 4 elements in matrix respectively and results the highest value or average value corresponding max pooling or average pooling (Kiourt, et al., 2020). The explanation can be viewed in figure 4.3 which is showed on the next page.

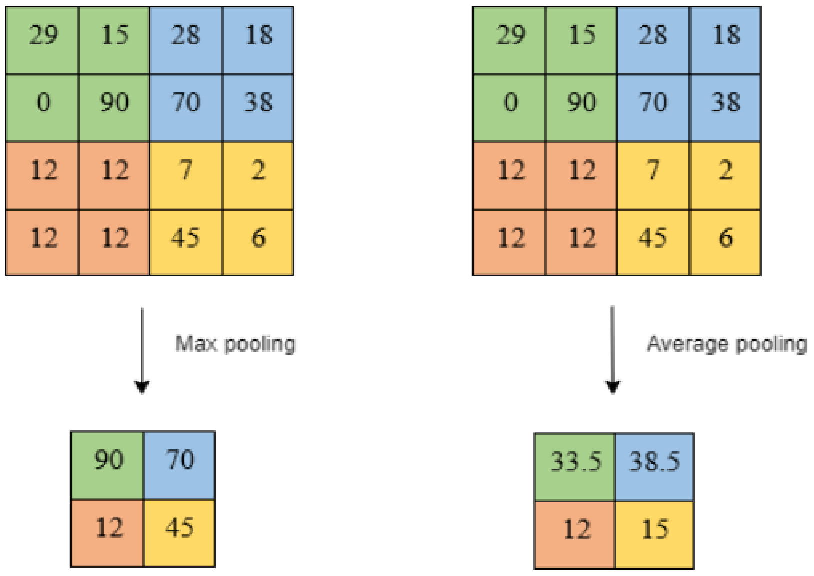


Figure 5: Max-pooling and average pooling (Kiourt, et al., 2020)

The classification level of the network includes a flattening layer, a denser layer and a prediction layer which is one of activation functions. The purpose of flattening layer is to transform multidimensional array resulted from last convolutional layer to a one-dimension array (Kiourt, et al., 2020).After that, dense layer which is also called fully connected layer will connect each neuron of itself to each output node of flattening layer corresponding with class label. Finally, the output of activation function will give prediction in form of a one-hotencod which represents true - “1” for matched classes and false - “0” for the others. (Koirala, 2018).

DEEP LEARNING FRAMEWORKS:

Deep Learning framework offer libraries, tools and application to developers and scientists for easily designing, training and validating complex deep neural networks with a high-level programming inter- face. The best part about Deep Learning framework is that user is not required to handle mathematics and algorithms. Further-more, most of frameworks are optimized for using GPU power to provide high performance during training model. This section briefly introduces 3 most common Deep Learning frameworks which are TensorFlow, PyTorch and Keras (Goyal, 2021). TensorFlow is an open-source framework for numerical computation using data flow graphs. There are many nodes in the graph represent mathematical operations, and the tensors are represented at the graph edges. The framework is written in C++ programming language and provides more than two hundred operations which can handle multidimension array, mathematical operations as well as control flow. It is developed by a machine learning team at Google and supports GPU-accelerated, TPU, CPU as well as mobile devices. Because of flexible architecture and ability to apply in many kinds of devices, Ten- sorFlow is considered to use in both researching tasks and production development. However, it uses low-level API, which could prevent inexperience users from using. Recently, TensorFlow 2 is released with Keras as its high-level API which helps to overcome the drawback. (Abadi, et al., 2016).

DEEP LEARNING METHODS FOR FOOD DETECTION

Nowadays there are many methods to build a new Deep Learning model with specific target. These methods are based on 3 strategies: building and developing new architecture for solving a problem; applied transfer learning and fine tuning on pre-trained models with new domain; and using deep learn- ing platforms which provide deep learning models pre-trained with large dataset, hence it can be used with many purposes. This section discusses 3 mentioned approaches in context of food detection and provide a general view about available food image datasets.

Experimental log- training epoch,accuracy,loss,val_accuracy,val_loss

```
0,0.0601051039993763,4.926839828491211,0.22397813200950623,4.198718070983887
1,0.23298650979995728,3.965515375137329,0.4308856129646301,2.9495327472686768
2,0.3572503626346588,3.2320077419281006,0.5282398462295532,2.3641769886016846
3,0.43704280257225037,2.7976932525634766,0.5910171270370483,2.04571795463562
4,0.49566906690597534,2.5092012882232666,0.6263070106506348,1.8618606328964233
5,0.5338817238807678,2.3004257678985596,0.6581907272338867,1.7008998394012451
6,0.5663770437240601,2.1355504989624023,0.6774793863296509,1.600325584411621
7,0.5902369022369385,2.00510311126709,0.6943916082382202,1.507713794708252
8,0.6159321069717407,1.8819572925567627,0.7046498656272888,1.44483304 02374268
9,0.6324900388717651,1.7829915285110474,0.713244616985321,1.3909118175506592
10,0.6492724418640137,1.7050282955169678,0.7213640809059143,1.3454564809799194
11,0.6632423996925354,1.6275471448898315,0.7312262654304504,1.2903163433074951
12,0.6748620271682739,1.5584192276000977,0.7333254218101501,1.2671607732772827
13,0.6858742237091064,1.4968417882919312,0.7337610721588135,1.2473195791244507
14,0.6998838186264038,1.4320111274719238,0.7376821637153625,1.2323052883148193
15,0.7084664702415466,1.3798527717590332,0.7530497312545776,1.1677051782608032
```

```
16,0.720020055770874,1.3260307312011719,0.7538815140724182,1.1484973430633545
17,0.7295270562171936,1.282737374305725,0.7528120875358582,1.1518810987472534
18,0.7394037246704102,1.2341399192810059,0.7598621845245361,1.1234118938446045
19,0.7497953176498413,1.1860905885696411,0.7604166865348816,1.1076292991638184
20,0.7559220194816589,1.1514289379119873,0.7627138495445251,1.1099566221237183
21,0.7626825571060181,1.116029977798462,0.7616444826126099,1.1049764156341553
22,0.7694166302680969,1.084214210510254,0.764931857585907,1.083146095275879
23,0.7790820598602295,1.0435117483139038,0.7697639465332031,1.064534306526184
24,0.7852483987808228,1.010107159614563,0.7667142152786255,1.0691211223602295
25,0.7939102649688721,0.9786399602890015,0.7642585635185242,1.085853934288025
26,0.7999049425125122,0.9468371868133545,0.7692094445228577,1.062933325767517
27,0.8065333962440491,0.9160622954368591,0.7721007466316223,1.0597246885299683
28,0.8132014870643616,0.8867677450180054,0.7699223756790161,1.0637348890304565
29,0.8187603950500488,0.8603400588035583,0.7608919739723206,1.0951259136199951
```

Displaying of Output

The output is displayed on the monitor of the device. The output is the drawing made by tracking the food images which is taken as input. This involves the above steps as described. The user can mainly upload the food images as input then it recognizes based on the persons height, weight and gender. Then it gives the results as output in the form of Nutrition Analysis.

SYSTEM DESIGN

Food Image Dataset

Regardless of approaches, CNN always requires sufficient data, even large datasets to build an efficient model and avoid over fitting which usually causes by small dataset. As a result, the model only works with that small data and it has not clue when interact with practical data. Though the development of AI technologies, there are limitations on the datasets about nutrition contents (Ciocco, et al., 2016). Table 1 represents some of the most popular datasets about food categories.

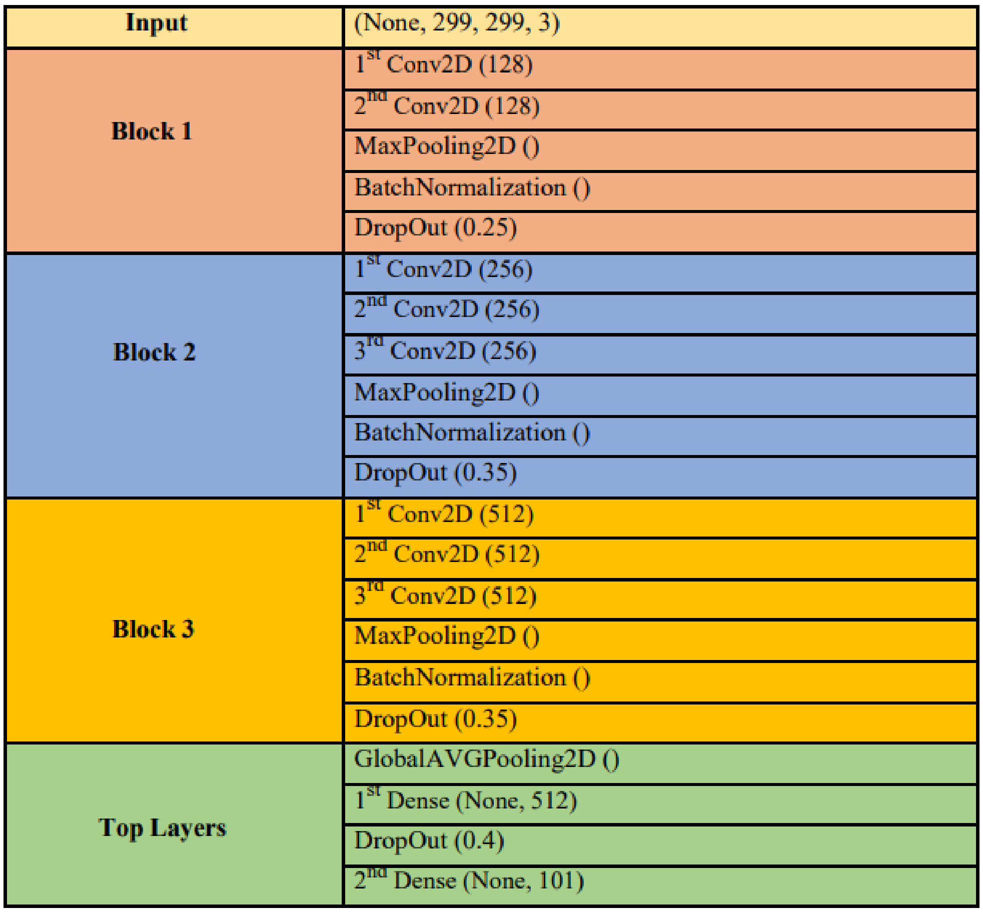
Name	Classes	Images	Year	Description
Food-101N	101	310,000	2018	General food categories

ABLE 5.1: Some popular food datasets

The UECFOOD-100 was introduced in 2012 by Yuji Matsuda, Hajime Hoashi and Keiji Yanai. It contains 100 kinds of food, and each photo has a bounding box indicating the location of the food item in the photo. The main purpose when building this dataset was for implementing a practical food recogni-tion system using in Japan, hence, most of 100 food categories are popular only in Japan. In 2014, Japanese scientists released a new food dataset called UECFOOD-256 (Kawano & Yanai, 2014). It can be considered as an expanded version of UECFOOD-100, while it keeps a dataset structure same as previous version and increases from 100 to 256 classes. Again, the dataset was builtin order to develop a mobile application used in Japan, hence, it is limited for other kinds of food outside the Asian country. Bossard, Guillaumin and Van Gool from the Computer Vision Lab, ETH Zurich, Switzerland introduced Food-101 in the same year. The dataset has 101 food classes, and each class contains a thousand images. It is also split into training and test set with ratio 3:1. The dataset was checked manually, and the training set contains some noise on purpose such as wrong label and colour issues. However, the dataset does not have information for localizing food item in each photo. (Bossard, et al., 2014).

Deep Learning Methods

This section provides fundamental knowledge about 3 aforementioned approaches in CNN, and some projects related to these approaches in food detection context. Firstly, when solving a real-world problem with AI technology, there is always option for building a new architecture from scratch. Thisis a com-plex and time-consuming process that requires many skills from researchers, developers such as mathematics, programming, algorithms, as well as understanding and ability to pre-process data. For this approach, when building new CNN architecture, developer will add convolutional layers, until the model starts performing well and overfits. After that, the model's hyper parameter isoptimized through tuning process until results high accuracy and good fitting. This process requires a powerful hardware system for computing. One of introduced models for this approach is PureFoodNet which consumed a full month for training and configuring on a modern computing system



Input	(None, 299, 299, 3)
Block 1	1st Conv2D (128)
	2nd Conv2D (128)
	MaxPooling2D ()
	BatchNormalization ()
	DropOut (0.25)
Block 2	1st Conv2D (256)
	2nd Conv2D (256)
	3rd Conv2D (256)
	MaxPooling2D ()
	BatchNormalization ()
	DropOut (0.35)
Block 3	1st Conv2D (512)
	2nd Conv2D (512)
	3rd Conv2D (512)
	MaxPooling2D ()
	BatchNormalization ()
	DropOut (0.35)
Top Layers	GlobalAVGPooling2D ()
	1st Dense (None, 512)
	DropOut (0.4)
	2nd Dense (None, 101)

TABLE 2: Pure Food-Net architecture (Simonyan & Zisserman, 2015)

Table 2 presents architecture of the Pure Food-Net, which contains 3 convolution blocks and a classification block. The block 1 contains 2 convolutional layers, each layer has 128 filters, after that max pooling, batch normalization and drop out techniques are applied to avoid over fitting. As a sequence, the number of filters is increased to 256 and 512 corresponding with block 2 and 3. At top layers, the prediction is refined with dense layers to output 101 classes corresponding with 101 classes of Food-101 dataset. Pure Food-Net was demonstrated that the depth neural network is beneficial for the classification accuracy (Simonyan & Zisserman, 2015).

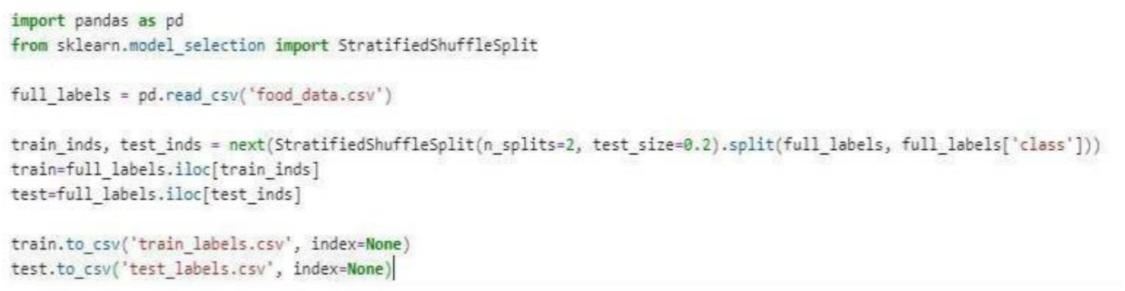
Pre-Processing Data

In order to reach a model with good detector, TensorFlow needs a large number of food photos which consist of random objects along with the desired objects, in addition to variety of backgrounds and light-Ing condition. In contrast, it consumes more computation power and takes more time for preparing data. Therefore, at scale of this application, the model is designed to have the ability to detect only 10 classes of food. Furthermore, it is necessary to indicate the location of desired objects in each image with bounding boxes. As introduced before, UECFOOD-256 is a dataset that provides location of the food item in the photos, therefore, 10 classes of food which are not only popular in Japan but also around the world are picked from UECFOOD-256 and present in table below.

Class	Images in folder	Index in UECFOOD-256	Index in modified dataset
Rice	620	1	1
croissant	120	13	2
hamburger	233	17	3
pizza	134	18	4
sandwiches	163	19	5
fired chicken	154	55	6
steak	108	61	7
simmered pork	109	74	8
green salad	342	87	9
French fries	153	98	10

TABLE 3: Modified dataset

Tensor Flow Object Detection API requires inputs as Tensor Flow record format; therefore, it is necessary write image information into XML file which annotates objects' s location as well as other necessary information. After that, there is a function to read XML file and extract information into CSV file. Finally, from CSV files and images, Tensor Flow record files for training and testing are created and ready for constructing model. The functions for writing information to XML is presented in Appendix 1, after that there is an iteration to each class directory to read images corresponding with bounding box information. It is noticed that class information is important and must be written to XML file, therefore find category function is de-fined to return a correct food name. The code is shown in figure 6, in which the modified dataset is kept in food data directory, after running, XML files are created and stored in annotations folder. An example of output XML file is presented on the next page. It is noticed that there are 2 classes: hamburger and French fries indicated in this image.



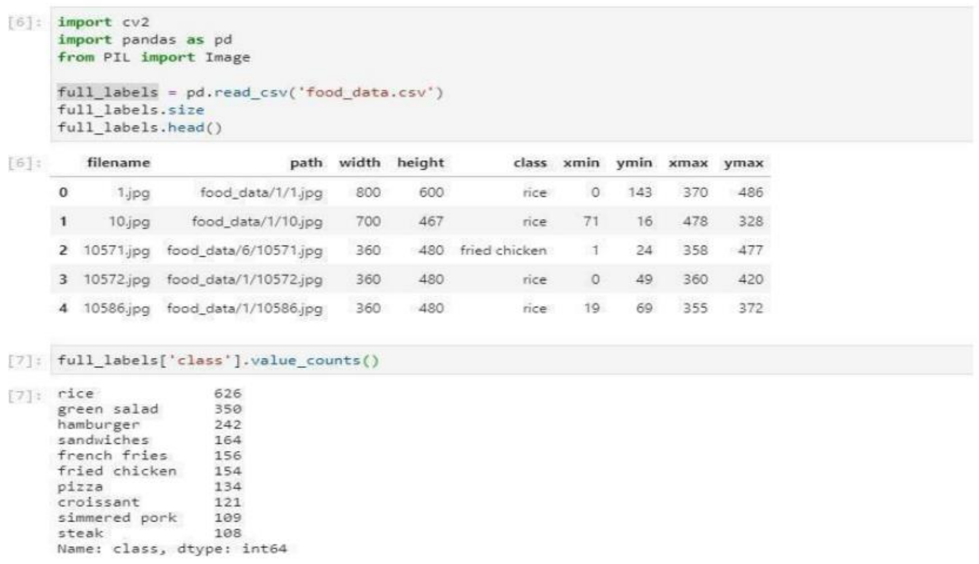
```
import pandas as pd
from sklearn.model_selection import StratifiedShuffleSplit

full_labels = pd.read_csv('food_data.csv')

train_inds, test_inds = next(StratifiedShuffleSplit(n_splits=2, test_size=0.2).split(full_labels, full_labels['class']))
train=full_labels.iloc[train_inds]
test=full_labels.iloc[test_inds]

train.to_csv('train_labels.csv', index=None)
test.to_csv('test_labels.csv', index=None)
```

Fig 6: Split data into training and testing set

The outputs are train_label.csv and test_label.csv, each of set has enough 10 classes of food, and theimage which is in train set cannot exist in test set. Furthermore, images of each class are split with ratio 8:2 into train and test set correspondingly. For example, the total image in rice class is 626 (thereare 620 images in rice directory, but rice also can present in images of other directories, after splitting train set has 501 images and test set has 125 images of rice class. The exploration of these dataset is presented later in chapter 4. The final step of preparation data is creating TensorFlow record files based on train_label.csv and test_label.csv as well as food images. The method implementation is deduced from documentation of TensorFlow Object Detection API and modified into specific use case of this thesis data. The code is presented in Appendix 3, and run-in environment which is installed TensorFlow Object Detection API and other required libraries.

Exploration Datasets

During preparation data for training model, food_data.csv is created from XML files, using Python with necessary library, the data is displayed in figure 7. The CSV file has 4 columns which indicate file name, path, width, height of image and 5 columns which indicate food item's class and positionof item in image. It is noticed that rice, green salad and hamburger contribute dominant images to dataset. Each class in top 3 has more than 200 images, while other class has under 200 images. Especially, rice has the number of images greater than sum of green salad and hamburger.



```
[6]: import cv2
     import pandas as pd
     from PIL import Image

     full_labels = pd.read_csv('food_data.csv')
     full_labels.size
     full_labels.head()
```

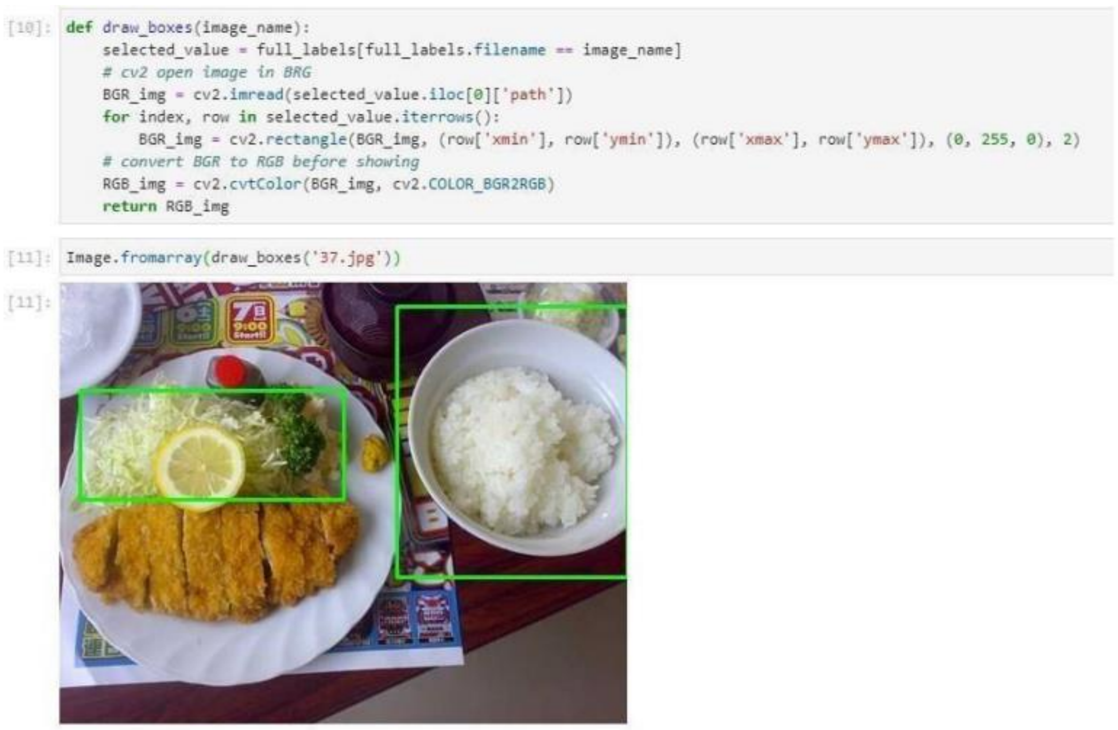
[6]:

	filename	path	width	height	class	xmin	ymin	xmax	ymax
0	1.jpg	food_data/1/1.jpg	800	600	rice	0	143	370	486
1	10.jpg	food_data/1/10.jpg	700	467	rice	71	16	478	328
2	10571.jpg	food_data/6/10571.jpg	360	480	fried chicken	1	24	358	477
3	10572.jpg	food_data/1/10572.jpg	360	480	rice	0	49	360	420
4	10586.jpg	food_data/1/10586.jpg	360	480	rice	19	69	355	372

```
[7]: full_labels['class'].value_counts()

[7]: rice              626
     green salad       350
     hamburger         242
     sandwiches        164
     french fries      156
     fried chicken     154
     pizza             134
     croissant         121
     simmered pork     109
     steak             108
     Name: class, dtype: int64
```

FIGURE 7: Exploration food data

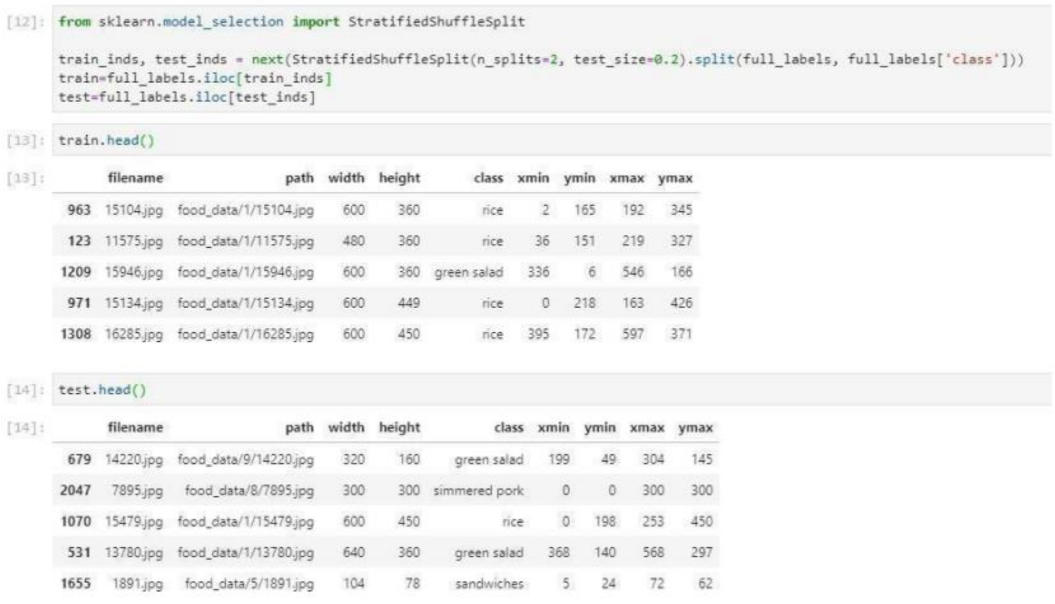
Using bounding box information in dataframe, draw_boxes function can show images and localize food items in image, the function and used are presented in image below. This function is important for re-checking data. It guarantees the information of each image is collected in a right way during the preparation process. For instance, there are some images that they have more than 1 food item, hence, their XML files must annotate all objects. And the CSV file must have different row for each object of one image.



```
[10]: def draw_boxes(image_name):
          selected_value = full_labels[full_labels.filename == image_name]
          # cv2 open image in BRG
          BGR_img = cv2.imread(selected_value.iloc[0]['path'])
          for index, row in selected_value.iterrows():
              BGR_img = cv2.rectangle(BGR_img, (row['xmin'], row['ymin']), (row['xmax'], row['ymax']), (0, 255, 0), 2)
          # convert BGR to RGB before showing
          RGB_img = cv2.cvtColor(BGR_img, cv2.COLOR_BGR2RGB)
          return RGB_img

[11]: Image.fromarray(draw_boxes('37.jpg'))

[11]:
```

FIGURE 8: Draw box function and usage

After re-checking, the data-frame is well built. Then, it is fed into sklearn library's function for splitting data. The function creates 2 new data-frames which are train and test data-frame from the original. It can be seen in figure 14 on the next page, 1731 and 433 are numbers of row in train and test data-frame respectively. In addition, the row which is in train data-frame will not exist in test set to guarantee the mean of evaluating model process. The ratio is defined 8:2 in split method. For instance, steak class has total 108 images in original data frame, after splitting, train set has 86 images, and the number of images goes into test set is 22. The numbers of row grouped by class in each data-frame are illustrated in Appen-dix 14.



```
[12]: from sklearn.model_selection import StratifiedShuffleSplit

      train_inds, test_inds = next(StratifiedShuffleSplit(n_splits=2, test_size=0.2).split(full_labels, full_labels['class']))
      train=full_labels.iloc[train_inds]
      test=full_labels.iloc[test_inds]

[13]: train.head()
```

[13]:

	filename	path	width	height	class	xmin	ymin	xmax	ymax
963	15104.jpg	food_data/1/15104.jpg	600	360	rice	2	165	192	345
123	11575.jpg	food_data/1/11575.jpg	480	360	rice	36	151	219	327
1209	15946.jpg	food_data/1/15946.jpg	600	360	green salad	336	6	546	166
971	15134.jpg	food_data/1/15134.jpg	600	449	rice	0	218	163	426
1308	16285.jpg	food_data/1/16285.jpg	600	450	rice	395	172	597	371

```
[14]: test.head()
```

[14]:

	filename	path	width	height	class	xmin	ymin	xmax	ymax
679	14220.jpg	food_data/9/14220.jpg	320	160	green salad	199	49	304	145
2047	7895.jpg	food_data/8/7895.jpg	300	300	simmered pork	0	0	300	300
1070	15479.jpg	food_data/1/15479.jpg	600	450	rice	0	198	253	450
531	13780.jpg	food_data/1/13780.jpg	640	360	green salad	368	140	568	297
1655	1891.jpg	food_data/5/1891.jpg	104	78	sandwiches	5	24	72	62

FIGURE 9: Train and test data-frame

IMPLEMENTATION

Assessment trained model with real images The model is tested with images that does not come from UECFOOD-256 dataset to assess how good the model predictions with real-word images. The test is also written and implemented in Google Colab. The practical images consist of pictures captured by smartphone and pictures downloaded from the in-ternet, then uploaded to test_images folder. The function can be referenced in appendix, basically, it goes through each image, detects possible items and draws bounding boxes. The result of one image is shown in as figure 10, the model correctly detects the dish is green salad and localizeobject's position in image.

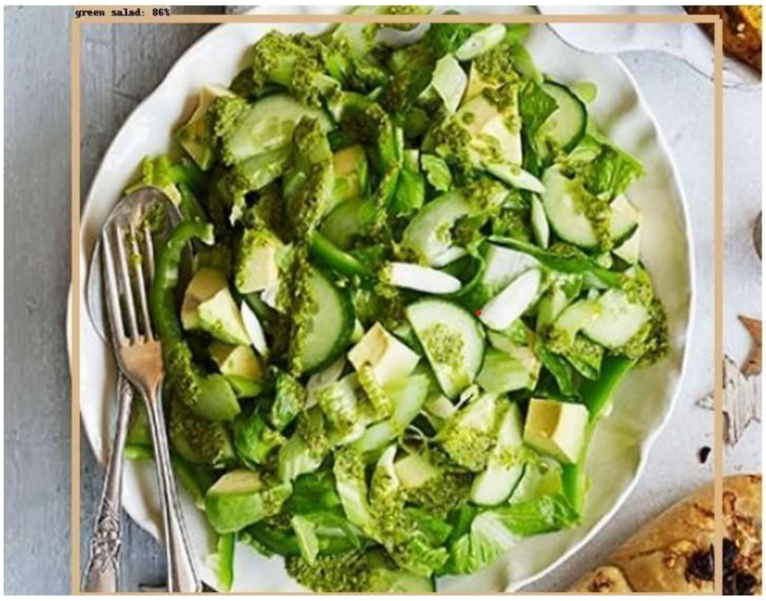


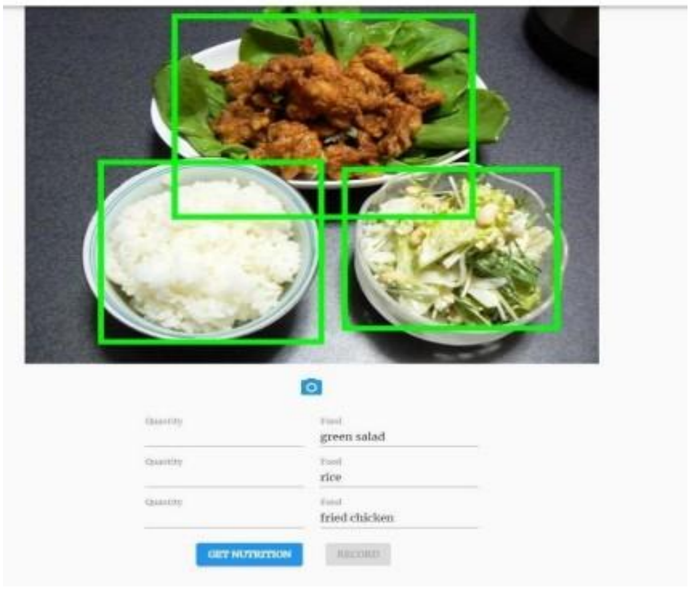
FIGURE 10. Test model with real image

Web application

The web application is deployed and available at this link: 'https://www.nutritionix.com/food/'

The user interface for registering new user is illustrated in figure 10. After successful sign up, user can login and experience food detection function, one notice that the web application is responsive with small screen device such as table or smartphone, in addition, user can select dark mode or light mode of interface. When logged in, user can upload an image for detecting food items by clicking on camera button, select image and then upload. If the model cannot detect any food in uploaded image, one alert message would be sent to user, otherwise, user would have a view which is similar to figure 11 on the next page. The application will draw bounding boxes for these items that it can recognize. It also prints out the list of detected objects, each object has an empty text fields that user can input quantity.

Interface in case model found food items



Quantity	Food green salad
Quantity	Food rice
Quantity	Food fried chicken

Fig 11: Interface in case model found food items

These quantity fields are not obligate to input, because they are set 100g as default value. User can immediately press get nutrition button when there is a list of detected food items. After getting nutrition data from Calories Ninja and summarizing, a table is shown up which presents detailed nutrition information. It depends on screen of device, then nutrition facts are horizontally presented in desktop or table devices same as figure 11. On the other hand, in small device like smartphone, the information is vertically displayed. Finally, user can press record button and that data is stored to database.

SYSTEM TESTING

Testing is an important phase in the development life cycle of the product. This is the phase, where the remaining errors, if any, from all the phases are detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. During the testing, the program to be tested was executed with a set of test cases and the output of the program for the test cases was evaluated to determine whether the program was performing as expected. Errors were found and corrected by using the below stated testing steps and correction was recorded for future references. Thus, a series of testing was performed on the system, before it was ready for implementation.

Levels of Testing

1] 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.

2] 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.

3] System Testing 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.

4] Acceptance Testing Acceptance testing is a test conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is basically done by the user or customer. However, other stockholders can be involved in this process.

5]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.

6] 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.

7]Summary The above section consists of the various kinds of testing that are carried out for checking and ensuring proper working of the system.

CONCLUSION

Food tracking is an efficient method in order to prevent overweight, especially obesity. Through tracking data, people can have insights about their eating behaviors, as a result, they can balance between their daily intake of calories and consumption calories. Furthermore, it is important to control the percentage of protein, carbohydrates and fat that one person absorbs every day. Other criteria suchas cholesterol, sodium and sugar are useful values for in prevention and treatment diseases like highblood pressure, heart disease, and metabolic syndrome. In order to generate and enhance a quick and useful method for food tracking, Deep Learning neural network is used to create a food detection model. Then, the model is embedded into a web applicationwhich can be used in both desktop and mobile device. At scale of a proof of concept, UECFOOD- 256 dataset is used to narrow down and generate a dataset with 10 classes. Applying transfer learningand fine-tuning techniques on TensorFlow Detection Object API with custom data, a food detectionmodel is created with acceptable precision. The model proves to operate as designed expectation in web application which written by React JavaScript and backed by Firebase.

FUTURE ENHANCEMENT

There are several potential future enhancements for a computer vision-based food recognition systemwith nutrition analysis. Here are some possibilities:

1. Improved Accuracy: One of the key areas for improvement is the accuracy of the system. With the help of more advanced deep learning techniques, the system can be trained to recognize a wider variety of foods with greater accuracy. Moreover, more sophisticated algorithms can be developed to analyze the nutritional content of food items more accurately.

2. Integration with Wearable Devices: The system can be integrated with wearable devices such as smartwatches or fitness trackers to provide real-time information about the nutritional content of the food being consumed. This will help users make healthier food choices and monitor their daily calorie intake more effectively.

3. Automatic Portion Size Estimation: Estimating the portion size of the food is an important aspect of nutrition analysis. Future enhancements could include using computer vision techniques to estimate the portion size of the food items automatically, thereby improving theaccuracy of the nutritional analysis.

4. 3D Food Modeling: 3D modeling of food items can provide a more detailed and accurate representation of the food's shape, size, and texture, allowing for more precise nutritional analysis. Future enhancements could include developing algorithms for 3D food modeling and integrating them into the system.

5. Personalized Nutritional Recommendations: The system can be enhanced by providing personalized nutritional recommendations based on the user's dietary requirements, health goals, and other factors. This could be achieved by incorporating machine learning techniquesthat analyze the user's dietary patterns and preferences.

Overall, the future enhancements for a computer vision-based food recognition system with nutrition analysis are vast and exciting. With advancements in deep learning, computer vision, and other related technologies, the system can be further developed to provide more accurate and personalized nutritional recommendations to users.

RESULTS AND SNAPSHOTS

The results of a computer vision-based food recognition system with nutrition analysis can vary depending on the accuracy of the machine learning model and the quality of the data used for training and testing. Here are some possible results that can be obtained from such a system:

1. Food Recognition Accuracy: The accuracy of food recognition is a crucial metric for the system. A well-trained model can accurately recognize different types of food items in images, even when they are partially occluded or presented at different angles.

2. Nutritional Analysis Accuracy: The accuracy of the nutritional analysis is another critical metric for the system. The system should be able to accurately estimate the nutritional content of different food items based on their images, including calories, carbohydrates, protein, fat, vitamins, and minerals.

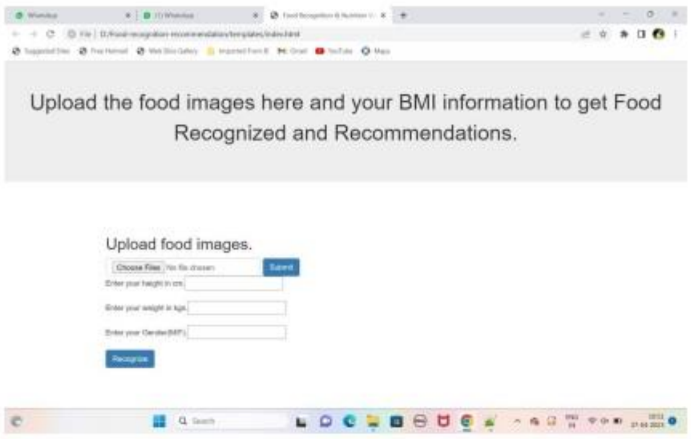
3. Real-Time Nutrition Information: The system can provide real-time nutrition information to users, enabling them to make healthier food choices. Users can take a picture of their meal using their smartphone, and the system will provide the nutritional content of each food item in the meal.

4. Personalized Nutritional Recommendations: The system can provide personalized nutritional recommendations to users based on their dietary requirements, health goals, and other factors. This can help users make informed decisions about what to eat and how much to eat.

5. Improved Dietary Habits: By providing users with accurate nutrition information and personalized recommendations, the system can help improve their dietary habits over time. This can lead to better overall health outcomes, such as weight loss, reduced risk of chronic diseases, and improved mental wellbeing.

Overall, a well-designed and well-implemented computer vision-based food recognition system with nutrition analysis can provide accurate and valuable nutrition information to users, helping them make healthier food choices and improve their overall health and wellbeing.

SNAPSHOTS



Upload the food images here and your BMI information to get Food Recognized and Recommendations.

Upload food images.

Choose Files No file chosen
Submit

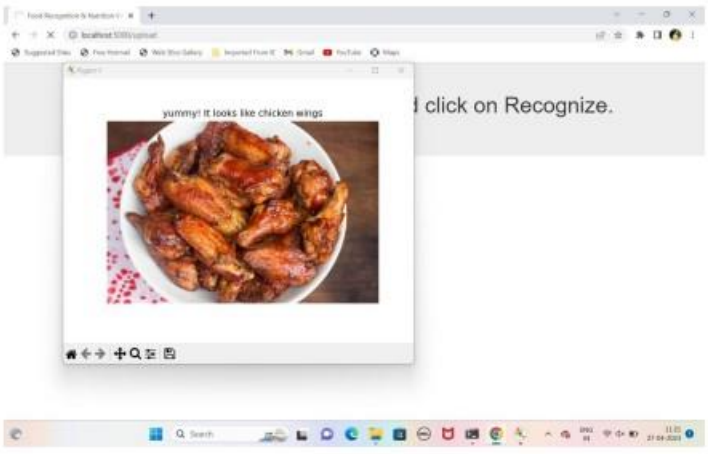
Enter your height in cm

Enter your weight in kgs

Enter your Gender(M/F)

Recognize

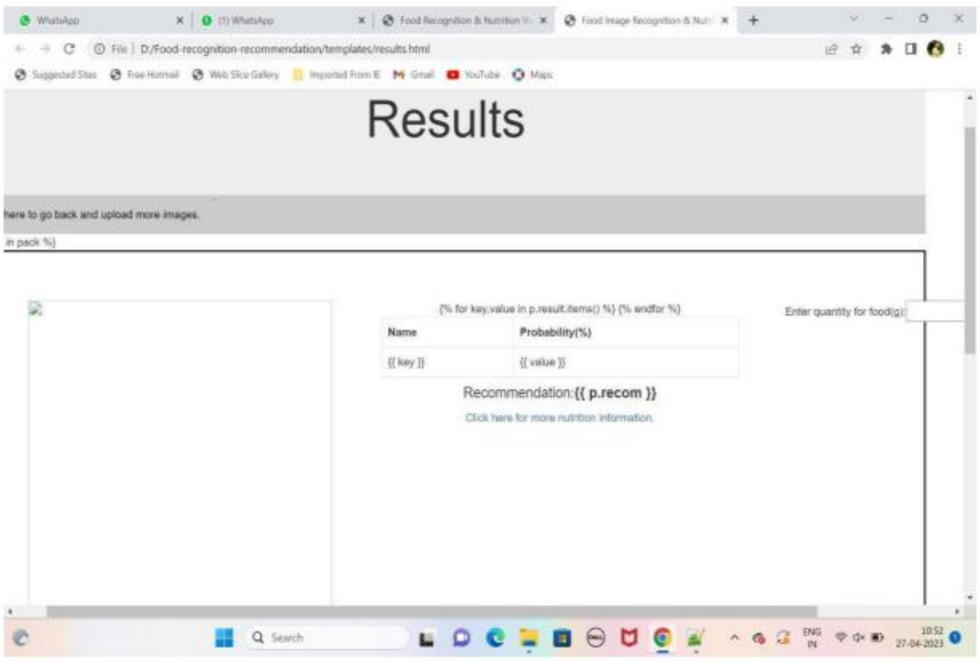
Fig.12: Front Page of the Food image to upload



yummy! It looks like chicken wings

d click on Recognize.

Figure 13 : Recognition of food image


WhatsApp
(1) WhatsApp
Food Recognition & Nutrition V
Food Image Recognition & Nutr
File | D:/Food-recognition-recommendation/templates/results.html
Suggested Sites
Free Hotmail
Web Slice Gallery
Imported From IE
Gmail
YouTube
Maps
Search
10:52
27-04-2023

Results

here to go back and upload more images.

in pack %}

{% for key,value in p.result.items() %} {% endfor %}

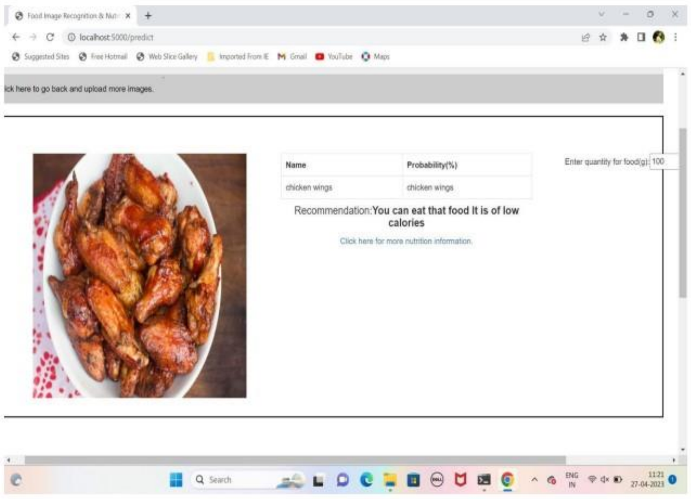
Name	Probability(%)
{{ key }}	{{ value }}

Recommendation:**{{ p.recom }}**

Click here for more nutrition information.

Enter quantity for food(g):

Figure 14: Results of the food image



ick here to go back and upload more images.

Name	Probability(%)
chicken wings	chicken wings

Recommendation:**You can eat that food It is of low calories**

Click here for more nutrition information.

Enter quantity for food(g): 100

Figure 15 : Shows the how much the calories is present

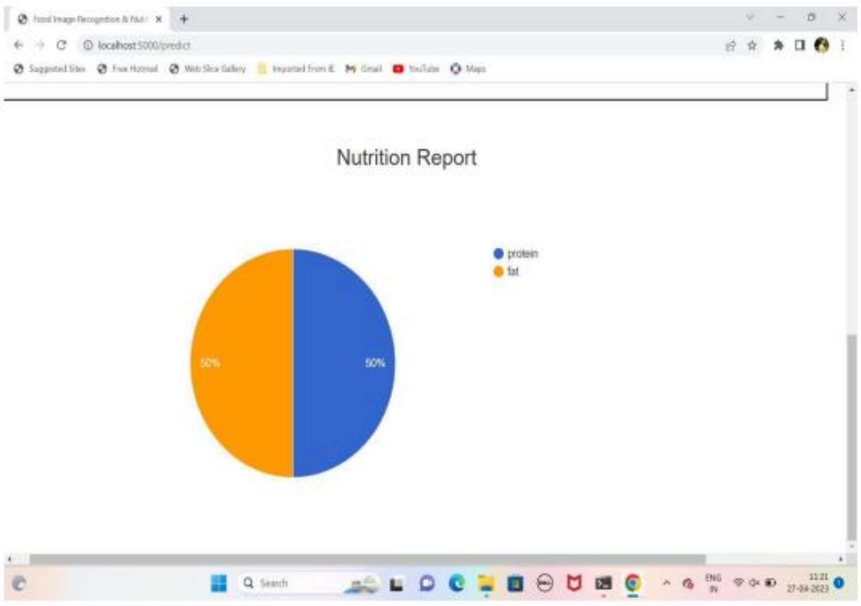
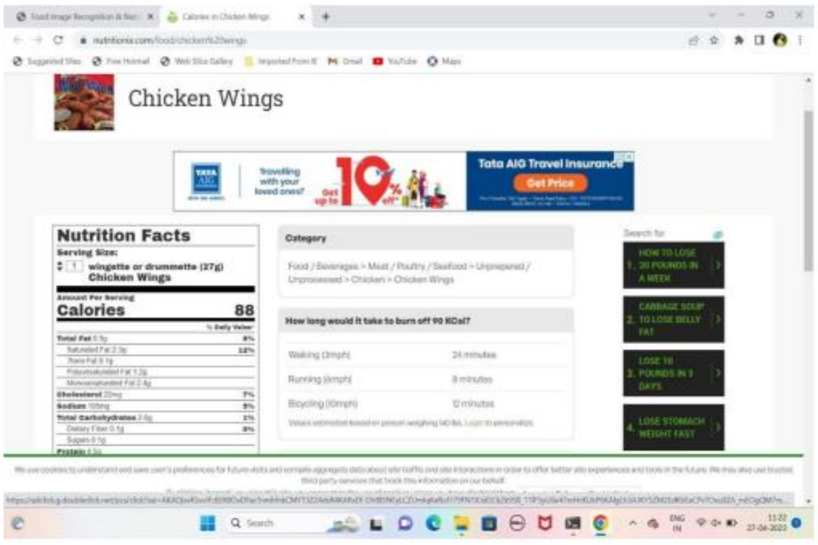


Figure 16: Shows the Nutrition content present in the food



Chicken Wings

Nutrition Facts
Serving Size:
1 wingette or drummette (27g) Chicken Wings

Amount Per Serving
Calories 88

% Daily Value*

Total Fat 5.7g	8%
Saturated Fat 2.3g	12%
Trans Fat 0.1g	
Polyunsaturated Fat 1.2g	
Monounsaturated Fat 2.4g	
Cholesterol 22mg	7%
Sodium 105mg	5%
Total Carbohydrates 2.6g	1%
Dietary Fiber 0.1g	0%
Sugars 0.1g	
Protein 4.5g	

Category

Food / Beverages > Meat / Poultry / Seafood > Unprepared / Unprocessed > Chicken > Chicken Wings

How long would it take to burn off 90 KCal?

Walking (3mph)	24 minutes
Running (6mph)	8 minutes
Bicycling (10mph)	12 minutes

Values estimated based on person weighing 140 lbs. Login to personalize.

nutritionix.com/food/chicken%20wings
Tata AIG Travel Insurance
Get Price
Search for
1. HOW TO LOSE 30 POUNDS IN A WEEK
2. CABBAGE SOUP TO LOSE BELLY FAT
3. LOSE 10 POUNDS IN 3 DAYS
4. LOSE STOMACH WEIGHT FAST

Figure 17: More information about the food image

REFERENCES

[1]. Abadi, M. et al., 2016. TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265-283.Amazon, 2021. What is Amazon Rekognition. [Online] Available at: https://docs.aws.amazon.com/rekognition/latest/dg/what- is.html [Accessed 6 March 2021].

[2]. Bossard, L., Guillaumin, M. & Gool, L. V., 2014. Food-101 – Mining DiscriminativeComponents with Random Forests. European Conference on Computer Vision, Volume 8694, pp. 446-461.Brazier, Y., 2017. How many calories do you need?. [Online] Available at: https://www.medicalnewstoday.com/articles/263028 [Accessed 1 March 2021].

[3]. CalorieNinjas, 2020. CaloriesNinja - Easy, Free Nutrition Facts Search. [Online] Available at: https://calorieninjas.com [Accessed 1 March 2021].Chen, J. & Ngo, C.-w., 2016. Deep-based Ingredient Recognition forCooking Recipe Retrieval. Proceedings of the 24th ACM international conference on Multimedia, pp. 32-41.

[4]. Chollet, F., 2015. Keras. [Online] Available at: https://keras.io/ [Accessed 6 March 2021].Cioca, G., Napoletano, P. & Schettini, R., 2018. CNN-based features for retrieval and classification of food images. Computer Vision and Image Understanding, Volume 176-177, pp. 70-77.

[5]. Ciocca, G., Napoletano, P. & Schettini, R., 2016. Food Recognition: A New Dataset, Experiments, and Results. IEEE Journal of Biomedical and Health Informatics, 21(3), pp. 588- 598.Ciocca, G., Napoletano, P. & Schettini, R., 2017. Learning CNN-based Features for Retrievalof Food Images. In: New Trends in Image Analysis and Processing – ICIAP 2017. Catania, Italy: Springer International Publishing, pp. 426-434.

[6]. Facebook, 2021. React. [Online] Available at: https://reactjs.org/ [Accessed 7 March 2021].Fung, T. T., Long, M. W., Hung, P. & Cheung, L. W. T., 2016. An Expanded Model for Mindful Eating for Health Promotion and Sustainability: Issues and Challenges for Dietetics Practice. Academy of Nutrition and Dietetics, Issue 116, pp. 1081- 1086.

[7]. Microsoft, 2020. What is Computer Vision?. [Online] Available at: https://docs.microsoft.com/en-us/azure/cognitive-services/computervision/overview [Accessed 7 March 2021].Paszke, A. et al., 2017. Automatic defferentiation inPyTorch. 31st Conference on Neural Information Processing Systems.

[8]. Simonyan, K. & Zisserman, A., 2015. Very Deep Convolutional Networks for Large Scale Image Recognition. 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 730-734.Singh, V., 2018. Computer Vision Based Food Recognition with Nutrition Analysis Dept. Of CSE, DSATM 2022-23 Page 71

[9]. Introduction to Firebase. [Online] Available at: https://medium.com/codingurukul/introduction-to-firebase-f9f6ccc8a785 [Accessed 7 March2021].Wang, X. et al., 2015. Recipe recognition with large multimodal food dataset. 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1- 6.

[10]. World Health Organization, 2020. Overweight and Obesity. [Online] Available at: https://www.who.int/news-room/fact-sheets/detail/obesity-and overweight [Accessed 1 March 2021].Yanai, K. & Kawano, Y., 2015. Food imagerecognition using deep convolutional network with pre-training and finetuning. 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1-6