**A Project Report on**

Food Seeker

Submitted in partial fulfillment of award of

### **BACHELOR OF TECHNOLOGY**

Degree

in

(**COMPUTER SCIENCE AND ENGINEERING**)

By

JUHI RASTOGI -1608210071

PRATHAM RATHORE-1608210114

MOHD. TABREZ KHAN-1608210088

GUNJAN RADHAWAL-1608210060

MOHD. BILAL-1608210086

(2016-2020)

Ms. Priyanka Goel

Asst. Professor

CSE dept.

SUPERVISOR



**Dept. of Computer Science & Engineering**

**Moradabad Institute of Technology**

**Moradabad (U.P.)**

**CERTIFICATE**

This is to certify that the PROJECT REPORT entitles “**FOOD SEEKER”** has been duly submitted by Juhi Rastogi (1608210071), Pratham Kumar Singh Rathore (1608210114), Mohd. Tabrez Khan (1608210088), Mohd Bilal (1608210086), Gunjan Radhawal (1608210060)**”** is their own work and has been carried out under my supervision. It is recommended that the candidates may now be evaluated for their project work by the University.

**Ms. Priyanka Goel**

**Project guide**

**Date: (Asst. Professor)**

**ABSTRACT**

The content of my project report is food recognition and detection of the respective recipe with just one camera click. In this project we are making two platforms for food recognition and recipe detection that is, an Android Application and a Web Portal. Basically our project is used to label the food you take the picture of, then searches our database for recipe.

We apply a convolutional neural network (CNN) to the tasks of detecting and recognizing food images. Because of the wide diversity of types of food, image recognition of food items is generally very diﬃcult. We applied CNN to the tasks of food detection and recognition through parameter optimization. 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. We are constructing the dataset of the most frequent food items through the google image downloader script which is created using the python and TensorFlow technology, and those images are used to evaluate the recognition performance. CNN showed significantly higher accuracy than did traditional support-vector-machine-based methods with handcrafted features. For food image detection, CNN also showed signiﬁcantly higher accuracy than a conventional method did. Therefore, we use CNN algorithm in our project.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

Automatic image-based food recognition is a particularly challenging task. Traditional image

analysis approaches have achieved low classification accuracyin the past, whereasdeep learning

approaches enabled the identification of food typesandtheir ingredients. The contents of food dishes

are typically deformable objects, usually including complex semantics, which makes the task of

defining theirstructure very difficult. Deep learningmethodshave already shown very promising

results in suchchallenges, so this chapter focuses on the presentation of some popularapproachesand

techniquesapplied inimage-based food recognition.The three main lines of solutions, namely the

design from scratch,the transfer learning and the platform-based approaches, are outlined, particularly

for the task at hand, and are tested and compared to reveal the inherent strengths and weaknesses. The

chapter is complemented with basic background material, a section devoted to the relevant datasetsthat

are crucial in light of the empirical approaches adopted, and some concluding remarks that underline

the future directions.

**ACKNOWLEDGEMENT**

I express my deepest sense of gratitude towards my project guide **Ms. Priyanka Goel** for her patience, inspiration, guidance, constant encouragement, moral support, keen interest, and valuable suggestions during preparation of this Project report.

My heartfelt gratitude goes to all faculty members of Computer Science & Engineering Dept., who with their encouraging and caring words and most valuable suggestions have contributed, directly or indirectly, in a significant way towards completion of this Project report.

I am indebted to all my classmates for taking interest in discussing my problem and encouraging me.

I owe a debt of gratitude to my father and mother for their consistent support, sacrifice, candid views, and meaningful suggestion given to me at different stages of this work.

Last but not the least I am thankful to the Almighty who gave me the strength and health for completing my report.

JUHI RASTOGI

PRATHAM KUMAR SINGH RATHORE

MOHD. TABREZ KHAN

MOHD BILAL

GUNJAN RADHAWAL

**TABLE OF CONTENT**

**CERTIFICATE ii**

**ABSTRACT iii**

**ACKNOWLEDGEMENT iv**

1. INTRODUCTION 10
   1. MOTIVATION 11
   2. OVERVIEW OF PROPOSED SYSTEM 12
   3. SOFTWARE REQUIREMENT ANALYSIS 13
   4. APPLICATION 14
2. LITERATURE SURVEY 15

2.1) STAGES OF IMAGE PROCESSING 15

2.1.1) IMAGE ACQUISITION 15

2.1.2) IMAGE ENCHANCEMENT 16

2.1.3) IMAGE RESTRORATION 16

2.1.4) COMPRESSION 17

2.1.5) MORPHOLOGICAL IMAGE PROCESSING 17

2.1.6) SEGMENTATION 19

2.1.7) REPRESENTATION AND DESCRIPTION 19

2.1.8) RECOGNITION AND INTERPRETATION 20

2.2) MACHINE LEARNING 20

2.2.1) CNN (CONVOLUTIONAL NEURAL NETWORK) 20

2.2.2) BUILDING A DATASET OF ORDINARY FOOD ITEM 23

2.3) BRIEF INTRODUCTOION OF DEEP LEARNING 23

2.4) ANDROID 26

2.4.1) FEATURES 27

2.4.2) ANDROID APPLICATION 29

2.4.3) ANDROID-ENVIRONMENT SETUP 29

2.4.4) ANDROID ACTIVITIES 30

2.5) NAÏVE BAYES 32

2.5.1) CLASSIFICATION MODEL 33

2.5.2) NAÏVE BAYES MODEL 34

2.5.3) DECISION TREE MODEL 35

3) MODULES OF PROJECT 37

3.1) MODULE 1 37

3.1.1) WEB APP FOR FOOD RECOGNITION AND DETECTION 37

3.1.2) DRAG AND DROP 39

3.1.3) PROCESS 41

3.1.4) WEB SCRAPING 41

3.1.5) DATA REPRESENTATION 42

3.1.6) MODELS OF BUILDING 43

3.1.7) MODELS BUILDING 45

3.2) MODULE 2 47

3.2.1) GOOGLE IMAGE DOWNLOADER SCRIPT 47

3.2.2) FOOD RECOGNITION 48

3.3) MODULE 3 50

3.3.1) CREATING THE DATASET 50

3.3.2) CHALLENGES FOR CREATING DATASET 51

3.4) MODULE 4 51

3.4.1) ANDROID APPLICATION 51

3.4.2) IMAGE UPLOADER 53

3.4.3) OPTIONS FOR UPLOADING PICTURE 55

3.4.4) GETTING FEATURES FROM IMAGE 56

3.5) FLOWCHART OF THE PROJECT 58

1. METHODOLOGY 61

4.1) PREFACE 61

4.2) SCRAPING AND PREPARING THE DATA 64

4.3) DATA ANALYSIS AND VISUALISATION 65

4.4) TOPIC MODELLING 66

4.4.1) LATENT SEMANTIC ANALYSIS 67

4.4.2) LATENT DIRICHLET ANALYSIS (LDA) 68

4.5) FEATURE EXTRACTION 68

4.6) TRAINING PRETRAINED DATA 69

4.7) DEPLOYING THE FLASK 70

1. CONCLUSION 72

5.1) CONCLUSION 72

5.2) LIMITATIONS 73

5.3) FUTURE SCOPE 74

REFERENCES 76

**LIST OF FIGURES**

**Figure No. Name of Figure Page No.**

* 1. A Typical CNN Structure for Image Classification 13

2.1 Stages of Image Processing 15

* 1. Image Compression Model 17

2.3 Probing of an Image with a Structuring element 18

2.4 Architecture of CNN 22

* 1. Typical SAE Structure 25
  2. Chart representation android function 31
  3. Classification Model 34

2.8 Food addiction model 34

2.9 Naïve bayes model 35

2.10 Decision tree model 36

3.1 Feature visualization of convolutional 37

3.2 Frontend of web application 39

3.3 Code of drag and drop 39

3.4 Code for drag and drop 40

3.5 Code for drag and drop 40

3.6 Web scraping 41

3.7 Recognition of dish 42

3.8 Dataset cleaning 42

3.9 CNN Architecture 43

3.10 Image classifier Model Comparisons 44

3.11 Training and Validation Accuracy with Early Stopping 44

3.12 Training and Validation Loss with Early Scrapping 45

3.13 Analyzing of dish 45

3.14 Recognition of dish 46

3.15 Recognition of dish 46

3.16 Flowchart of creating dataset 48

3.17 Images of few food items 49

3.18 Recognition of downloader script 49

3.19 Food recognition demo image 53

3.20 Image uploading in android application 54

3.21 Options for selecting image 55

3.22 Analyzing of food item 57

3.23 Recognition of dish 57

3.24 Recognition of food item 58

3.25 Flowchart of script 59

3.26 Flowchart of complete project 60

4.1 Plate of food 62

4.2 Schema Method 63

4.3 Architecture of model based CNN 70

**CHAPTER 1**

**INTRODUCTION**

The project performs the food recognition and detection of the respective recipe with just one camera click. In this we are making two platforms for food recognition and recipe detection that is, an Android Application and a Web Portal. Basically our project is used to label the food you take the picture of, then searches our database for recipe.

Healthy diet is important to human health. Natural products have been widely used as food, and they can also be processed to meet the demand of consumers. Food (natural products and processed food) attributes such as type, compositions, nutrients, and process styles are concerned issues for healthy diet. It is a fact that people from different regions have different eating habits. Knowing the attributes of food is important to inspect food quality and safety for consumers all over the world. Life log is very helpful to know one’s own life style. By logging various aspects of our life such as exercise, activity, etc., the details of our life can be visualized from various point of view. Food is one of the most important components of our life. Records of what we eat is essential for our health care. We have been investigating the so called “Food Seeker” system, where the user takes photos of his foods and uploads them to the system, and the system performs image processing to detect food images and estimate food recipe. In this research paper we try to present image processing for food images. And there are still some work related to image processing of food images. Kitamura et al.[1] proposed our previous system which detects food images and estimates the food recipe. They extracted the three kinds of features; color, circle and SIFT features, and put them into Support Vector Machine (SVM). And they were capable for detecting food images by 92% accuracy and the food balance by 38%. the two scientist attempted to calculate the food menu out of 50 items. They use and machine learning algorithm method to merge with the Bag of Features, color histogram, and other texture features, and achieved the accuracy by 61.24% classiﬁcation rate. Zhu et al attempted to estimate the food item and the volume by using the food images before and after eating. The dataset was limited to 50 images, and they prepared the same white dishes on a checkerboard to segment the food items and estimated. This work is supported by JST CREST. Wu and Yang used videos of eating behavior to estimate the number of calories in a given meal. Their target was only fast foods such as MacDonald’s, and they used key point matching of SIFT features. In 2010, Yang etal. Also treated the fast food images in Pittsburgh Food Image Dataset (PFID), and proposed a method to classify the food category by using pair-wise local features. Pair-wise means that distance or orientation between two pixels in the image and the local features means color histogram and Bag of SIFT and then, put them into SVM. In this paper, we present a method to make use of the corrections made by the user simply by Naive Bayes which is one of the Bayesian networks. We are trying two types of experiments about food image detection. Firstly, we are comparing the accuracy of the performance between SVM and Naive Bayes. [1] The F-measure of the former value is 0.89 and the latter one is 0.85. Secondary, we investigate how improve the accuracy by using user feedbacks. The food-seeker system enables the user to correct the results of the image processing. Then, making use of the user’s corrections as feedback, we can update the Bayesian network to improve the performance. The recall is improved from 0.89 to 0.92.

**1.1 MOTIVATION**

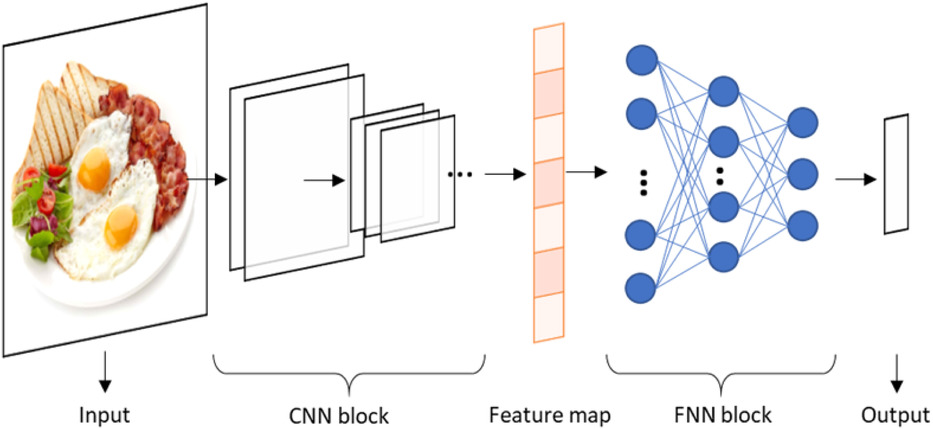
Nowadays, deep learning has been introduced into food field by analyzing RGB images and spectra images of food. However, due to the fact that understanding and APP of deep learning is a difficult issue for researchers and workers related in food industry, the researchers are on the way. The objective of this survey is to present a comprehensive overview of the latest research progress in the APP of deep learning in food field, and provide guidance for researchers and workers in this field.

Recent technological advances such as smart-phones equipped with cameras and other rich sensors, pervasive networks and artificial intelligence have powered new uses of technology related with food. For illustration, conventional food gathering for purpose of diet monitoring requires expertise and effort from the user, and is prone to inaccuracies and forgetting. On the contrary, a self-working food annotation system could perform automatic analysis, annotation and logging with minimum human intervention. For illustration, photos from android-phones are convenient yet powerful entry points to many applications involving recognition, retrieval or recommendation. In this direction, food-oriented social networks and restaurant review services have bloomed, where food enthusiasts (e.g., foodies, gourmets, cooks) connect and share information (e.g., recipes, photos, comments about restaurants).Thus, authentic food analysis from the images is essential part for the applications. Even though there are remarkable advances in computer vision, food recognition will still remain a very challenging problem even for humans. We largely rely on contextual and prior information. Similarly, context and prior knowledge can be integrated in automatic food analysis systems [5].

**1.2 OVERVIEW OF PROPOSED SYSTEM**

This online application enables the end users to take picture of the food, read the database and check for the label. The results, after taking food pictures will directly appear on the screen. This app has many areas available to extensibility, from utilizing more advanced machine learning models for food recognition, adding in recommendation systems, integrating better routing and price comparisons, and granting user submitted recipes with voting and peer reviews.

We apply a Convolutional Neural Network (CNN) to the tasks of detecting and recognizing food images. Because of the wide diversity of types of food, image recognition of food items is generally very difficult. Though, deep learning has been shown as the very powerful and important image recognition technique, and CNN is a state of the art approach to deep learning. 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. We applied CNN to the tasks of food detection and recognition through parameter optimization. CNN including a set of components (convolutional layers, pooling layers, fully connected layers, and so on) is currently considered as one of the most popular machine intelligence models for big data analysis in various research areas. A typical architecture of CNN model for classification problems is displayed in Fig 1.1.



**Fig 1.1 A typical CNN structure for image classification.**

**1.3 SOFTWARE REQUIREMENT ANALYSIS**

The platform and language used for the project:

* The platform used for the development of the project is Java, JavaScript, PHP, and Python.
* The IDE used is Android Studio.
* Tensor flow as library.

The minimum Hardware and Software requirements that are currently being used in the project are:

* Dual-core 64 bit processor
* 4GB of memory(RAM)
* Up to 24GB of internal storage
* Android Studio version
* Java, Android SDK : 2GB,
* Windows SDK: 4GB plus ample space for multiple complex projects.
* Windows 10, Windows 8.1, Windows 8 and Windows 7.

**1.4 APPLICATION**

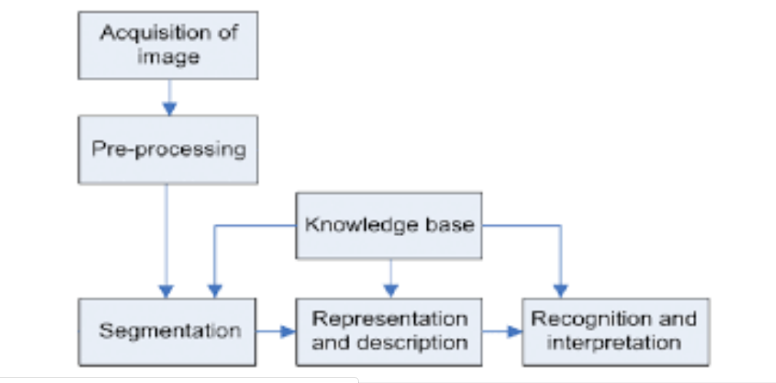
We focus on context and knowledge modeling, but there are also interesting applications of automatic food recognition to self-service restaurants and dining halls. For instance, accurate detection and segmentation of the different food items in a food tray can be used for monitoring food intake and nutritional information. It is also used for the purpose of detection of recipes of the food item that you take from your phone’s camera and give the desired output according to food item detected and the corresponding recipe related to that food that will help the user and as well as chef who are using this application.

The photos from smart-phones are convenient yet powerful entry points to many applications involving recognition, retrieval or recommendation [5]. In this direction, food-oriented social networks are used, where food enthusiasts (e.g., foodies, gourmets, cooks) connect and share information (e.g., recipes, photos, comments about restaurants).Thus, reliable food analysis from images is essential for these applications. Despite remarkable advances in computer vision, food recognition in the wild still remains a very challenging problem even for humans. We largely rely on contextual and prior information. Similarly, context and prior knowledge can be integrated in food analysis systems [13].

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 STAGES IN IMAGE PROCESSING**



**Fig 2.1 Stages for Image processing**

**2.1.1 IMAGE ACQUISITION**

This is the first step or process of the fundamental steps of digital image processing. In the present scenario images have become the most suitable way to keep our past alive. Nowadays people are very busy in earning their livelihood and day to day life, even though they want to keep their golden moment alive for the rest of life. Images have touched almost all the fields like medical, sports, social networking and many more. It is the need of time to know how the images are being captured and stored into memory. To deal with images and before analyzing them the most important thing is to capture the image. This is called as Image Acquisition. Image Acquisition is achieved by suitable camera. We use different cameras for different application. If we need an X-Ray image, we use a camera (film) which is sensitive to X- Rays. If we want Infra-Red image, we use cameras which are sensitive to Infra-Red radiations. For normal images (family pictures etc.) we use cameras which are sensitive to visual spectrum [15].

**2.1.2 IMAGE ENCHANCEMENT**

Image enhancement techniques are [mathematical techniques](https://www.sciencedirect.com/topics/engineering/mathematical-technique) that are aimed at realizing improvement in the quality of a given image. The result is another image that demonstrates certain features in a manner that is better in some sense as compared to their appearance in the original image. One may also derive or compute multiple processed versions of the original image, each presenting a selected feature in an enhanced appearance. Simple image enhancement techniques are developed and applied in an ad hoc manner. Advanced techniques that are optimized with reference to certain specific requirements and objective criteria are also available.

**2.1.3 IMAGE RESTORATION**

The concerns of the image restoration are the removal or reduction of degradations which are included during the acquisition of images e.g.; Noise, pixel value errors, out of focus blurring or camera motion blurring using prior knowledge of the degradation phenomenon. This means it deals with the modelling of the degradation and applying the process (inverse) to reconstruct the image. The image restoration has got a wide scope of usage [12].

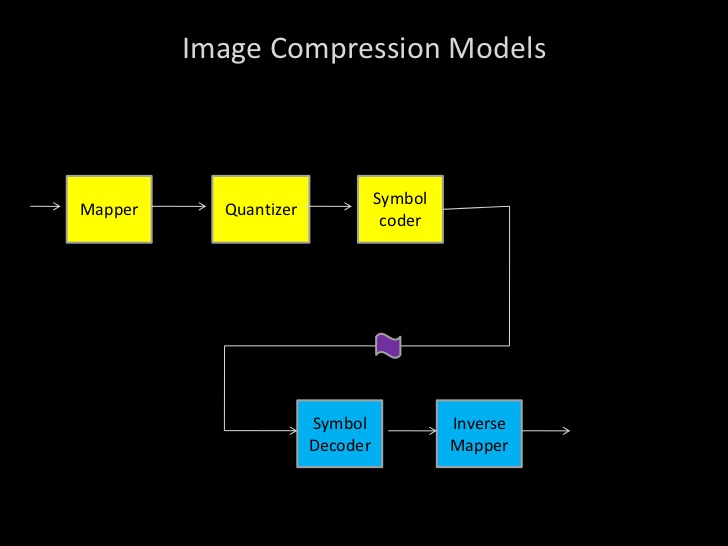
The purpose of image restoration is to compensate for or undo effects. The orientation of the image restoration techniques is towards modelling the degradations such as blur and noise which involves the applications of various filters to obtain the original scene approximation. Input image is degraded by a degradation function say h(x,y) and channel transmission noise n(x,y), degraded image g(x,y) can be obtained. In mage restoration the target is to obtain the approximate target to the input. The blurred image can be described with the following equation.

g(x,y) = h(x,y) \* f(x,y) + n(x,y) (2.1)

**2.1.4 COMPRESSION**

At its core, image compression is when you remove or group together certain parts of an image file in order to reduce its size. Why do that? Here are a few reasons.

1. For [website optimization](https://www.keycdn.com/blog/website-performance-optimization)- Sites with uncompressed images can take longer to load, and can cause your visitors to bounce because of this.
2. For sending and uploading images- Uploading an uncompressed image can take a while, and some email servers have a file size limit.
3. For reducing the storage impact on your hard drive. The compression is of 2 types: Lossless and Lossy compression [15].

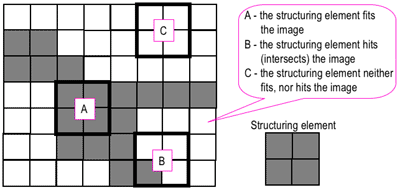


**Fig. 2.2 Image Compression Model**

**2.1.5 MORPHOLOGICAL IMAGE PROCESSING**

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to grey-scale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbor-hood of pixels. Some operations test whether the element "fits" within the neighbor-hood, while others test whether it "hits" or intersects the neighbor-hood:



**Fig.2.3 probing of an image with a structuring element**

A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one:

* The matrix dimensions specify the size of the structuring element.
* The pattern of ones and zeros specifies the shape of the structuring element.
* An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

A common practice is to have odd dimensions of the structuring matrix and the origin defined as the center of the matrix. Structuring elements play in morphological image processing the same role as convolution kernels in linear image filtering. When a structuring element is placed in a binary image, each of its pixels is associated with the corresponding pixel of the neighborhood under the structuring element. The structuring element is said to fit the image if, for each of its pixels set to 1, the corresponding image pixel is also 1. Similarly, a structuring element is said to hit, or intersect, an image if, at least for one of its pixels set to 1 the corresponding image pixel is also 1.Zero-valued pixels of the structuring element are ignored, i.e. indicate points where the corresponding image value is irrelevant.

* + 1. **SEGMENTATION**

Segmentation partitions an image into distinct regions containing each pixels with similar attributes. To be meaningful and useful for image analysis and interpretation, the regions should strongly relate to depicted objects or features of interest. Meaningful segmentation is the first step from low-level image processing transforming a greyscale or color image into one or more other images to high-level image description in terms of features, objects, and scenes. The success of image analysis depends on reliability of segmentation, but an accurate partitioning of an image is generally a very challenging problem. Segmentation techniques are either contextual or non-contextual. The latter take no account of spatial relationships between features in an image and group pixels together on the basis of some global attribute, e.g. grey level or color. Contextual techniques additionally exploit these relationships, e.g. group together pixels with similar grey levels and close spatial location [12].

* + 1. **REPRESENTATION AND DESCRIPTION**

Representing regions in 2 ways:

* Based on their external characteristics (its boundary): – Shape characteristics
* Based on their internal characteristics (its region): – Regional properties: color, texture. Describes the region based on a selected representation: boundary or textural features
* Representation: length, orientation, the number of
* Description: concavities in the boundary, statistical measures of region.
  + 1. **RECOGNITION AND INTERPRETATION**

Just like the phrase “What-you-see-is-what-you-get” says, human brains make vision easy. It doesn’t take any effort for humans to tell apart a dog, a cat or a flying saucer. But this process is quite hard for a computer to imitate: they only seem easy because God designs our brains incredibly good in recognizing images. A common example of image recognition is optical character recognition (OCR). A scanner can identify the characters in the image to convert the texts in an image to a text file. With the same process, OCR can be applied to recognize the text of a license plate in an image.

Image interpretation covers a wide range of techniques to extract meaningful information from raw image data. [Statistical Shape Models](https://www.sciencedirect.com/topics/computer-science/statistical-shape-model) (SSMs) provide a means to describe the variation in shape of an object class across a set of images, allowing the qualitative and quantitative analysis of image data. For the detailed analysis of the shape of an object using SSMs, the object of interest needs to be annotated using many points in every image. Depending on the object type (e.g. faces or skeletal structures), this annotation may not only be tedious and time-consuming but may also require significant expertise to correctly place the points. In this chapter, we are going to introduce the basic methodology behind SSMs, and describe a number of automated annotation methods to enable fast and objective shape analyses with a wide variety of applications [15].

**2.2 MACHINE LEARNING**

**2.2.1 CNN (CONVOLUTIONAL NEURAL NETWORK)**

To know what we eat, we often make a record of every-day meals. Such food recording is usually a manual exercise using textual description, but manual recording is tedious and time consuming. To overcome this difficulty, there have been attempts to assist food recording by using information technology. Image recognition of food items would be a good solution to food recording. Taking a picture would then suﬃcient record. However, we know that there is a wide diversity of types of food. Even within the same food category, there is considerable diversity. Therefore, despite the attempts at food item recognition, recognition performance is not yet satisfactory.

Regarding food image recognition, Zhu et al. described food recognition using a small dataset, which was intended to be used in a smart-phone-based food-logging system as part of their Technology Assisted Dietary Assessment project. Hoashi et al. examined 85 food items, achieving 62.5% accuracy for the recognition of Japanese food images collected from the Web. They used multiple kernel learning for feature fusion as their machine learning method. The Pittsburgh Fast-Food Image Dataset is a dataset of American fast-food images, which was used to evaluate a food-recognition method in [11, 6].Food balance, an aspect of nutritional content, was estimated by image processing. Image retrieval was applied to food recording. Deep learning has recently been used in image recognition. Deep learning is a collective term for algorithm shaving a deep architecture that solves complex problems. The most distinctive characteristic is that better image features for recognition are automatically extracted via training. The convolutional neural network (CNN) is one of the methods that satisfy the requirements of the deep learning approach. CNN is now a state-of-the-art technique for image recognition challenges such as the Large Scale Visual Recognition Challenge. [4]

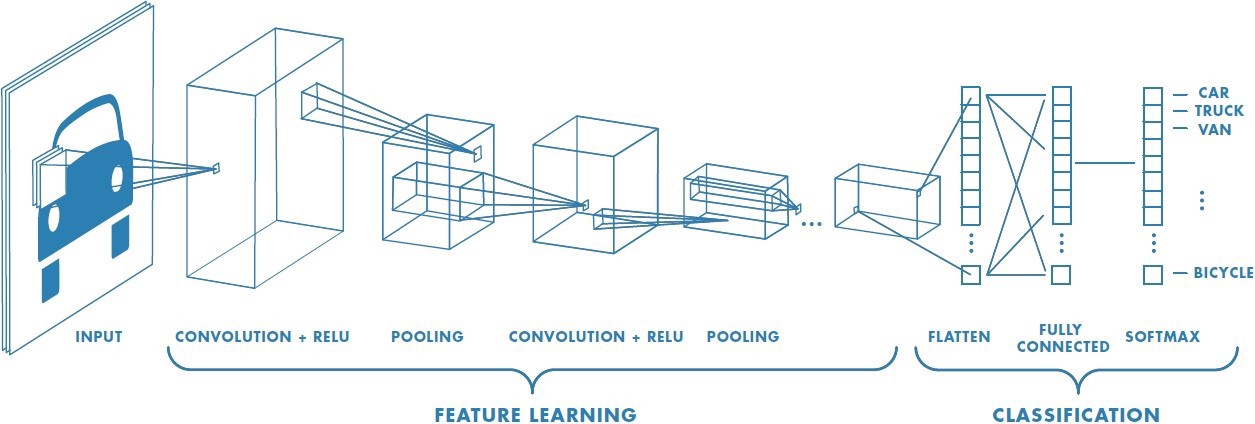
In this, we apply CNN to the recognition and detection of food images and evaluate its performance. Our contributions are as follows:

(1) We built a dataset for food recognition experiments by using food-domain images obtained from a food logging system available for public use;

(2) We optimized CNN’s hyper parameters, showing that CNN signiﬁcantly improved the food recognition accuracy compared with a conventional method using a support vector machine (SVM) with hand-crafted features;

(3) Through observation of our trained CNN, we found that color features dominate the food recognition process;

(4) We showed that CNN has signiﬁcantly better performance for the task of food detection when compared with a baseline method.

The CNN oﬀers a state-of-the-art technique for image recognition. It is a multilayer neural network, whose neuron stake small patches of the previous layer as input. It is robust against small shifts and rotations. A CNN system comprises a convolution layer and a pooling (or sub-sampling) layer. In the convolution layer, unlike for general fully connected neural networks, weights can be considered as n\*(n <input size) ﬁlters. Each input convolves these ﬁlters. Each layer has many ﬁlters that generate diﬀerent outputs. For the image recognition task, the diﬀerent features are extracted by these ﬁlters. The ﬁlters are often called (convolution) kernels. The pooling layer produces the outputs by activation over rectangular regions. There are several activation methods, such as maximum activation and average activation. This makes the CNN’s outputs more invariant with respect to position. A typical CNN comprises multiple convolution and pooling layers, with a fully connected layer to produce the ﬁnal result of the task. In image classiﬁcation, each unit of the ﬁnal layer indicates the class probability [4]. 

**Fig.2.4 Architecture of CNN**

A CNN has hyper parameters that include the number of middle layers, the size of the convolution kernels, and the active functions.

**2.2.2 BUILDING A DATASET OF ORDINARY FOOD ITEM**

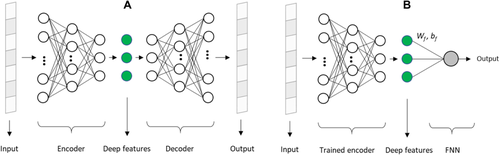
A number of images of ordinary meals are required for the evaluation of food item recognition. A meal image typically involves several food items. In the evaluation of food item recognition, each food item region of the image need to be identiﬁed and isolated for the dataset. Some food apps available for smart-phones can produce very good data for this purpose. The general public can use Food learning for their own food recording using both photos and text. The user takes a photo of a meal, and speciﬁes each region involving a food item via the touch panel display of the smart-phone by inputting the name of the food item. The food item name is usually chosen from a standard food database. As a result, very clean data about image regions of named food items are produced. In our experiments with food recognition, the domains that users speciﬁed were slightly enlarged because users tend to specify undersized regions for the food items. FL is a publicly available app, and, with the number of users growing, the food item dataset is increasing. This data is from everyday meals logged by general public FL users. The diversity of the types of food was large and there was heavy bias towards some items.

**2.3 BRIEF INTRODUCTION OF DEEP LEARNING**

Machine learning has been active in various fields, which acts as an effective tool for data processing. For the lack of ability to analyse raw natural data, traditional machine learning techniques usually needs to be supplemented by a manual feature extraction method. With the development of hardware computing ability and storage capacity, the abilities of machine learning can be promoted by adding more complex structures to achieve deep representation of the data. Representation learning enables a machine to extract the features from raw data for detection, classification, or regression. Deep learning can be understood as a kind of representation‐learning method that refines multilevel representation by utilizing the deep ANN composed by multiple layers of neurons (nonlinear modules). Due to the strong feature learning ability of deep learning method, many complex problems can be solved in a rapid and effective way. Deep learning models demonstrate powerful capabilities in classification/regression tasks, provided that adequate data support was available which represents the specific problem. With the strong ability of automatic feature learning, deep learning method starts to be applied in the field of food science, mainly referring to food category recognition, fruit and vegetable quality detection, food calorie estimation, and so on. We will introduce in detail in section “Deep learning applications in food [6].”

CNN including a set of components (convolutional layers, pooling layers, fully connected layers, and so on) is currently considered as one of the most popular machine intelligence models for big data analysis in various research areas. Convolution operations are implemented by traversing input matrices with convolution kernels that can be understood as filters for feature extraction. Different from filters used in conventional image processing method whose parameters need to be set manually, the parameters inside the kernel can be learned automatically by deep learning method. Convolutional layers are built by a set of convolution kernels, whose parameters (channels, kernel size, strides, padding, activation, and so on) should be set and optimized according to the practical problem. The computed output from convolutional layer is then sub-sampled by pooling layers. A group of chained convolutional layers and pooling layers can learn high-level features representing the original input. The fully connected network (FNN) block, composed by fully connected neural units, is usually placed at the end as the classifier or used to generate numerical output for regression problems exploiting the learned feature map [4].

Figure below illustrates another widely used deep learning model named stacked auto encoders (SAE), whose structure is similar to conventional ANNs. SAE can be used as an unsupervised learning method to obtain features from input. See Figure (A), the input is encoded into a vector with fewer dimensions than that of itself. The trained encoder coupled with FNN shown in Figure (B) can be employed as a deep learning model to solve classification or regression problems.

[](https://onlinelibrary.wiley.com/cms/asset/3573e6c4-6d1a-4a93-af04-a0bce82a5063/crf312492-fig-0002-m.jpg)

**Fig 2.5 Typical SAE structure.**

**(A) SAE model composed of an encoder and a decoder that would be trained.**

**(B) The decoder is removed and the encoder is retained**

Besides classification and regression, deep learning also demonstrates strong capabilities to process image segmentation tasks. Fully convolutional network (FCN) is a landmark progress in the field of image segmentation. Similar to general convolution networks, FCN contains several convolution layers for feature extraction, but the FNN block in an ordinary CNN is replaced by a deconvolution layer that up-samples the feature map and expands the same width and height of output as those of the input images. For instance, when a food image is fed into an FCN model, the CNN and sub-sampling block extracts valuable information of the input food image. Then, the up-sampling block outputs the segmentation result that is an image with the same rows and columns as the input. Each pixel inside the output image represents a category. In Figure, red, blue, green, and orange pixels represent “beef,” “eggs,” “vegetables,” and “bread,” respectively. Thus, FCN can be regarded as a pixel-to-pixel network [6].

The following is an explanation to the training and optimization process of the network: The weights of deep neural networks (DNNs) are initialized randomly or by Xavier method and tuned during the training procedure including forward and back propagation process. In the forward propagation process, the difference between the output value (and predicted value) and the label value (or ground truth) is calculated according to the defined loss function. In the backpropagation process, the weight of the neural network is updated to minimize the loss function via Stochastic Gradient Descent (SGD), and Adam algorithms. The hyper parameters of the network, such as learning (which controls the pace of weight adjustment), batch size, number of convolution kernels and layers, and so on, could be fitted by evaluating the performance (the output of loss function) on validation set. Besides, the combination of feature graphs, the form of convolution kernels, and the parallel network structure should be considered depending on the specific problems.

Besides the three kinds of frequently used models mentioned in this section, there are many other network structures such as recurrent neural network (RNN), long short‐term memory (LSTM), single-shot multi-box detection (SSD), region‐based CNN), sequence to sequence, text, and so on. Processing data types are not only limited to RGB images, but also can be any other forms of data such as video, audio, voice, text, natural language, and so on. The diets and eating habits can affect health of human beings. Especially for diabetics, allergic people, and so on, they should strictly monitor and control their dietary behaviour. Food recognition and classification is an important task to help human beings record the daily diets. Images of food are one of the most important information to reflect the characteristics of food. Moreover, image sensing is a relatively easy and low‐cost information acquisition tool for food appearance analysis. For natural products like food and processed food, the large variations in food shape, volume, texture, color, and compositions make food recognition a challenging task. Various background and layout of food stuffs also introduce variations for food recognition and classification. At present, due to the common use of CNN, image analysis has been the most commonly used pattern in food recognition and classification.

**2.4 ANDROID**

Android is a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) developed by [Google](https://en.wikipedia.org/wiki/Google). It is based on a modified version of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and other [open source](https://en.wikipedia.org/wiki/Open-source_software) software, and is designed primarily for [touch-screen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smart-phones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). In addition, Google has further developed [Android TV](https://en.wikipedia.org/wiki/Android_TV) for televisions, [Android Auto](https://en.wikipedia.org/wiki/Android_Auto) for cars, and [Wear. OS](https://en.wikipedia.org/wiki/Wear_OS) for wrist watches, each with a specialized user interface. Variants of Android are also used on [game consoles](https://en.wikipedia.org/wiki/Video_game_console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), [PCs](https://en.wikipedia.org/wiki/Personal_computer) and other electronics. Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the [first commercial Android device](https://en.wikipedia.org/wiki/HTC_Dream) launched in September 2008. The operating system has since gone through multiple major releases, with the current version being [9 "Pie"](https://en.wikipedia.org/wiki/Android_Pie), released in August 2018. Google released the first [Android Q](https://en.wikipedia.org/wiki/Android_Q) beta on all Pixel phones on March 13, 2019. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the [Apache License](https://en.wikipedia.org/wiki/Apache_License). Android is also associated with a suite of [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software) developed by Google, called [Google Mobile Services](https://en.wikipedia.org/wiki/Google_Mobile_Services) (GMS) that very frequently comes pre-installed in devices, which usually includes the [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) web browser and [Google Search](https://en.wikipedia.org/wiki/Google_Search_(mobile_app)) and always includes core apps for services such as [Gmail](https://en.wikipedia.org/wiki/Gmail), as well as the [application store](https://en.wikipedia.org/wiki/Application_store) and [digital distribution](https://en.wikipedia.org/wiki/Digital_distribution) platform [Google Play](https://en.wikipedia.org/wiki/Google_Play), and associated [development platform](https://en.wikipedia.org/wiki/Google_Play_Services). These apps are licensed by manufacturers of Android devices certified under standards imposed by Google, but AOSP has been used as the basis of competing Android ecosystems, such as [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com)'s [Fire OS](https://en.wikipedia.org/wiki/Fire_OS), which use their own equivalents to GMS. Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion [monthly active users](https://en.wikipedia.org/wiki/Monthly_active_users), the largest [installed base](https://en.wikipedia.org/wiki/Installed_base) of any operating system, and as of December 2018, the [Google Play](https://en.wikipedia.org/wiki/Google_Play) store features over 2.6 million apps. In June 2014, Google announced [Android One](https://en.wikipedia.org/wiki/Android_One), a set of "hardware reference models" that would "allow [device makers] to easily create high-quality phones at low costs", designed for consumers in developing countries.

* + 1. **FEATURES**

1. Interface: Android's default user interface is mainly based on [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation_interface), using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard). [Game controllers](https://en.wikipedia.org/wiki/Game_controller) and full-size physical [keyboards](https://en.wikipedia.org/wiki/Computer_keyboard) are supported via [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) or [USB](https://en.wikipedia.org/wiki/USB). The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide [haptic feedback](https://en.wikipedia.org/wiki/Haptic_technology) to the user. Internal hardware, such as [accelerometers](https://en.wikipedia.org/wiki/Accelerometer), [gyroscopes](https://en.wikipedia.org/wiki/Gyroscope) and [proximity sensors](https://en.wikipedia.org/wiki/Proximity_sensor) are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a [racing game](https://en.wikipedia.org/wiki/Racing_game) by rotating the device, simulating control of a [steering wheel](https://en.wikipedia.org/wiki/Steering_wheel). Android devices boot to the home-screen, the primary navigation and information "hub" on Android devices, analogous to the [desktop](https://en.wikipedia.org/wiki/Desktop_metaphor) found on personal computers. Android home-screens are typically made up of app icons and [widgets](https://en.wikipedia.org/wiki/Software_widget); app icons launch the associated app, whereas widgets display live, auto-updating content, such as a [weather forecast](https://en.wikipedia.org/wiki/Weather_forecast), the user's email inbox, or a [news ticker](https://en.wikipedia.org/wiki/News_ticker) directly on the home-screen. A home-screen may be made up of several pages, between which the user can swipe back and forth. Third-party apps available on [Google Play](https://en.wikipedia.org/wiki/Google_Play) and other app stores can extensively re-[theme](https://en.wikipedia.org/wiki/Theme_(computing)) the home-screen, and even mimic the look of other operating systems, such as [Windows Phone](https://en.wikipedia.org/wiki/Windows_Phone). Most manufacturers customize the look and features of their Android devices to differentiate themselves from their competitors [9].
2. Requirements in different fields: Applications ("[apps](https://en.wikipedia.org/wiki/Mobile_app)"), which extend the functionality of devices, are written using the [Android software development](https://en.wikipedia.org/wiki/Android_software_development) kit (SDK) and, often, the [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) programming language. Java may be combined with [C](https://en.wikipedia.org/wiki/C_(programming_language))/C++, together with a choice of non-default [runtimes](https://en.wikipedia.org/wiki/Runtime_library) that allow better C++ support. The [Go](https://en.wikipedia.org/wiki/Go_(programming_language)) programming language is also supported, although with a limited set of [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (API). In May 2017, Google announced support for Android app development in the [Kotlin programming language](https://en.wikipedia.org/wiki/Kotlin_(programming_language)).

The SDK includes a comprehensive set of development tools, including a [debugger](https://en.wikipedia.org/wiki/Debugger), [software libraries](https://en.wikipedia.org/wiki/Software_library), a handset [emulator](https://en.wikipedia.org/wiki/Emulator) based on [QEMU](https://en.wikipedia.org/wiki/QEMU), documentation, sample code, and tutorials. Initially, Google's supported [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) was [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)) using the Android Development Tools (ADT) plug-in; in December 2014, Google released [Android Studio](https://en.wikipedia.org/wiki/Android_Studio), based on [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA), as its primary IDE for Android application development. Other development tools are available, including a [native development kit](https://en.wikipedia.org/wiki/Native_development_kit) (NDK) for applications or extensions in C or C++, [Google App Inventor](https://en.wikipedia.org/wiki/Google_App_Inventor), a visual environment for novice programmers, and various [cross platform mobile web applications frameworks](https://en.wikipedia.org/wiki/Multiple_phone_web_based_application_framework). In January 2014, Google unveiled framework based on [Apache Cordova](https://en.wikipedia.org/wiki/Apache_Cordova) for porting [Chrome](https://en.wikipedia.org/wiki/Google_Chrome) [HTML 5](https://en.wikipedia.org/wiki/HTML_5) [web applications](https://en.wikipedia.org/wiki/Web_app) to Android, wrapped in a native application shell.

1. Memory management: Since Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a minimum. When an application is not in use the system [suspends its operation](https://en.wikipedia.org/wiki/Process_state) so that, while available for immediate use rather than closed, it does not use battery power or CPU resources. Android manages the applications stored in memory automatically: when memory is low, the system will begin invisibly and automatically closing inactive processes, starting with those that have been inactive for the longest amount of time. Life hacker reported in 2011 that third-party task killer applications were doing more harm than good.
2. Prerequisites: Android programming is based on Java programming language. If you have a basic understanding of Java programming, then it will be fun to learn Android application development.

**2.4.2 ANDROID APPLICATIONS**

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play** or the **Amazon App-store**. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and is growing fast. Every day more than 1 million new Android devices are activated worldwide. Android is an open source and Linux-based **Operating System** for mobile devices such as smart-phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by Android [10].

**2.4.3 ANDROID – ENVIRONMENT SETUP**

You will be glad to know that you can start your Android application development on either of the following operating systems:

* Microsoft Windows XP or later version.
* Mac OS X 10.5.8 or later version with Intel chip.
* Linux including GNU C Library 2.7 or later.

Second point is that all the required tools to develop Android applications are freely available and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

* Java JDK5 or JDK6
* Android SDK
* Eclipse IDE for Java Developers (optional)
* Android Development Tools (ADT) Eclipse Plugin (optional)

Here last two components are optional and if you are working on Windows machine then these components make your life easy while doing Java based application development. So let us have a look at how to proceed to set the required environment.

**2.4.4 ANDROID ACTIVITIES**

An activity represents a single screen with a user interface. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched.

If you have worked with C, C++ or Java programming language then you must have seen that your program starts from **main()** function. Very similar way, Android system initiates its program within an **Activity** starting with a call on onCreate() callback method. There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity as shown in the below Activity lifecycle diagram: (image courtesy: android.com)

**Fig 2.6 chart represent android functions**

The Activity class defines the following callbacks i.e. events. You don't need to implement all the callback methods. However, it's important that you understand each one and implement those that ensure your app behaves the way users expect.

|  |  |
| --- | --- |
| **Callback** | **Description** |
| onCreate() | This is the first callback and called when the activity is first created. |
| onStart() | This callback is called when the activity becomes visible to the user. |
| onResume() | This is called when the user starts interacting with the application. |
| onPause() | The paused activity does not receive user input and cannot execute any code and called. |
| onStop() | This callback is called when the activity is no longer visible. |
| onDestroy() | This callback is called before the activity is destroyed by the system. |
| onRestart() | This callback is called when the activity restarts after stopping it. |

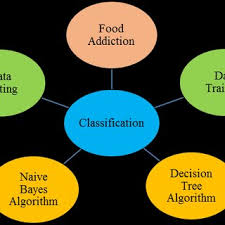
**2.5 NAIVE BAYES**

In food consumption, food addiction is behavioral and biological overlaps have been observed between eating and addictive disorders. Food addiction influence to healthy life. Food addiction caused by overeating, bingo eating, eating disorder, eating addiction, mindless eating, craving, chocaholic, and emotional eating. Determination between addiction and normal condition in food consumption, need to classification. Classification is very important in determine signs of food addiction. Classification using Naïve Bayes Algorithm and Decision Tree Algorithm. Class target is Class Normal and Class Addiction. Classification using Naïve Bayes Algorithm by criterion is Calorie Dense Food, Fatty Food, Sweet Food, Diet and Stress. This criterion as causal factor of food addiction. Classification using Decision Tree Algorithm by criterion is Stress, Fatty Food and Calorie Dense Food. This criterion as causal factor of food addiction. The experimental result is a Classification Model. This model became data source for national policy in public health. People can be addicted to food consumption caused by overeating. Based on analysis in the paper, an addictive perspective suggests an interactive effect between an individual’s predisposition for developing an addiction and an addiction agent. From analysis on paper, Food addiction based on factor neurochemistry like alteration dopamine and endogenous opioids. Also, alteration on limbic system and self-medication behaviors. Food may be addictive is sweets, carbohydrates, fats, sweet/fat combinations and/or high salt foods. Eating topography has been identified in response to some food caused by eating addiction. Response to some food like persistent desire to eat certain foods or cravings. Also, need to eat more of the food to get same felling. Negative effect of behaviors when eating like emotional eating and mindless eating. Assumption that addictive were ‘bad’, people felt that ‘addiction’ should be reduced in food consumption. Based on research in the paper shown that food addiction is factor contributing to over-eating then to obesity [9].

Determination between addiction and normal condition in food consumption, need to classification. Classification became very important in determine signs of food addiction. Food addiction influence healthy life. Result of classification on food addiction as data source for national policy in public health. Classification is a form of data analysis that extracts models describing important data classes. Effective and scalable methods heave developed in for Decision Tree Induction, Naïve Bayes Classification, rule-based classification and many other classification methods. Specification of Naïve Bayes is a statistical classifier(performs probabilistic prediction), foundation based on Bayes Theorem, performance (comparable performance with decision tree and selected neural network classifier), incremental (each training example can incrementally increase/decrease the probability that a hypothesis is correct-prior knowledge can be combined with observed data), and standard (even when Bayesian methods are computationally interact-able, they can provide a standard of optimal decision making against which other methods can be measured).Advantages is easy to implement and good results obtained in most of the cases. Specification of Decision Tree is speed (relatively faster learning speed/than other classification methods), easy (convertible to simple and easy to understand classification rules), access (can use SQL queries for accessing databases), and accuracy (comparable classification accuracy with other methods) [9].

**2.5.1 CLASSIFICATION MODEL**

Classification is the processing of finding a set of model that distinguish data class for the purposes of being able to use the model to predict the class of objects whose class label is unknown. Classification is the process to construct a model based on the training set and uses it to classify new data or test set. It is a supervised learning as observations; measurements are accompanied by known class labels in a large amount of training set and new data is classified based on training set. Classification is one of main activities in the distillation of knowledge by learning supervised approach. Classification on model can be depicted in a framework

****

**Fig 2.7 Classification model**

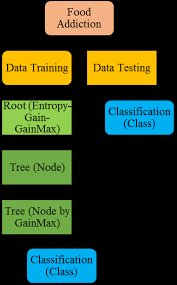
Figure shown that food addiction caused by eating disorder, eating addiction, over eating, bingo eating, emotional eating, and mindless eating. Craving so make people food addiction. Factor chocaholic can be seen as addictive to some food in certain quantity [10].

**Fig 2.8 Food addiction model**

**2.5.2 NAIVE BAYES MODEL**

Naïve Bayes Classifier is classification with the method probability and statistic, namely predict opportunities in future based on experience so time formerly known as Bayes Theorem. The theorem combined with Naïve where assumed conditions attribute between on and another is free. Probabilistic approach as Naïve Bayes Classifier, having several advantages among others, simple, quick and high accuracy. Data Testing used to prior probability, posterior probability, max value of posterior probability, and Class as target in classification. Data Training as data source based on Food Addiction. Classification based Bayes Theorem: Probability (B given A) = Probability (A and B) / Probability (A).

To calculate the probability of B given A, the algorithm counts the number of cases, A and B occur together, and divides it by the number of case where A occurs alone. Let X be a data tuple. In Bayesian terms, X is considered ‘evidence’. Let H be some hypothesis, that the tuple X belongs class C. P (H | X) is the posterior probability of H conditioned on X. In contrast, P (H) is the prior probability of H [9].



**Fig 2.9 Naive Bayes model**

**2.5.3 DECISION TREE MODEL**

A decision tree is a predictive modeling technique from the fields of machine learning and statistics that builds a simple tree-like structure to classify the data according their categories. A decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute. Each branch represents an outcome of the test. Least nodes represent classes or class distributions, shown Figure shown Decision Tree Model step by step. Food Addiction as data source. Data Training used to determine entropy an information gain. Then, selected max value gain. Choice node by gain maximum. Target is Class. Decision Tree based on Decision Tree Induction.

Step 1, tree is constructed in a top-down recursive divide-and-conquer manner.

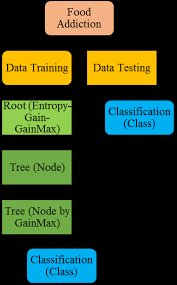
Step 2, at start, all the training examples are at the root.

Step 3, attributes are categorical (if continuous-value, they are discretized in advance).

Step 4, examples are partitioned recursively based on selected attributes.

Step 5, test attributes are selected on the basis of a heuristic or statistical measure.

Conditions for stopping partitioning when all samples or a given node belong to the same class. There are no remaining attributes for further partitioning-majority voting is employed for classifying the leaf, and there are no samples left. Entropy is a measure of uncertainty associated with a random variable. Interpretation, if higher entropy then higher uncertainty and if lower entropy then lower uncertainty [10].

****

**Fig 2.10 Decision tree model**

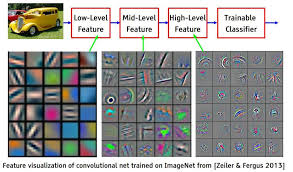
**CHAPTER 3**

**MODULES OF PROJECT**

**3.1 MODULE 1**

**3.1.1 WEB APPLICATION FOR FOOD RFECOGNITION AND DETECTION**

As we all know that web application is a software application that runs on a remote server. So it give the leverage to the user that they used this application from anywhere and from any place they want. They can used any food image that is available in there system to recognize and detect that food item along with their respective recipe. It give them leverage to detect recipe of any dish that they want at any time and at any place. Due to this excessive availability of the food item it difficult for the user to recognize the food recipe and can’t even detect the name of that particular food item so for that case our application comes into play it will detect the name of the food item along with the recipe of that particular food item. And to excess this, Web browsers are used to access Web applications, over a network, such as the Internet. CNN has hyper parameters related to the number of layers, training, preprocessing, and initialization. In this section, we describe the calibration of the number of layers and the number and size of kernels in each layer. There are two types of LRN, involving either one map or across the maps. We divided the dataset into six sets: four sets were used for training, one for validation, and one for testing. We conducted a 6-fold cross validation. With respect to kernel size, 5x5 was the best.



**Fig 3.1 Feature visualization of convolutional neural network**

Comparing color features, the accuracy with CNNs was improved for all items. The blue cells in the results for Color SPM + SVM show high confusion for some combinations of food items of similar colors such as “curry and rice” and “deep-fried chicken.” For CNN, the confusion of these items is much reduced.

We also consider the food detection task. This is diﬀerent from food item recognition in that food detection is a binary classiﬁcation of food and non-food images. Given a whole image that may contain food and background, food detection classiﬁes the image as food or non-food. We conducted an experiment to evaluate CNN’s performance.For this detection task, we used a diﬀerent dataset because they were collected from social media. The input images were scaled down to 80×80 and cropped to 64×64 pixels randomly by python module. For comparison, the baseline was the food detection system used by FL. In our project we are using CNN, we divided all food and non-food images in the dataset into10 groups. Eight groups were used for training, one for validation and one for testing. Training ended when the validation error ceased to evolve. We conducted 10-fold cross-cause of having to include non-food images. Compares the accuracy of the baseline method and CNN, with CNN achieving 93.8% accuracy, which is significantly higher than that for the baseline method. These images would seem to be hard to detect, even by human observation.

Firstly, we do the frontend of the application using the HTML and CSS framework. It will contain the drop box where you can drop your food image to recognize and after pressing the analyze button you will get the details of the image. We create this application for the ease of the user to access the application remotely and from anywhere they want. The frontend will look like this:



**Fig 3.2 Frontend of web Application**

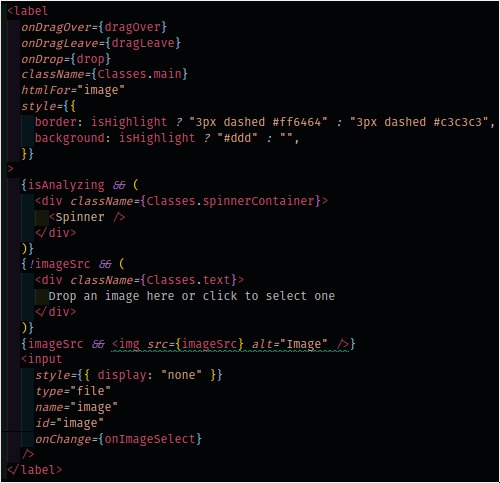
**3.1.2 DRAG AND DROP**

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.In HTML, any element can be dragged and dropped.

The example below is a simple drag and drop example:



**Fig 3.3 Code for Drag and Drop**



**Fig 3.4 Code for Drag and drop**



**Fig 3.5 Code for drag and drop**

**3.1.3 PROCESS**

The process to building my application comprises of 4 stages:

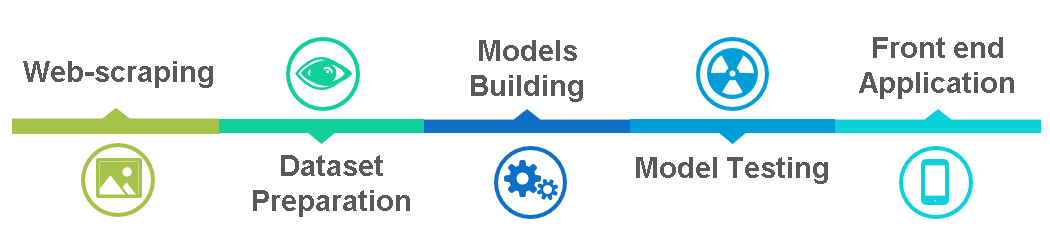
1) Web-scraping

2) Dataset preparation

3) Models Building

4) Model Testing.

5) Front end application

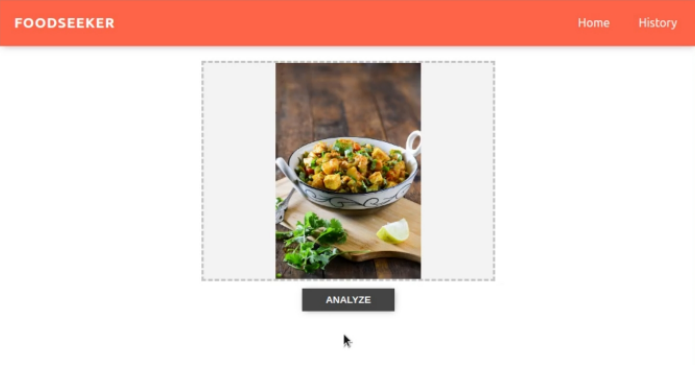


**Fig 3.6 Web Scrapping**

**3.1.4 WEB-SCRAPING**

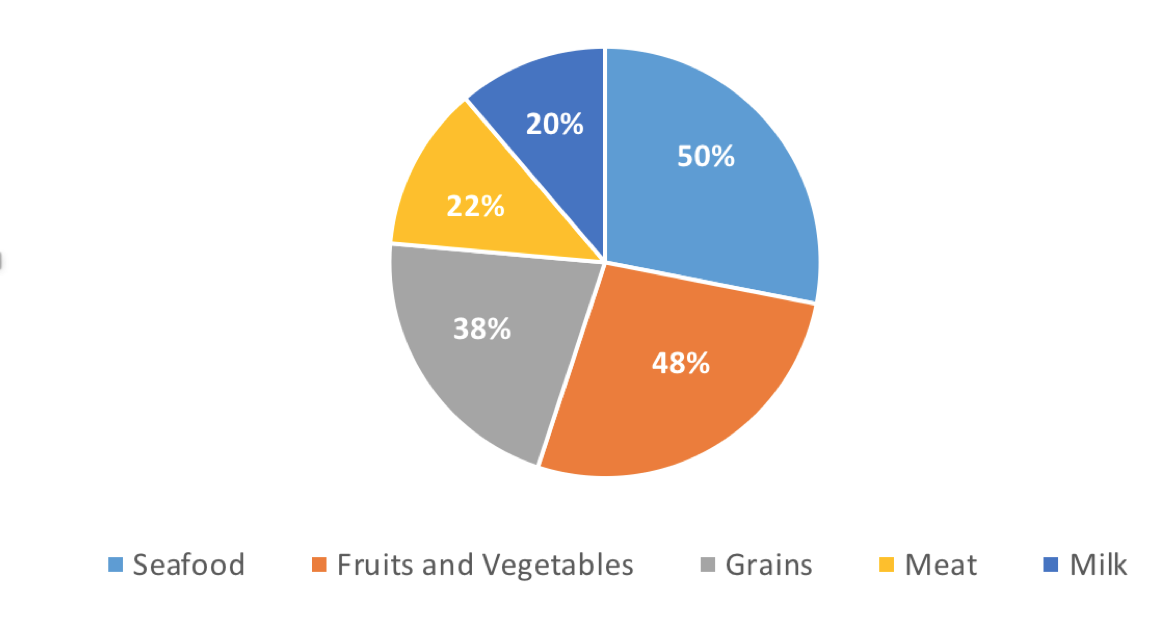
There was no dataset of the size we want, so we had to turn to web-scraping (such is life!) with keywords on google\_images\_downloader\_script to obtain images from Google. For the “Food” category, words like “dishes”, “cuisines” are used. For the “Food” category, words used include “Apple”, “Pulses” and “Tomato”.

We created a third category for fun to cater to the snapshots of excuse notes my kids will leave me to escape their chores. (Yup, inertia actually begins at a tender age.) For this category/class, we used words like “Eggs’ handwritten note”, “Chicken”, “Rajma Chawal” etc. to scrape google images.

**Fig 3.7 Recognition of dish**

**3.1.5 DATA PREPARATION**

After obtaining more than 3,000 images, it is important to ensure the images are correct. Going into each category, we eye-balled the pictures, removing i) conflicting pictures, ii) contrasting pictures showing before-and-after, iii) duplicates which won’t help in training, iv) vegetables pictures, v) Fruits images, vi) pictures of the Dishes, vii) Food Logos [11].



**Fig 3.8 Dataset Cleaning**

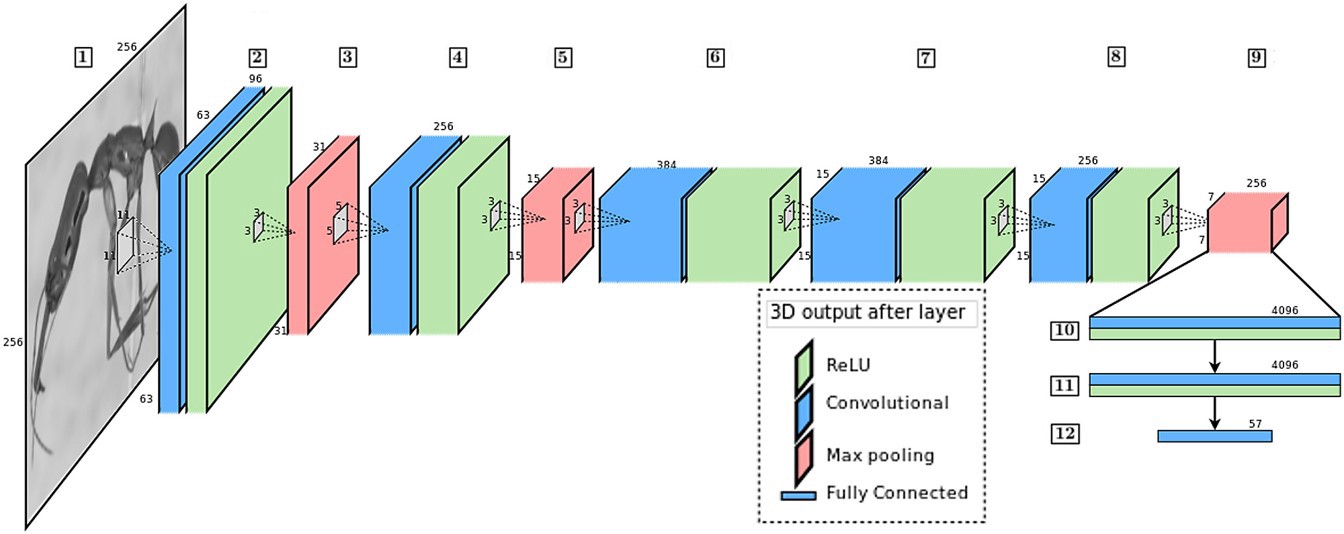
This dataset spring-cleaning is essential. Remember the principle: garbage in, garbage out.At the end of the day, we was left with more than 600 pictures for ”dishes” and “vegetables” and more than 900 for “fruits”.

**3.1.6 MODELS BUILDING**

Starting with machine-learning, we used different algorithms.

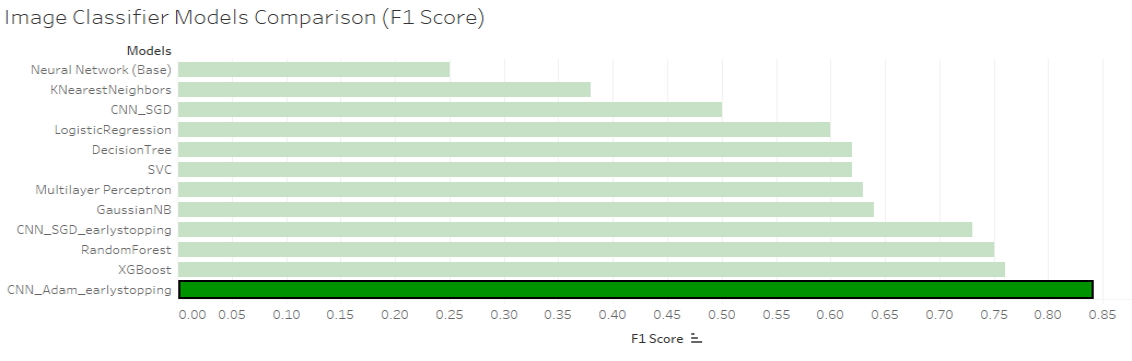
1. K-Nearest Neighbors
2. Gaussian Naive Bayes
3. Logistic Regression
4. Support Vector Classifier

Let’s explore Deep Learning using Convolutional Neural Network (CNN) which is reputable for working with images.



**Fig 3.9 CNN Architecture**

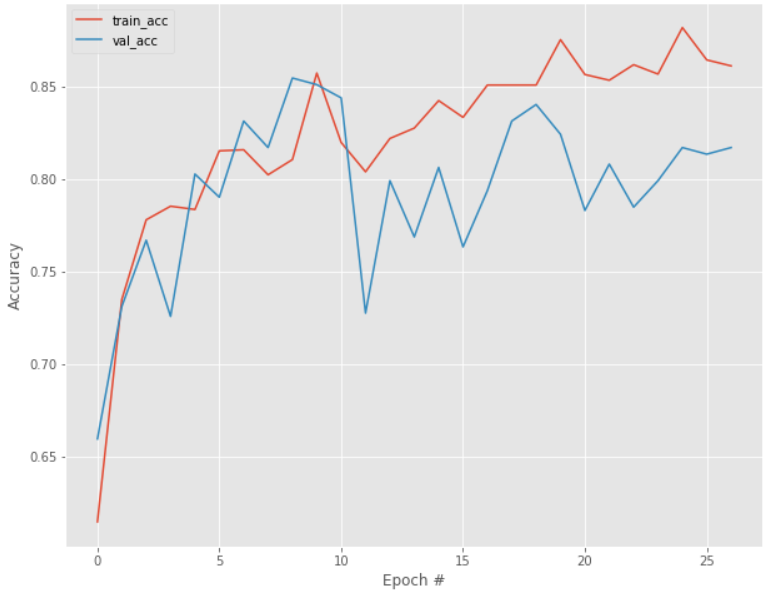
CNN Architecture: The CNN architecture comprises of many layers. Each layer will extract certain features (e.g. contrast, shapes, edges, texture) from the training pictures for each class. The trained model is then subsequently applied to unseen pictures which are then classified using the trained feature elements [11].



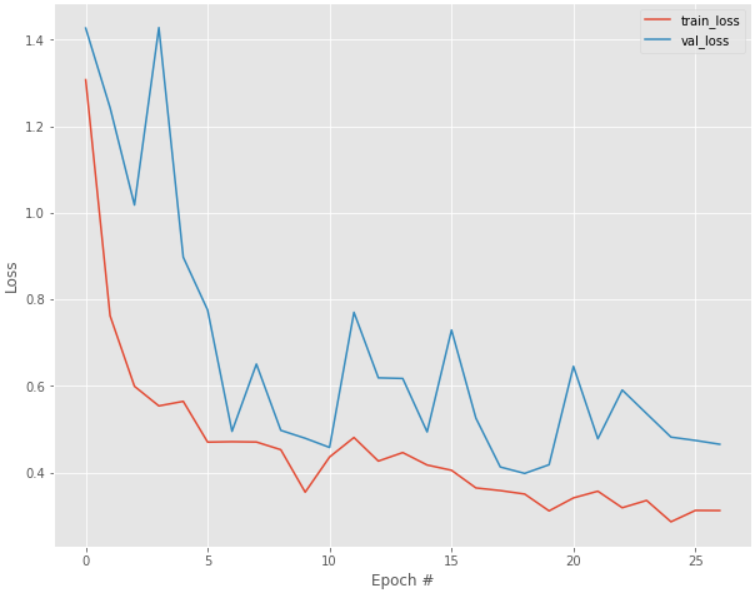
**Fig 3.10 Image Classifier Model Comparisons**

Comparison of Models with F1 score:

Indeed, the CNN model with the Adam optimizer and early stopping is the best model, beating the XGBoost with a new 0.84 F1 score associated with the trained network. The superiority of the Adam optimizer lies in its adaptive learning rate and favored due to its relatively less parameters tuning. Below are the codes for my model and classification report for the associated F1 score.



**Fig 3.11 Training and Validation accuracy with early stopping**

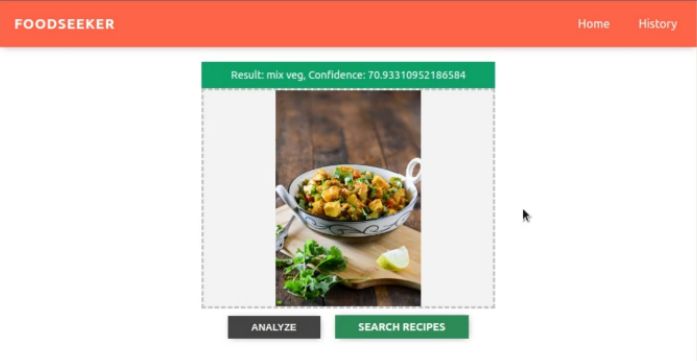


**Fig 3.12 Training and Validation loss with early stopping**

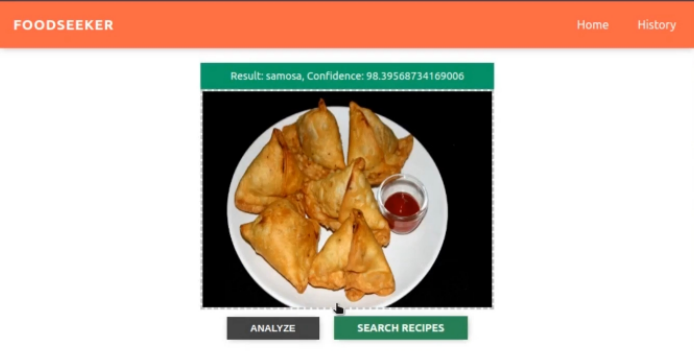
**3.1.7 Models Building**

Putting the model to the test with some unseen pictures, one from each class, here are the results:

**Fig 3.13 Analyzing of dish**



**Fig 3.14 Recognition of dish**



**Fig 3.15 Recognition of dish**

Image classification is a subset of image recognition which has widespread use in the security industry (facial recognition), virtual search engine (object finder in stores), healthcare (emotion detection in patients) and gaming and augmented reality. In a way, smartphone cameras have made all these advances possible with multitudes of pictures that can be easily created.

* 1. **MODULE 2**
     1. **Google Image Download Script**

By using the Google Images Downloader Script we use this to download photos from Google Images and creating the dataset of our own and train the data to recognize the food items that we take the picture of. With the help of this script we also take some erroneous images from google images therefore we also delete those images to improve the quality of the dataset and increase the accuracy to recognize the image.

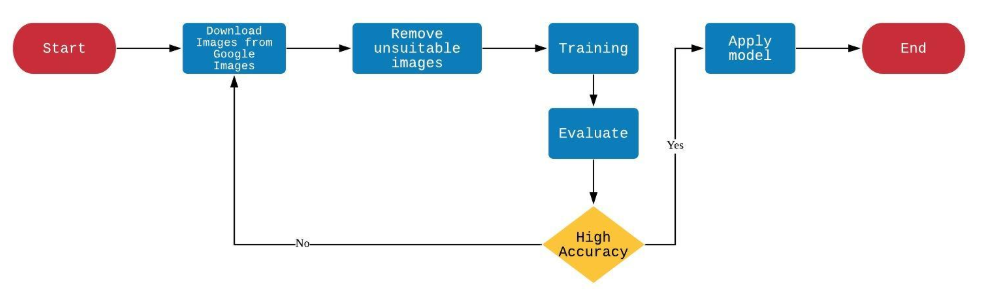
And after that we run train script to create the dataset so to recognize the food item in the picture that we click from our camera. After training the dataset again and again almost 4 million training steps are performed to increase the accuracy of recognition of the food items and after that training with such large number of times the accuracy of the system is increased to 78.8%. And to further improve the accuracy of the project we started collecting more and more photos that will enhance the accuracy of the system so for that we are starting collected about 1000 photos per food. After the training process is completed of the created database that we create for the process of recognition of food item. Then we export the TensorFlow models that will further integrate into Mobile App (Android App) that we created using the technology of the java.

And then we install TensorFlow Lite Library to Mobile Project of the dataset that we created to recognize the food item and our android application that installed in our mobile devices can use this model that we have trained via TensorFlowLite.In this, we would like to develop a collection of model that can use to recognize food and detect the food. In deep learning the most important part is the creation of dataset .So,in order to create dataset of North Indian food like Samosa, Khammam, Dhokla and South Indian food like Idli, Dosa I followed this wonderful post “How to create a deep learning dataset using Google Images” from none other than the OpenCV in which he has shown how through few lines of code you can download the bulk of images from Google within no time.

When I found an idea for this camp, I realized that food is an important part of country's culture so that every people should have knowledge about food of place where they stay at. Moreover, disable people, especially blind people also need help to know what food they eat. For foreigner, I will be a useful food's guideline at place their visit and also be an assistant to track what they eat and suggest what they should or should not eat.

* + 1. **Food Recognition**

It is a model to recognize food in a photo. We have downloaded above 600 photo per food on Google Image. After that, we have deleted unnecessary images and just keep correct images. We used these images to train the model.

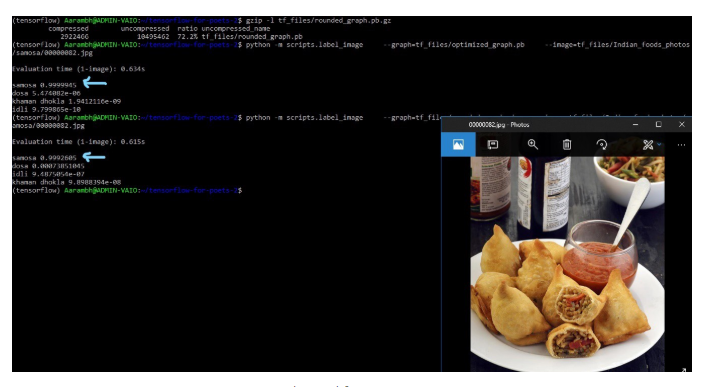


**Fig 3.16 Flow chart for creating dataset**

We were trying to build an Image classifier and then following the above mentioned post we started downloading the images of Idli, Samosa, Khammam, Dhokla and Dosa. We never restricted myself in terms of number of images in a particular folder related to a particular category we collected as much images as possible of every food category.



**Fig. 3.17 Images of few food items**

****

**Fig. 3.18 Recognition using Downloader Script**

* 1. **MODULE 3**
     1. **CREATING THE DATASET**

After obtaining more than 3,000 images, it is important to ensure the images are correct. Going into each category, we eye-balled the pictures, removing i) conflicting pictures, ii) contrasting pictures showing before-and-after, iii) duplicates which won’t help in training, iv) cleaning agents pictures.

Food datasets with images have proliferated in recent years. Although some datasets can be used to evaluate multiple tasks, we will roughly distinguish between three groups, according to the main task: general food recognition, recipe analysis/retrieval. Large datasets are mostly collected from data in the web, while smaller ones are often captured by the authors. General food recognition datasets typically consist of images and the corresponding (food) class labels, and they are mainly used to train food classifiers. These datasets have evolved to include progressively more food classes, from early datasets with a few number of cuisine-specific images to larger datasets that include a much larger number of images per class, and cover wider ranges of foods and cuisines. These larger datasets are very suitable for fine tuning deep CNNs leading to state-of-the-art food recognition.

Datasets for recipe analysis and retrieval incorporate ingredients and possibly other cooking information. The number of ingredients can vary from a few tens to several thousands. Other interesting attributes that some datasets include are course and cuisine types structured cooking instructions, cooking and cutting attributes and flavor attributes. They are typically used for cross-modal recipe retrieval, ingredient recognition and cuisine analysis. Training a large model requires a large amount of time. Since large dataset like Image Net, contains so many images, though we create our own dataset of Images using the script creating using Tensor flow as library and Python as the language.

Python is a multi-purpose language and widely used for scripting. We can write Python scripts to automate day-to-day things. Let’s say we want to download google images with multiple search queries. Instead of doing it manually we can automate the process. We use the dataset with large number of images which are small, clearly labelled and with no noise which make it ideal for the task and take considerably less pre-processing.

* + 1. **CHALLENGES FOR CREATING DATABASE**

The first goal is to be able to automatically classify an unknown image using the dataset, but beyond this there are a number of possibilities for looking at what regions / image components are important for making classifications, identify new types of food as combinations of existing tags, build object detectors which can find similar objects in a full scene. The challenge is to train models which can look at images of food items and detect the individual food items present in them. We use dataset of food images collected through the script creating using python and TensorFlow where numerous images of their daily food intake. This growing data set has been annotated - or automatic annotations have been verified - with respect to segmentation, classification, and weight / volume estimation. This is an evolving dataset, where we will release more data as the dataset grows over time.

* 1. **ANDROID**
     1. **ANDROID APPLICATION**

**Definition:**

Simply snap a food photo and get the information of your meal. Food-Seeker App is powered by our Food AI API. Food AI API is based on the latest innovations in deep learning and image classification technology to quickly and accurately identify food items.

**Conclusion:**

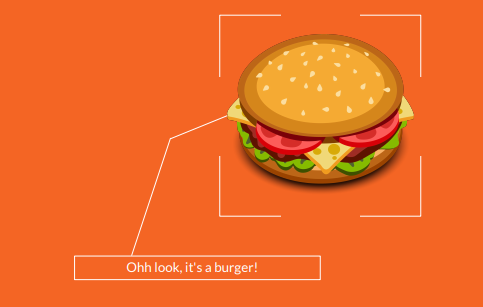
the accuracy of dietary assessment by analyzing the

food images captured by mobile devices

Substantial research has demonstrated that digital imaging accurately estimates dietary intake in many environments and it has many advantages over other methods. However, how to derive the food information (e.g., food type and portion size) from food image effectively and efficiently remains a challenging and open research problem.

We propose a new Convolutional Neural Network (CNN)-based food image recognition algorithm to address this problem. We applied our proposed approach to two real-world food image data sets and achieved impressive results. To the best of our knowledge, these results outperformed all other reported work using these two data sets. Our experiments have demonstrated that the proposed approach is a promising solution for addressing the food image recognition problem. Our future work includes further improving the performance of the algorithms and integrating our system into a real-world mobile and cloud computing-based system to enhance the accuracy of current measurements of dietary in-take. Since all these data were collected by food sharing websites, images do not contain any bounding box information indicating the food location.

Each image only contains the label information indicating the food type. Most of the images are popular western food images. In order to reduce bias and improve the accuracy of self-report, we proposed new algorithms to analyze the food images captured by mobile devices (e.g., smart-phone). The key technique innovation in this paper is the deep learning-based food image recognition algorithms. Our proposed algorithms are based on Convolutional Neural Network (CNN). Our experimental results on two challenging data sets using our proposed approach exceed the results from all existing approaches. In the future, we plan to improve performance of the algorithms and integrate our system into a realword mobile devices and cloud computing-based system to enhance the accuracy of current measurements of dietary caloric intake.



The most recent food recognition systems are developed based on deep convolutional neural network

architectures

The most recent food recognition systems are developed based on deep convolutional neural network

Archit

**Fig 3.19 Food recognition demo image**

**3.4.2 IMAGE UPLOADER**

**Definition:**

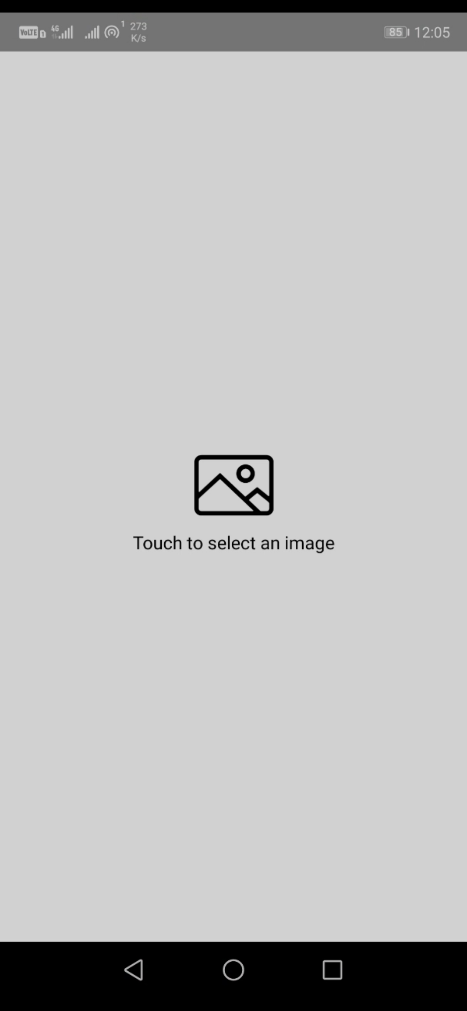
It is first step of the project or the front end of our project where you click an image and upload it for the further detection process. The results after taking food pictures will directly appear in the screen. The app has many areas open to extensibility, from utilizing more advanced machine learning models for food recognition, adding in recommendation systems, integrating better routing and price comparisons, and allowing user submitted recipes with voting and peer reviews.

**Background:**

The goal is to upload the image so that to recognize and get the recipe of the related image or further detection of the image. Android provides several options for you to save persistent application data. The solution you choose depends on your specific needs, such as whether the data should be private to your application or accessible to other applications (and the user) and how much space your data requires.

**Conclusion:**

The very first step is to get the image for further processing of the image and extracting the features from it. This method only works for a small number of food categories and is difficult to extend to composite or homemade food. The majority of their techniques for food recognition are based on traditional signal processing techniques with hand-engineered features. Recently, due to the occurrence of large annotated dataset like ImageNet, Microsoft COCO, and the development of powerful machine equipped with GPU, it is plausible to train large and complex CNN models for accurate recognition, which surpassed most of the methods adopted using hand-crafted features

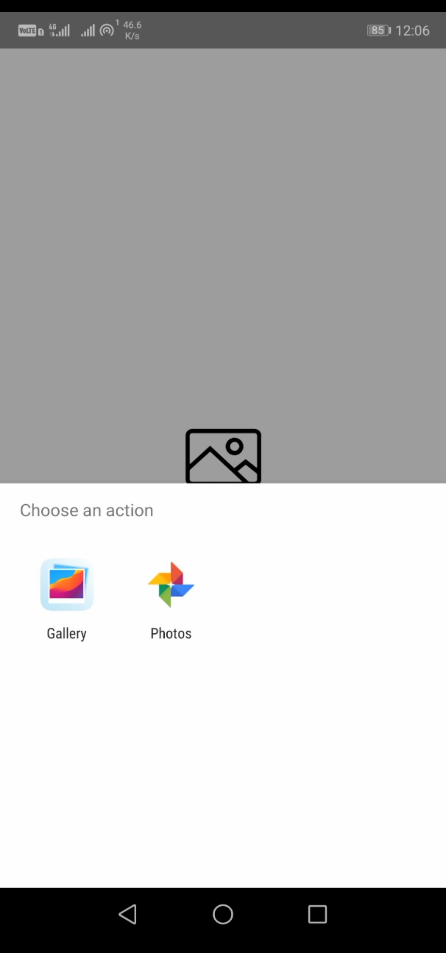
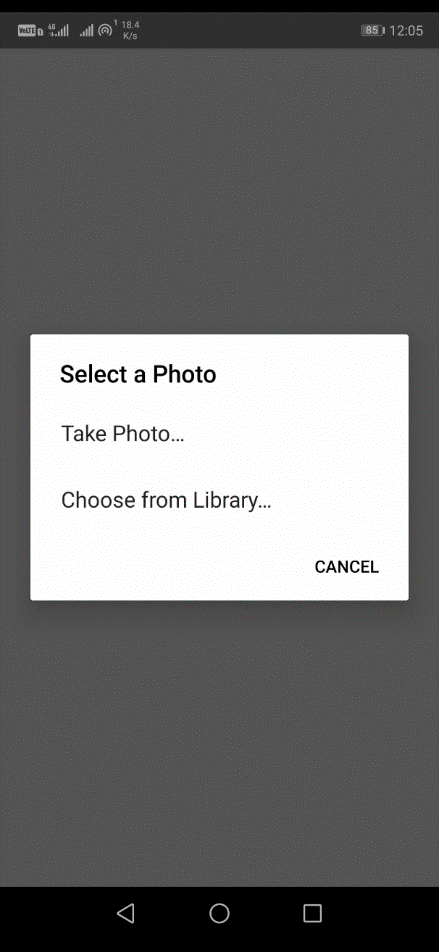


**Fig 3.20 Image Uploading in Android application**

**3.4.3 OPTIONS FOR SELECTING THE IMAGE**

**Definition:**

In this we will providing the ways to select the image for the purpose of image recognition and detection. The options availability will make it easier for the user to easily select the option that from where they want to select. As sometimes we went to the restaurant and we take the picture of that dish and forgot to recognize at that time or don’t have that enough time to recognize at the particular moment so that he can excess that make afterwards and recognize and get the related particular recipe. It will particularly work according for the efficiency of the user and give them leverage to recognize their favorite food item whenever and where ever they want.



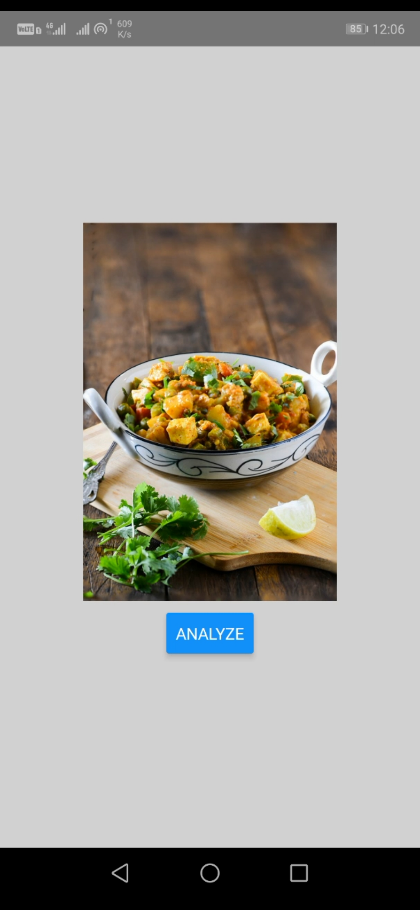
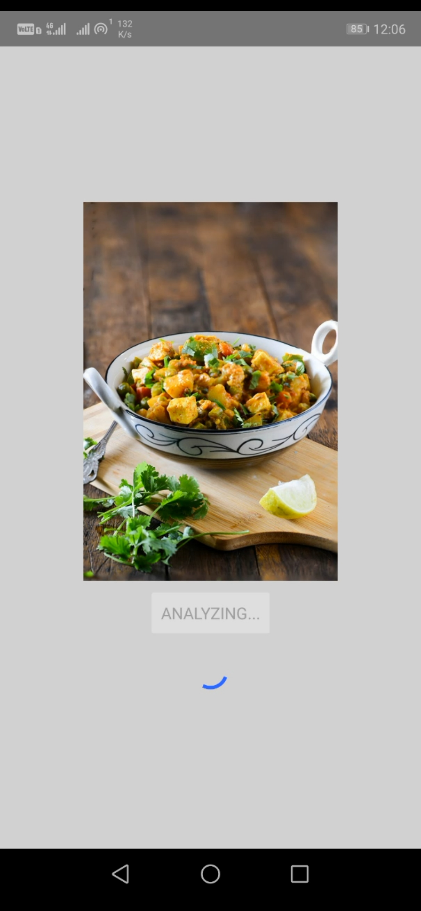
**Fig 3.21 Options for selecting image**

**3.4.4 GETTING FEATURES FROM THE IMAGE**

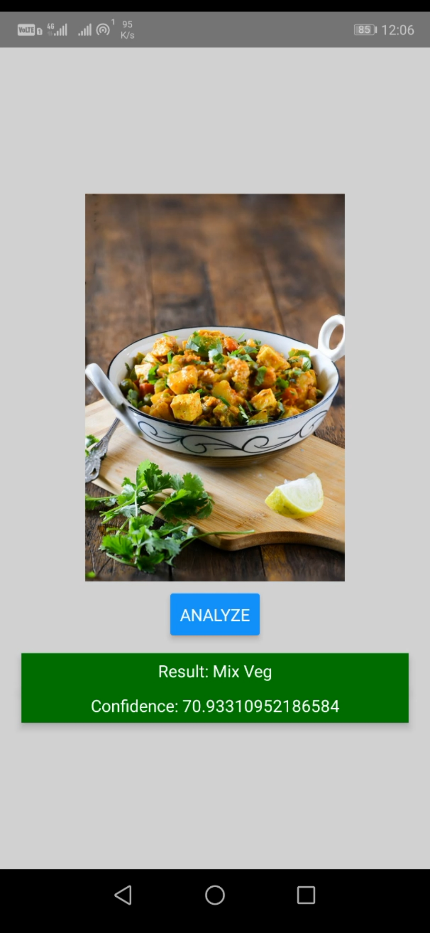
**Definition:**

In this we get the features or the information about the input images in the application. By this we get information regarding the food, we recognize the food item in the picture and get the respective recipe from it. From conventional image recognition methods, deep learning architectures like Convolutional Neural Networks (CNNs) can use the raw image as input as they incorporate the feature construction step directly into the learning process, by updating their parameters and connections as a function of the error on a set of training data .With special focus on recipe analysis and retrieval, food recommendation and restaurant context as emerging directions. The number of categories can explode since the name of dishes in a menu or in a recipe database can be very large. This increases the variability significantly, since the same dish can have very different appearances.

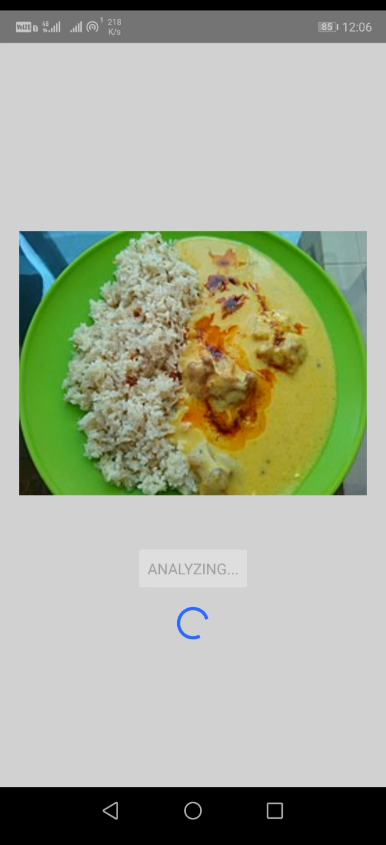
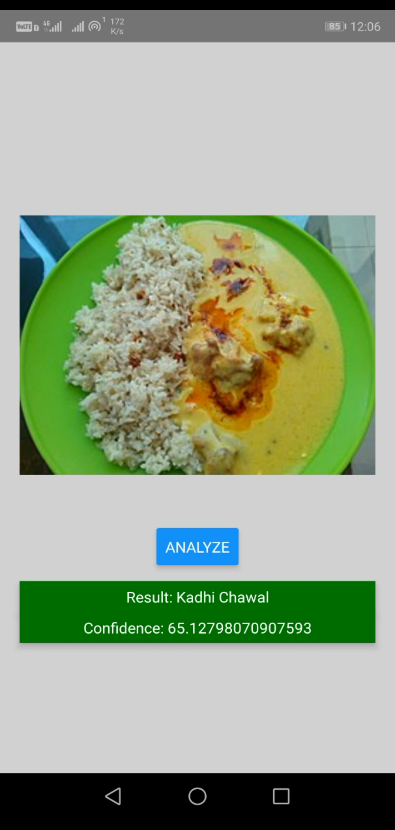
In this we get recognized image of the input image. The visual recognition paradigm changed rapidly after the appearance of the dataset, with more than one million images, demonstrating the power of data-driven feature learning in the form of deep CNNs. Since then, CNNs have also been the most effective architecture to address visual recognition, and food recognition in particular. Similarly, in the first works addressing food recognition, data consisted of a few categories in narrow domains (e.g., fast food). Recognition was address with handcrafted features, pooling and shallow classifiers such as support vector machines.



**Fig 3.22 Analyzing of food item**



**Fig 3.23 Recognition of dish**

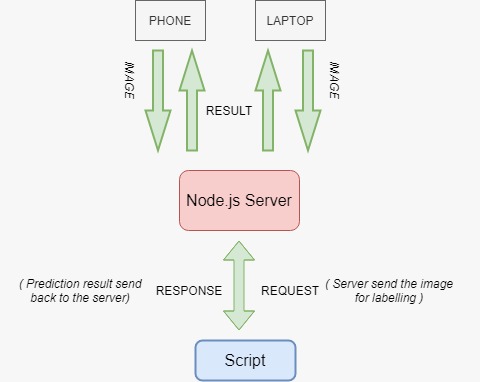


**Fig 3.24 Recognition of food item**

**3.5 FLOWCHART OF PROJECT**

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.

Following is the representation of our script that how the work is done. In this first we take the picture from the phone or any photo that is stored in your system then there are 2 types’ clients one is browser and second one is the mobile. Then our image will go to the NodeJS server that will act as the gateway for the image, then that NodeJS server will run the Python script and the test will perform on that image and then the python will return the response to the NodeJS server and response is then further send back to the client system. Below is the following flow chart:

****

**Fig 3.25 Flow chart for the script**

Now, the following flow graph show the complete process of the project. How each and every we will perform to get the desired output and perfect detection and recognition of the food item that user want to recognize and also the recipes of the related food item. Below is the flow chart of the complete project.



**Fig 3.26 Flow chart of complete project**

**CHAPTER-4**

**METHODOLOGY**

**4.1 PREFACE**

Hardly any other area affects human well-being to a similar extent as nutrition. Every day countless of food pictures are published from users on social networks; from the first home-made cake to the top Michelin dish, the joy is shared with you in case a dish was successfully cooked. It is a fact that no matter how different you may be from each other, good food is appreciated by everyone. Advances in the classification of individual cooking ingredients are sparse. The problem is that there are almost no public edited records available. This work deals with the problem of **automated recognition** of a photographed cooking dish and the subsequent output of the appropriate recipe. The distinction between the difficulty of the chosen problem and previous supervised classification problems is that there are large overlaps in food dishes (aka **high intra-class similarity**), as dishes of different categories may look very similar only in terms of image information.



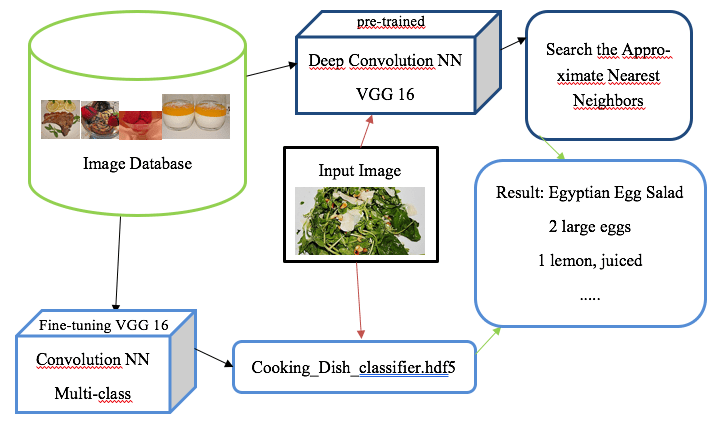
**Fig 4.1 Plate of Food**

The tutorial is subdivided into **smaller parts** in line with the motto divide and conquer: According to the current state, the largest German-language dataset of more than 300'000 recipes will be scraped and analyzed. Then, a newly developed method, according to the author’s knowledge, will be presented: the combination of object recognition or cooking court recognition using Convolutional Neural Networks (short CNN) and the search for the nearest neighbors (Next-Neighbor Classification) in a record of over 800,000 images. **This combination helps to find the correct recipe more likely,** as the top-5 categories of the CNN are compared to the next-neighbor category with ranked correlation. Rank correlation based approaches such as Kendall Tau essentially measure the probability of two items being in the same order in the two ranked lists. Mathematically, Kendall Tau is computed as:

https://miro.medium.com/max/251/1*4tAfykueT2XHPdeF6y-vMg.png (4.1)

Where, N = Total number of pairs C = Number of concordant pairs D = Number of discordant pairs. The exact pipeline looks like the following: (1) For every recipe W there are Knumber of pictures. For each of these images feature vectors are extracted from a pre-trained Convolution Neural Network trained on 1000 categories in the ILSVRC 2014 image recognition competition with millions of images. The feature vectors form an internal representation of the image in the last fully connected layer before the 1000-category Softmax Layer, which was removed beforehand. These feature vectors are then dimensionally reduced by PCA (Principal Component Analysis) from an N \* 4096 matrix to an N \* 512 matrix. As a result, one chooses the top 5 images with the smallest Euclidean distance to the input image, i.e. the top 5 optical, just from the picture information, similar pictures to the Input image.

1. Furthermore, a CNN is trained with **C** number of categories with pictures of **W** recipes. **C** has been **determined dynamically using topic modeling and semantic analysis of recipe names**. As a result we obtain for each category a probability to which the input image could belong.
2. The top-k categories from the CNN are compared with the categories from the top-k optically similar images with Kendall Tau correlation.



**Fig 4.2 Schema method**

**1 Data preparation**  
 1.1 Clearing data  
 1.2 Data augmentation

1. **Data analysis and visualization, split data (Train, Valid, Test)**
2. **Topic Modeling**  
   3.1 Latent Dirichlet Allocation (LDA)   
   3.2 Non-negative Matrix Factorization
3. **Feature Extraction**  
   4.1 k-nearest neighbor’s  
   4.2 t-SNE visualization
4. **Transfer Learning: Training pre-trained CNN (Convolutional Neural Network)**

**6 Deploying the data**.

**4.2. SCRAPING AND PREPARING THE DATA**

In order to be able to train a model at all, you need enough data (so-called data augmentation and fine-tuning of pre-trained models can be used as a remedy). Only because of this amount of data generalization of the training set can be continually increased to some degree and high accuracy can be achieved in a test set. The first part of this tutorial deals with the data acquisition, analysis and visualization of features and their relationships. Without exception, the quality and quantity of the data set is not negligible. More data leads to more dimensions, but more dimensions do not necessarily lead to a better model and its representation. Deviating patterns in the data set which disturb the learning can be unintentionally amplified by more dimensions, a generalization and learning of the data record is impaired for the neural network, the signal-to-noise ratio decreases. Without exception, the quality and quantity of the data set is not negligible. That’s why Europe’s biggest cooking platform will be scraped: each recipe, finally 316'756 recipes (as of December 2017), are downloaded with a total of 879'620 images. It is important not to proceed too fast when downloading and to protect the servers with too many queries, since otherwise a ban of the own IP address would make the data collection more difficult. More data leads to more dimensions, but more dimensions do not necessarily lead to a better model and its representation. Deviating patterns in the data set which disturb the learning can be unintentionally amplified by more dimensions, a generalization and learning of the data record is impaired for the neural network, the signal-to-noise ratio decreases.

A next important step is feature selection to disadvantage unimportant data. Preparing raw data for the neural net is commonplace in practice. In the first pass, the recipe name, the average application for the recipe, the number of ratings, the difficulty level, the preparation time and the publication date are downloaded. In the second pass, then the ingredient list, the recipe text, all images, and the number of times the recipe has been printed. With these features, the data record can be described very well and helps to gain a strong understanding of the data set, which is important to select the algorithms. Data such as recipe name, rating, date from the upload of the recipe, etc. are stored in a CSV file. If the recipe has an image, the thumbnail is placed in the search thumbnails folder. We will make usage of multiprocessing to ensure shorter download time. Next we need to scrape the list of places which serve the similar dishes.

**4.3 DATA ANALYSIS AND VISUALIZATION**

In order to get a first impression, we usually plot a heat-map to get first insights which possible features are interesting. The highest correlation have votes and average rating. Also interesting is the comparison between preparation time and number of ratings. Most reviews are based on recipes with short preparation time. Food datasets with images have proliferated in recent years. Table I shows a list of datasets reported in the literature and some of their characteristics. Although some datasets can be used to evaluate multiple tasks, we will roughly distinguish between three groups, according to the main task: general food recognition, recipe analysis/retrieval, and restaurant based recognition. Large datasets are mostly collected from data in the web, while smaller ones are often captured by the authors. General food recognition datasets typically consist of images and the corresponding (food) class labels, and they are mainly used to train food classifiers. These datasets have evolved to include progressively more food classes, from early datasets with a few number of cuisine-specific images to larger datasets that include a much larger number of images per class, and cover wider ranges of foods and cuisines. These larger datasets are very suitable for fine tuning deep CNNs leading to state-of-the-art food recognition.

* 1. **TOPIC MODELLING**

The goal of this procedure is to **divide all pictures/images into n-categories**. For a supervised classification problem, we have to provide the neural network with labeled images. It is only with these labels that learning becomes possible. The problem is that **Chefkoch.de does not categorize their pictures**. So we have to do this on our own. Topic modeling is a machine learning technique that automatically analyzes text data to determine cluster words for a set of documents. This is known as ‘unsupervised’ machine learning because it doesn’t require a predefined list of tags or training data that’s been previously classified by humans.

Since topic modeling doesn’t require training, it’s a quick and easy way to start analyzing your data. However, you can’t guarantee you’ll receive accurate results, which is why many businesses opt to invest time training a topic classification model.

Since topic classification models require training, they’re known as ‘supervised’ machine learning techniques. What does that mean? Well, as opposed to text modeling, topic classification needs to know the topics of a set of texts before analyzing them. Using these topics, data is tagged manually so that a topic classifier can learn and later make predictions by itself.It’s simple, really. Topic modeling involves counting words and grouping similar word patterns to infer topics within unstructured data. Let’s say you’re a software company and you want to know what customers are saying about particular features of your product. Instead of spending hours going through heaps of feedback, in an attempt to deduce which texts are talking about your topics of interest, you could analyze them with a topic modeling algorithm. By detecting patterns such as word frequency and distance between words, a topic model clusters feedback that is similar, and words and expressions that appear most often. With this information, you can quickly deduce what each set of texts are talking about. Remember, this approach is ‘unsupervised’ meaning that no training is required.

Topic modeling and topic classification do have one thing in common. They’re the most commonly used topic analysis techniques. Apart from that, they’re both very different and the one you choose, well, that depends on several factors. In theory, unsupervised machine learning algorithms such as topic modeling require less manual input than supervised algorithms. That’s because they don’t need to be trained by humans with manually tagged data. However, they do need high-quality data, and not only that – they need it in bucket loads, which may not always be easy to come by. At the end of your topic modeling analysis, you’ll receive collections of documents that the algorithm has grouped together, as well as clusters of words and expressions that it used to infer these relations.

* + 1. **LATENT SEMANTIC ANALYSIS (LSA)**

Latent Semantic Analysis (LSA) is one of the most frequent topic modeling methods analysts make use of. It is based on what is known as the [distributional hypothesis](https://en.wikipedia.org/wiki/Distributional_semantics) which states that the semantics of words can be grasped by looking at the contexts the words appear in. In other words, under this hypothesis, the semantics of two words will be similar if they tend to occur in similar contexts. That said, LSA computes how frequently words occur in the documents – and the whole corpus – and assumes that similar documents will contain approximately the same distribution of word frequencies for certain words. In this case, syntactic information (e.g. word order) and semantic information (e.g. the multiplicity of meanings of a given word) are ignored and each document is treated as a bag of words.

The standard method for computing word frequencies is what. This method computes frequencies by taking into consideration not only how frequent words are in a given document, but also how frequent words are in all the corpus of documents. Words with a higher frequency in the full corpus will be better candidates for document representations than less frequent words, regardless of how many times they appear in individual documents. As a result, representations are much better than those that only take into consideration word frequencies at document level.

**4.4.2 LATENT DIRICHLET ALLOCATION (LDA**)

LDA is a probability model which assumes that each name can be assigned to a topic. First, the name body must be cleaned, i.e. stop words are removed and words are reduced to their root. The clean vocabulary serves as input. Latent Dirichlet Allocation (LDA) and LSA are based on the same underlying assumptions: the distributional hypothesis, (i.e. similar topics make use of similar words) and the statistical mixture hypothesis (i.e. documents talk about several topics) for which a statistical distribution can be determined. The purpose of LDA is mapping each document in our corpus to a set of topics which covers a good deal of the words in the document.

What LDA does in order to map the documents to a list of topics is assign topics to arrangements of words, e.g. n-grams such as best player for a topic related to sports. This stems from the assumption that documents are written with arrangements of words and that those arrangements determine topics. Yet again, just like LSA, LDA also ignores syntactic information and treats documents as bags of words. It also assumes that all words in the document can be assigned a probability of belonging to a topic. That said, the goal of LDA is to determine the mixture of topics that a document contains.

a simila

* 1. **FEATURE EXTRACTION**

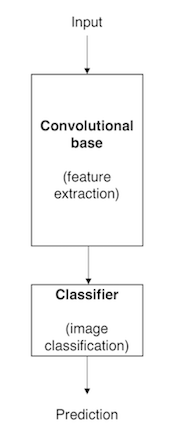
Decoupled from nature, neural networks work by reflecting the model of the human brain. The idea is that it learns from its mistakes, gradually adjusting the weights of the neuron to adapt to the data. With CNNs, the image information is first summarized to reduce the number of parameters. We assume that **the first layers in a CNN recognize rough structures in the picture. The further you proceed to the last Softmax layer, the finer the learned features become.** We can take advantage of this and takes pre-trained CNNs which have been trained with millions of pictures and remove the last layers to train them with our own data. This saves us millions of parameters and thus reduces computing time. The CNN chosen here is the VGG-16 which was trained in a classification competition 2014 on 1000 categories. If you remove the last layer, we get a feature extractor of the second-to-last layer. This forms a N x 4096 matrix, where n is the number of input pictures. We let the VGG-16 calculate the vector for every image we have. This vector is, so to speak, the **fingerprint of the picture:**an internal representation the neural network builds.

**4.6. TRANSFER LEARNING: TRAINING PRETRAINED CNN**

Several pre-trained models used in transfer learning are based on large convolutional neural networks (CNN) (Voulodimos et al. 2018). In general, CNN was shown to excel in a wide range of computer vision tasks (Bengio 2009). Its high performance and its easiness in training are two of the main factors driving the popularity of CNN over the last years.

A typical CNN has two parts:

Convolutional base, which is composed by a stack of convolutional and pooling layers. The main goal of the convolutional base is to generate features from the image. For an intuitive explanation of convolutional and pooling layers, please refer to Chollet (2017). Classifier, which is usually composed by fully connected layers. The main goal of the classifier is to classify the image based on the detected features. A fully connected layer is a layer whose neurons have full connections to all activation in the previous layer.



**Fig 4.3 Architecture of model based CNN**

One important aspect of these deep learning models is that they can automatically learn hierarchical feature representations. This means that features computed by the first layer are general and can be reused in different problem domains, while features computed by the last layer are specific and depend on the chosen dataset and task. According to Yosinski et al. (2014), ‘if first-layer features are general and last-layer features are specific, then there must be a transition from general to specific somewhere in the network’. As a result, the convolutional base of our CNN — especially its lower layers (those who are closer to the inputs) — refer to general features, whereas the classifier part, and some of the higher layers of the convolutional base, refer to specialized features.

**4.7 DEPLOYING THE FLASK**

A server-less application deployment. Flask one of the [two most popular](https://www.jetbrains.com/lp/devecosystem-2019/python/) Python development frameworks because of it's simplicity, extensibility, and community. For the same reasons, the [Server-less Framework](https://serverless.com/) is the most popular development tool for deploying server-less applications on Functions as a Service infrastructure like AWS Lambda. Flask is an un-opinionated web micro framework for Python. This means rather than providing a heap of built-in functionality it offers the extensibility of adding in different tools to it when appropriate.

The Server-less Framework can be used to deploy applications on several different platforms. The most common of these, and the focus of this guide, is AWS Lambda. In this guide, we'll use the Server-less Framework to deploy a Flask application as an AWS Lambda function sitting behind [Amazon API Gateway](https://serverless.com/amazon-api-gateway/).

**CHAPTER 5**

**CONCLUSION**

**5.1 CONCLUSION**

The integration of multimodal content, context and external knowledge helps human and machines to solve complex problems. In this spirit, we have described a general intelligent framework applied to food analysis, and reviewed some recent advances in several directions, including recipe analysis, food recommendation, restaurant oriented applications and related datasets. A future with even more pervasive intelligent and wearable devices, increasing obesity and cardiovascular diseases and increasing interest in discovering and understanding new foods and cuisines suggests research in this area will further develop. Nevertheless, there are still many open research problems and applications. While visual models have progressed significantly thanks to deep learning, multimodal representations and cross-modal alignments can still be improved. So can be the recognition of categories, ingredient and nutrients, and accurate the estimation of food and nutrients intake. More structured food-related knowledge such as knowledge graphs, together with delocalized, contextualized and personalized models can also be beneficial in challenging scenarios (e.g., restaurants). We hope this article helps provide an overview of recent research and open problems and outlines some potential research directions.

We investigated a large number of latest articles related to the APP of deep learning in food, described in detail the proposed structure, training methods, and the final evaluation result of DNNs used to process food image, spectrum, text, and other information in each surveyed article. In the aspect of performance, we compared the deep learning with other existing popular methods, and found that the deep learning method achieves better results than other methods in these reviewed studies. We concluded the advantages and disadvantages of deep learning methods and made a detailed discussion of the challenges and future perspective of deep learning in food domain. To authors’ knowledge, it is the first survey on the APPs of deep learning in the food domain. The purpose of this review is to encourage researchers and workers in this field to perform more experiments on food with deep learning methods, to present precise solutions for classification or regression problems and put them into practice for the benefits of food quality and safety inspection for human dietary health. At last, we recommend that (1) the combination of deep learning and multisource data fusion including RGB images, spectra, smell, taste, and so on, would be considered to make a more comprehensive assessment of food, the development of full‐automatic information acquisition equipment/systems with stable signal output for food and global food data sharing platforms should be studied in the future, since it is still very hard to obtain big data related to food due to the usage of semiautomatic or even manual information acquisition tools and incomplete data management and sharing platforms, (3) the potential of deep learning technology in data mining can be evaluated in food related areas rarely explored such as food sensory and consume, food supply chain, and so on, and (4) successful cases of deep learning such as in food (such as food image recognition, intelligent recipe recommendation APP, and fruit quality evaluation system) can be further transformed into practical products.

We have addressed the electiveness of CNNs for food image recognition and detection. First, we built a food image dataset from images uploaded by a large number of real users. Second, we applied CNN to the recognition of 10 food items and evaluated its performance. We found that CNN performed much better than did traditional methods using handcrafted features. Third, through observation of trained convolution kernels, we conﬁrmed that color features are essential to food image recognition. Fourth, we applied CNN to food detection, ﬁnding that CNN signiﬁcantly out-performed a baseline method.

**5.2 LIMITATIONS**

The main and basic limitation that is faced by every machine learner that uses machine learning algorithm for the purpose of detection and recognition is that any of the machine learning algorithm does not give 100% accuracy. Same as we face in our project that we can’t give the 100% accuracy using any of the machine learning algorithm.

Another limitation that, since we create our own data set that’s why is limited number of dishes to detect. We trained the model ourselves to recognize and detect the food items and give the corresponding recipe related to that food item but our detection and recognition is limited to few dishes. And even it is difficult to train and detect such huge number of data as there is availability of such large variety of cuisines.

**5.3 Future Scope**

In the future we try to improve our project by adding few more features to it, right now we are recognising the food item that user want to recognise and after that we detect that particular recognized food and give the corresponding recipes to it.

After that we try we improve our system to detect even that ingredients that are used for that particular food item and where these items are made available to you at your nearest location. Also where these food items are available in which restaurant in case you don’t want to make it at your home. So you can get go and eat your delicious cuisines. [13]

And we further increase our dataset to the maximum number of cuisines so that it is not difficult for the user to detect and recognise the food item and get the recipes and nearest location where it is available. And we also work on the accuracy of our dataset so that user won’t get the false output. We also try to add food voice recording features so that it beneficial for that blinds also. And also add the dietary food recognition and some suggestions for the diet purpose food. [12][15]

And we further try to give the schema for restoring blurred images. This project will enhance in future by making it suitable for any type of image. Either coloured image black and white image, any real time image, and blurred image to restore and get the desired output without any hindrance. Thus making this more reliable and efficient to use by any user.

**REFERENCES**

[1] K. Aizawa, K. Maeda, M. Ogawa, Y. Sato, M. Kasamatsu, K. Waki, and H. Takimoto: A smartphone-based food recording tool assisted by image retrieval. Journal of Diabetes Science and Technology, 8(2):203–208, 2014.

[2] H. Hoashi, T. Joutou, and K. Yanai. Image recognition of 85 food categories by feature fusion. In IEEE ISM, pages 296–301, 2010.

[3] F. Kong and J. Tan. Dietcam: Regular shape food recognition with a camera phone. In IEEE BSN, pages 127–132, 2011.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton. Image net classiﬁcation with deep convolutional neural networks. In NIPS, pages 1106–1114, 2012.

[5] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In ICML, 2012.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haﬀner. Gradient-based learning applied to food image recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[7] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classiﬁcation. In IEEE CVPR, pages 1794–1801. IEEE, 2009.

[8] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In IEEE CVPR, pages 2249–2256, 2010.

[9] F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, and E. J. Delp. The use of mobile device application android application. IEEE Journal of Selected Topics in Signal Processing, 4(4):756–766, 2010.

[13] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, “Food image

recognition with convolutional neural networks,” in 2015 IEEE 12th Intl

Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th

Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl

Conf on Scalable Computing and Communications and Its Associated

Workshops (UIC-ATC-ScalCom), 2015, pp. 690–693.

[14] K. Kitamura, T. Yamasaki, and K. Aizawa, “Foodlog: Capture, analysis

and retrieval of personal food images via web,” in Proceedings of the

ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating

Activities, 2009, pp. 23–30.

[15] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile

camera phones,” Pervasive and Mobile Computing, vol. 8, no. 1, pp.

147–163, 2012.

[16] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menu-

match: Restaurant-speciﬁc food logging from images,” in Applications

of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE,

2015, pp. 844–851.

[17] Y. Kawano and K. Yanai, “Foodcam: A real-time food recognition

system on a smartphone,” Multimedia Tools and Applications, pp. 1–

25, 2014.

[18] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp,

“The use of mobile devices in aiding dietary assessment and evaluation,”

Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 4, pp.

756–766, 2010.

[19] Z. Ahmad, N. Khanna, D. A. Kerr, C. J. Boushey, and E. J. Delp, “A

mobile phone user interface for image-based dietary assessment,” in

IS&T/SPIE Electronic Imaging. International Society for Optics and

Photonics, 2014, pp. 903 007–903 007.

[20] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition

and volume estimation of food intake using a mobile device,” in

Applications of Computer Vision (WACV), 2009 Workshop on, 2009,

pp. 1–8.

[21] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fern-

strom, J. P. DeLany, and R. Sclabassi, “Determination of food portion

size by image processing,” in Engineering in Medicine and Biology

Society, 2008. EMBS 2008. 30th Annual International Conference of

the IEEE, 2008, pp. 871–874.

[22] A. Mariappan, M. Bosch, F. Zhu, C. J. Boushey, D. A. Kerr, D. S. Ebert,

and E. J. Delp, “Personal dietary assessment using mobile devices,” vol.

7246, 2009, pp. 72 460Z–72 460Z–12.

[23] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, “Measuring

calorie and nutrition from food image,” Instrumentation and Measure-

ment, IEEE Transactions on, vol. 63, no. 8, pp. 1947–1956, 2014.

[24] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi,

“A novel svm based food recognition method for calorie measurement

applications,” in Multimedia and Expo Workshops (ICMEW), 2012 IEEE

International Conference on, 2012, pp. 495–498.

[13] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, “Food image

recognition with convolutional neural networks,” in 2015 IEEE 12th Intl

Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th

Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl

Conf on Scalable Computing and Communications and Its Associated

Workshops (UIC-ATC-ScalCom), 2015, pp. 690–693.

[14] K. Kitamura, T. Yamasaki, and K. Aizawa, “Foodlog: Capture, analysis

and retrieval of personal food images via web,” in Proceedings of the

ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating

Activities, 2009, pp. 23–30.

[15] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile

camera phones,” Pervasive and Mobile Computing, vol. 8, no. 1, pp.

147–163, 2012.

[16] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menu-

match: Restaurant-speciﬁc food logging from images,” in Applications

of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE,

2015, pp. 844–851.

[17] Y. Kawano and K. Yanai, “Foodcam: A real-time food recognition

system on a smartphone,” Multimedia Tools and Applications, pp. 1–

25, 2014.

[18] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp,

“The use of mobile devices in aiding dietary assessment and evaluation,”

Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 4, pp.

756–766, 2010.

[19] Z. Ahmad, N. Khanna, D. A. Kerr, C. J. Boushey, and E. J. Delp, “A

mobile phone user interface for image-based dietary assessment,” in

IS&T/SPIE Electronic Imaging. International Society for Optics and

Photonics, 2014, pp. 903 007–903 007.

[20] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition

and volume estimation of food intake using a mobile device,” in

Applications of Computer Vision (WACV), 2009 Workshop on, 2009,

pp. 1–8.

[21] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fern-

strom, J. P. DeLany, and R. Sclabassi, “Determination of food portion

size by image processing,” in Engineering in Medicine and Biology

Society, 2008. EMBS 2008. 30th Annual International Conference of

the IEEE, 2008, pp. 871–874.

[22] A. Mariappan, M. Bosch, F. Zhu, C. J. Boushey, D. A. Kerr, D. S. Ebert,

and E. J. Delp, “Personal dietary assessment using mobile devices,” vol.

7246, 2009, pp. 72 460Z–72 460Z–12.

[23] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, “Measuring

calorie and nutrition from food image,” Instrumentation and Measure-

ment, IEEE Transactions on, vol. 63, no. 8, pp. 1947–1956, 2014.

[24] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi,

“A novel svm based food recognition method for calorie measurement

applications,” in Multimedia and Expo Workshops (ICMEW), 2012 IEEE

International Conference on, 2012, pp. 495–498.

V [13] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, “Food image

recognition with convolutional neural networks,” in 2015 IEEE 12th Intl

Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th

Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl

Conf on Scalable Computing and Communications and Its Associated

Workshops (UIC-ATC-ScalCom), 2015, pp. 690–693.

[14] K. Kitamura, T. Yamasaki, and K. Aizawa, “Foodlog: Capture, analysis

and retrieval of personal food images via web,” in Proceedings of the

ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating

Activities, 2009, pp. 23–30.

[15] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile

camera phones,” Pervasive and Mobile Computing, vol. 8, no. 1, pp.

147–163, 2012.

[16] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menu-

match: Restaurant-speciﬁc food logging from images,” in Applications

of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE,

2015, pp. 844–851.

[17] Y. Kawano and K. Yanai, “Foodcam: A real-time food recognition

system on a smartphone,” Multimedia Tools and Applications, pp. 1–

25, 2014.

[18] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp,

“The use of mobile devices in aiding dietary assessment and evaluation,”

Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 4, pp.

756–766, 2010.

[19] Z. Ahmad, N. Khanna, D. A. Kerr, C. J. Boushey, and E. J. Delp, “A

mobile phone user interface for image-based dietary assessment,” in

IS&T/SPIE Electronic Imaging. International Society for Optics and

Photonics, 2014, pp. 903 007–903 007.

[20] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition

and volume estimation of food intake using a mobile device,” in

Applications of Computer Vision (WACV), 2009 Workshop on, 2009,

pp. 1–8.

[21] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fern-

strom, J. P. DeLany, and R. Sclabassi, “Determination of food portion

size by image processing,” in Engineering in Medicine and Biology

Society, 2008. EMBS 2008. 30th Annual International Conference of

the IEEE, 2008, pp. 871–874.

[22] A. Mariappan, M. Bosch, F. Zhu, C. J. Boushey, D. A. Kerr, D. S. Ebert,

and E. J. Delp, “Personal dietary assessment using mobile devices,” vol.

7246, 2009, pp. 72 460Z–72 460Z–12.

[23] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, “Measuring

calorie and nutrition from food image,” Instrumentation and Measure-

ment, IEEE Transactions on, vol. 63, no. 8, pp. 1947–1956, 2014.

[24] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi,

“A novel svm based food recognition method for calorie measurement

applications,” in Multimedia and Expo Workshops (ICMEW), 2012 IEEE

International Conference on, 2012, pp. 495–498.

[13] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, “Food image

recognition with convolutional neural networks,” in 2015 IEEE 12th Intl

Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th

Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl

Conf on Scalable Computing and Communications and Its Associated

Workshops (UIC-ATC-ScalCom), 2015, pp. 690–693.

[14] K. Kitamura, T. Yamasaki, and K. Aizawa, “Foodlog: Capture, analysis

and retrieval of personal food images via web,” in Proceedings of the

ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating

Activities, 2009, pp. 23–30.

[15] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile

camera phones,” Pervasive and Mobile Computing, vol. 8, no. 1, pp.

147–163, 2012.

[16] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menu-

match: Restaurant-speciﬁc food logging from images,” in Applications

of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE,

2015, pp. 844–851.

[17] Y. Kawano and K. Yanai, “Foodcam: A real-time food recognition

system on a smartphone,” Multimedia Tools and Applications, pp. 1–

25, 2014.

[18] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp,

“The use of mobile devices in aiding dietary assessment and evaluation,”

Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 4, pp.

756–766, 2010.

[19] Z. Ahmad, N. Khanna, D. A. Kerr, C. J. Boushey, and E. J. Delp, “A

mobile phone user interface for image-based dietary assessment,” in

IS&T/SPIE Electronic Imaging. International Society for Optics and

Photonics, 2014, pp. 903 007–903 007.

[20] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition

and volume estimation of food intake using a mobile device,” in

Applications of Computer Vision (WACV), 2009 Workshop on, 2009,

pp. 1–8.

[21] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fern-

strom, J. P. DeLany, and R. Sclabassi, “Determination of food portion

size by image processing,” in Engineering in Medicine and Biology

Society, 2008. EMBS 2008. 30th Annual International Conference of

the IEEE, 2008, pp. 871–874.

[22] A. Mariappan, M. Bosch, F. Zhu, C. J. Boushey, D. A. Kerr, D. S. Ebert,

and E. J. Delp, “Personal dietary assessment using mobile devices,” vol.

7246, 2009, pp. 72 460Z–72 460Z–12.

[23] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, “Measuring

calorie and nutrition from food image,” Instrumentation and Measure-

ment, IEEE Transactions on, vol. 63, no. 8, pp. 1947–1956, 2014.

[24] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi,

“A novel svm based food recognition method for calorie measurement

applications,” in Multimedia and Expo Workshops (ICMEW), 2012 IEEE

International Conference on, 2012, pp. 495–498.

[13] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, “Food image

recognition with convolutional neural networks,” in 2015 IEEE 12th Intl

Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th

Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl

Conf on Scalable Computing and Communications and Its Associated

Workshops (UIC-ATC-ScalCom), 2015, pp. 690–693.

[14] K. Kitamura, T. Yamasaki, and K. Aizawa, “Foodlog: Capture, analysis

and retrieval of personal food images via web,” in Proceedings of the

ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating

Activities, 2009, pp. 23–30.

[15] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile

camera phones,” Pervasive and Mobile Computing, vol. 8, no. 1, pp.

147–163, 2012.

[16] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menu-

match: Restaurant-speciﬁc food logging from images,” in Applications

of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE,

2015, pp. 844–851.

[17] Y. Kawano and K. Yanai, “Foodcam: A real-time food recognition

system on a smartphone,” Multimedia Tools and Applications, pp. 1–

25, 2014.

[18] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp,

“The use of mobile devices in aiding dietary assessment and evaluation,”

Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 4, pp.

756–766, 2010.

[19] Z. Ahmad, N. Khanna, D. A. Kerr, C. J. Boushey, and E. J. Delp, “A

mobile phone user interface for image-based dietary assessment,” in

IS&T/SPIE Electronic Imaging. International Society for Optics and

Photonics, 2014, pp. 903 007–903 007.

[20] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition

and volume estimation of food intake using a mobile device,” in

Applications of Computer Vision (WACV), 2009 Workshop on, 2009,

pp. 1–8.

[21] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fern-

strom, J. P. DeLany, and R. Sclabassi, “Determination of food portion

size by image processing,” in Engineering in Medicine and Biology

Society, 2008. EMBS 2008. 30th Annual International Conference of

the IEEE, 2008, pp. 871–874.

[22] A. Mariappan, M. Bosch, F. Zhu, C. J. Boushey, D. A. Kerr, D. S. Ebert,

and E. J. Delp, “Personal dietary assessment using mobile devices,” vol.

7246, 2009, pp. 72 460Z–72 460Z–12.

[23] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, “Measuring

calorie and nutrition from food image,” Instrumentation and Measure-

ment, IEEE Transactions on, vol. 63, no. 8, pp. 1947–1956, 2014.

[24] P. Pouladzadeh, G. Villalobos, R. Almaghrabi, and S. Shirmohammadi,

“A novel svm based food recognition method for calorie measurement

applications,” in Multimedia and Expo Workshops (ICMEW), 2012 IEEE

International Conference on, 2012, pp. 495–498.

[10] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, “Food image recognition with convolutional neural networks,” in 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015, pp. 690–693.

[11] K. Kitamura, T. Yamasaki, and K. Aizawa, “Foodlog: Capture, analysis and retrieval of personal food images via web,” in Proceedings of the ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities, 2009, pp. 23–30.

[12] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile camera phones,” Pervasive and Mobile Computing, vol. 8, no. 1, pp. 147–163, 2012.

[13] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, “Menumatch: Restaurant-specific food logging from images,” in Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on. IEEE, 2015, pp. 844–851.

[14] Y. Kawano and K. Yanai, “Foodcam: A real-time food recognition system on a smartphone,” Multimedia Tools and Applications, pp. 1– 25, 2014.

[15] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. Delp, “Stages for the purpose of image processing and recognition,” Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 4, pp. 756–766, 2010.

[16] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fernstrom, J. P. DeLany, and R. Sclabassi, “Determination of food portion size by image processing,” in Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, 2008, pp. 871–874.