Project and Professionalism (6CS020)

A2: Project Report

Restaurant app with food image classification

Student Id	: 1928193
Student Name	: Dikesh Rajbahak
Group	: C3G3
Supervisor	: Sagun Dhungana
Cohort	: 3
Submitted on	: 14-06-20

Declaration Sheet

Award Title: ***BSc (Hons) Computer Science***

(Award Title for your project, if in doubt refer to your course/Module Registration)

Declaration Sheet

(Presented in partial fulfillment of the assessment requirements for the above award.)

This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work. I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free license to do all or any of those things referred to in section 16(I) of the Copyright Designs and Patents Act 1988. (viz. to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make an adaptation of the work).

Student Name: Dikesh Rajbahak

Student ID Number: 1928193

Dikesh

Signature: …………………………………….……… Date: 13/06/2020

Abstract

Image classification is part of AI and is being used on daily basis from a normal person to industry level. Image classification is integrated in a restaurant webapp. Both aspect of this project, online food delivery and image classification is widely popular. In mid 1900s, scientists were learning the working of the human brain how it can classify object in quick glance and hoped to mimic the mechanism on the machine. They wanted to implement the mechanism of human vision in the computers, then started computer vision. The early phase of it was when "Lawrence Robert" who is also known as father of computer vision, mentioned the 3D features in a 2D image. An image was transformed and processed to find the edges. Image processing has been improving since then to this day with impressive results. Because of increase in data with large scale and improvement in hardware and neural network, image classification has good accuracy and outperform is certain conditions. Oppose to traditional neural networks, deep neural network has multiple hidden layers that performs different processing resulting in different perception for same data. CNN is a branch of deep neural network that has convolutional layer, non-linear layer, pooling layer and at last is fully connected layer. CNN is popular for its better performance in image processing and NLP. The advantage with the CNN is it prioritizes on main features and it can both learn and detect those features only without needing to go through all data. Being based on CNN, many architectures have been developed that includes Alexnet, VGGNet, etc. InceptionV3 is being used to server our purpose which uses 1×1 convolution that reduces computation and as a result the architecture can be increased in witdh and depth. Last layer of fully connected layer is replaced by global average pooling that produced better result. Softmax branches are used in multiple places in middle as intermediate. Dataset used for this project is "FOOD-101" which has 101 classes of food item, 1000 images in each class, 101000 images in total. The dataset includes wide variety of popular fast food items in USA. Because of the large volume of good quantity, the image classification is convincing.

Keywords: Image classification, Computer vision, Convolutional Neural Network, InceptionV3, deep neural network.

Acknowledgement

For this project to complete, many guidance was required and provided. I am fortunate enough to have assistance from many people. It was a great help to have advice and support when needed. I consider myself very lucky to be able to carry out this project with the help of everyone directly or indirectly involved. I consider this achievement of everyone who was able to help. I have deep gratitude toward Herald College, Kathmandu, University of Wolverhampton for letting me do this project for final year. This has helped me gain the knowledge in AI and computer vision which will surely be valuable for far future too.

I would further like to thank my supervision Mr. Sagun Dhungana for his remarkable assistance and dedication towards this project. This project wouldn’t have been possible without his presence and support. My deepest thanks for helping me improve this project.

To all my friends, family members and teachers who helped me in one way or another by sharing their time, knowledge, experience with moral, financial and physical support, thank you.

Contents

Chapter 1 Introduction 1
1.1 Academic questions 1
1.2 Aims and Objectives 1
1.2.1 Aims 1
1.2.2 Objectives 1
1.3 Proposed system 1
1.4 Statement of problem 2
1.5 Scope of study 2
1.6 Limitations 2
1.7 Introduction to Report Structure 3
Chapter 2 Literature review 4
2.1 Computer vision 4
2.2 History of Computer Vision 4
2.3 Deep Learning Neural Network 5
2.4 How does CNNs work? 6
2.5 Architecture in detail 7
2.5.1 Convolutional Layer 7
2.5.2 Nonlinear Layer 7
2.5.3 Pooling Layer 8
2.5.4 Fully connected (FC) layers 8
2.6 CNN based architecture 8
2.6.1 AlexNet (2012) 8
2.6.2 VGGNet (2014) 9
2.6.3 ResNet (2015) 9
2.6.4 GoogLeNet/Inception 10
2.7 Similar Project 10
2.8 Literature Summary 15
Chapter 3 Main content 16
3.1 Fact Finding Technique 16
3.1.1 Functional decomposition diagram 17
3.1.2 Business Process Modeling 17

3.2 System Requirement Specification (SRS) 20
3.3 Process diagram..... 22
3.3.1 Use Case Diagram..... 22
3.3.2 Class diagram 23
3.3.3 Sequence Diagram..... 24
3.3.4 Activity diagram 28
3.4 Database design..... 33
3.4.1 Entity Relation Diagram (ERD) 33
3.4.2 Data dictionary..... 34
3.5 User interface 36
3.5.1 Wireframe..... 36
3.6 System architecture 41
3.6.1 Server..... 41
3.6.2 Clients: 41
3.6.3 Network 41
3.6.4 Client/server architecture 41
3.6.5 System architecture setup..... 42
3.7 Development of the food image classification..... 42
3.8 Testing..... 48
3.8.1 Turing test..... 49
3.8.2 Black box testing for image classification..... 50
Chapter 4 Academic Question 51
Chapter 5 Conclusion 51
Chapter 6 Critical Evaluation 52
Chapter 7 Future Plan 52
Chapter 8 References..... 53
Chapter 9 Appendix..... 57

Figure 1 Camera Transformation (Roberts, 1963)..... 4
Figure 2 Google Lens identifying the objects on the picture and showing related items as result (Google, n.d.) 5
Figure 3 Simple Neural Network vs Deep Learning Neural Network (McGinty, 2018) 5
Figure 4 CNN architecture 6

Figure 5 Convolution Operation 7
Figure 6 Activation Function Cheat sheet (Sharma, 2017) 8
Figure 7 Architecture of AlexNet 9
Figure 8 Architecture of VGGNet 9
Figure 9 Inception Architecture 10
Figure 10 Image segmentation without pixel annotation 11
Figure 11. The architecture of a convolutional neural network model. 13
Figure 12 Functional Decomposition Diagram 17
Figure 13 View products by customer 17
Figure 14 Add/remove products.. 18
Figure 15 Add products to cart 18
Figure 16 Classify food image 19
Figure 17 Use case diagram of a restaurant website... 22
Figure 18 Class diagram of restaurant website 23
Figure 19 sequence diagram to view products 24
Figure 20 Sequence diagram to add products in cart 25
Figure 21 Sequence diagram to add products... 25
Figure 22 Sequence diagram to delete products... 26
Figure 23 Sequence diagram to edit product details... 26
Figure 24 Sequence diagram to classify image data 27
Figure 25 Activity diagram of customer in home page 28
Figure 26 Activity diagram of admin login 29
Figure 27 Activity diagram to add to cart.. 30
Figure 28 Activity diagram to classify an image 31
Figure 29 Activity diagram to add, edit or remove product item... 32
Figure 30 Entity Relation Diagram 33
Figure 31 Wireframe of homepage.. 36
Figure 32 Wireframe of cart page 37
Figure 33 Wireframe of checkout page.. 37
Figure 34 Wireframe of classification page 38
Figure 35 Wireframe of classified image.. 38
Figure 36 Wireframe for admin login.. 39
Figure 37 Wireframe of admin panel 39
Figure 38 Wireframe to add a product 40
Figure 39 Wireframe to view list of available products 40
Figure 40 One random image from each class from “FOOD-101” dataset............................ 43
Figure 41 Importing necessary libraries 44
Figure 42 Dividing test data and train data 44
Figure 43 Method for training model 45
Figure 44 Model fitting.. 46
Figure 45 Predict function 46
Figure 46 Load saved model 47
Figure 47 Call predict function by passing image 47
Figure 48 Visualization od accuracy and loss 47

Figure 49 Making order with new user 48
Figure 50 New user created 48
Figure 51 Cart value updated 49
Figure 52 Cart gets updated 49
Figure 53 Classification done right... 50
Figure 54 Classification done wrong 50

Chapter 1 Introduction

Technology has a broad field of implementation. Jobs have been easier, simpler and faster because of the technology. Everything is controlled through tip of a finger. Home deliveries are preferred nowadays. It has been more convenient, secure, easy to use and affordable because of the competition. With the clear benefits, it has taken over the hotel and restaurant sectors too. Technologies like touch screen, PDA, wireless LAN etc. is already in use for quite a time. With the wide use of internet, online food ordering is popular nowadays. Through a website designed for food delivery, the business owner can extend the business to a wider customer online and not needing them to visit the store. On the other side, the customers save time not needing to go all the way to restaurant and faster delivery. The online food ordering system is adopted by most fast food restaurants in the western world. This system is also being popular in context of Nepal. There are services like foodmandu, bhojdeal, foodmario and others that are providing the food delivery services in the valley. Along with this, the online payments and electronic payments are also gaining popularity.

1.1 Academic questions

- How does CNN classify images, how is it better and which architecture is better?

1.2 Aims and Objectives

1.2.1 Aims

- To build a food ordering system for restaurant
- To implement food classification in the website
- Keep accurate record of the orders and delivery.

1.2.2 Objectives

- Creating CNN model and train food dataset
- Increasing the accuracy of the food classification
- Proper implementation of food classification in a restaurant app
- Simple design for easy ordering of food from the available menu

1.3 Proposed system

The system I propose is an online delivery system for a restaurant that delivers food items in the menu to the customers designated area. The advantage to this is that it eases up the ordering process to both customers and the restaurant. The restaurant can work efficiently even if there is no dining space. When the order is placed on the webpage that is designed, the details are stored in the database and then later retrieved by the staffs at the restaurant almost at real time. While ordering, customers are supposed to choose food items listed in the website and then provide the delivery details. This allows the restaurant employees to quickly go through the order details and produce necessary items accordingly.

1.4 Statement of problem

The industries are expanding faster than ever, and people seek for ease to purchase products online while maintaining the cost efficiency. The traditional method has been obsolete in recent days where vendors have to purchase the product then sell it to the end customer. Its more tasking and people tend to prefer online ordering system since the process has been more secure than the past. Food can be ordered through online without the need to visit the restaurant or the food vendors. So, wide range of publicity and advertisement is needed for this to work efficiently.

1.5 Scope of study

This project is designed to allow the customers to order food and get it delivered in designated address and also to avoid the long queues of the customers at the counter to get food and to reduce workload on the employees.

Following things are amongst the other topics that is discussed below:

1. About the restaurant
2. The services offered there
3. Online purchase
4. Types of food provided.

1.6 Limitations

- The dataset used in this project is Food-101 that includes the images of 101 food items. There are large varieties of food items around the world and it couldn't address all.
- The items include popular fast foods in western market. That would not be good fit in Asian market.
- The model sometimes classifies wrong food items with the ones which looks similar.

1.7 Introduction to Report Structure

The structure of the report can be mentioned as follow:

Introduction: This section briefly explains the overall project including aims and objectives.

Literature review: This section includes the main research on literature. The research includes journal papers, experiments, books, findings. It helps in better understanding of the project with the research that has already been done in the past and helps to plan our project accordingly.

Development: This section involves with the development of the main system. All the details of planning, designing and execution of the system is included.

System testing: This section aims to test the system for possible bugs and error so that it could be avoided before the deployment of the system or to make changes in the future. Many methods of testing like white box testing, black box testing, design testing and application testing for the proposed system.

Academic question: In this section, academic question is answered which in included in introduction section.

Conclusion: This section includes finding of the project and possible future improvements.

References: This section has the reference of research materials like books, journals, research papers, websites.

Chapter 2 Literature review

2.1 Computer vision

The idea is generated by how human mind can interpret the objects in 3D and can focus on a particular subject and determine its nature. People can understand a person's emotions just by looking at their facial expression. Research on human vision has been going on for a significant time now and their multiple theories has been published, but with the level of complexity of a human mind, its nowhere enough. The researcher and mathematicians are trying to replicate the human vision in computer vision. There are many implementations of computer vision in real life like: machine inspection, automotive system, surveillance, biometrics. These are usually industry level applications (Szeliski, 2011). From an engineering point of view, the goal is to make computer vision function like a human vision and also surpass in some cases (Huang, n.d.). The computation power seemed to be limited to perform such heavy processing until recently, when deep learning and neural networks were innovated to solve those problems. Because of that technology some aspects have surpassed human limit. With data generated every second, computer vision gets powerful and accurate (Mihajlovic, 2019).

2.2 History of Computer Vision

Lawrence Robert is recognized as father of computer vision because of "Machine Perception of Three-Dimensional Solids" which mentions about the 3D feature in a 2D pictured of a line drawing.

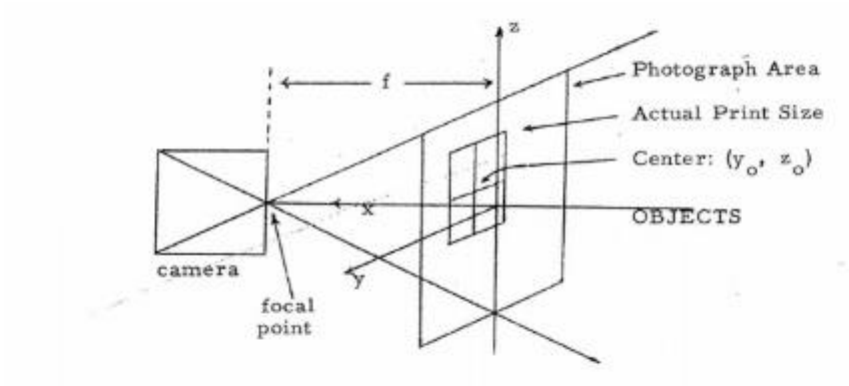


Figure 1 Camera Transformation (Roberts, 1963)

The image show how camera produce transformed image. The transformation of real-life picture is inversed (Zbigatron, 2018). Many researchers followed this work to study computer vision. Later in 1966, Marvin Minksy gave a summer project to a graduate student to connect a camera in a computer and figure out the picture produced. The achievement of that project is not publicly available, but it is an important milestone. In 1980, Kunihiko Fukshima introduced 'neocognitron' which was the base for the popular Convolutional Neural Network. In 2010, Google released Goggles which uses pictures taken in smartphones to recognize objects and search in Google. However, Goggles has discontinued on 2018 to be replaced with Google Lens which we use today that used Google Assistant. It can recognize object in a picture and show results related to the object or live translate language or explore nearby places. With the google assistant being popular, other smart assistants have been introduced like Siri from Apple, Alexa from Amazon, Cortana form Windows, Bixby from Samsung. In 2012, Google's neural network could recognize a cat with its deep neural network. In 2015, Google introduced tensorflow and made it open source. It has a huge library of data flow and programming support and heavily used in neural networks. In 2017, Apple released its flagship iPhone X with 3D face recognition being the only biometrics in the device. It uses dot projector, infrared camera and flood illuminator to me accurate in recognition and it even works in dark. In 2018, AI model created by Alibaba performed more accuracy in result of reading and comprehension test than human in Stanford University. The intel graphics chips has been improving significantly and being used for the intensive tasks of machine learning. (Verdict, 2020).

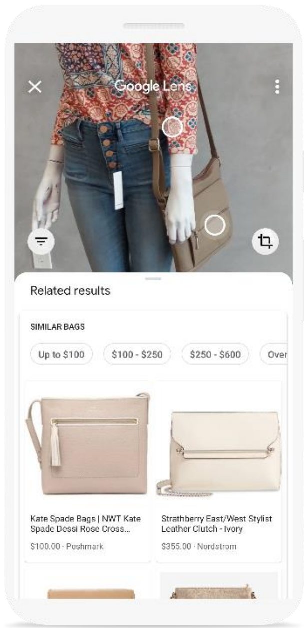


Figure 2 Google Lens identifying the objects on the picture and showing related items as result (Google, n.d.)

2.3 Deep Learning Neural Network

Considering the wide spread of Computer Vision, the mentioned achievements are mostly on consumers side. Fields like smart surveillance and monitoring, health services, sports, robotics, drones, self-driving cars have

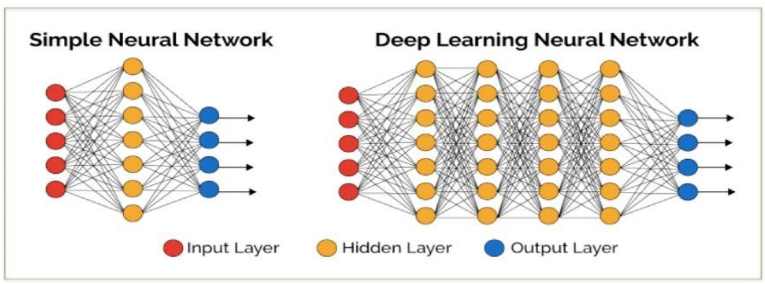


Figure 3 Simple Neural Network vs Deep Learning Neural Network (McGinty, 2018)

use case of computer vision in larger scale. The recent development in the computer vision is in the form of deep neural networks (Khan, et al., 2018). Deep learning is a branch of machine learning and an important part of computer vision. It has multiple layer of filtering the data (Arnold, et al., 2011). In recent years, image classification and object detection has improved drastically. The reasons are availability of larger dataset, powerful hardware and new algorithms and architectures. In the branch of DNN, lies CNN. CNN is like other neural networks and in addition the hidden layer consists of convolutional layers, pooling layers, fully connected layers and normalization layers (Nigam, 2018).

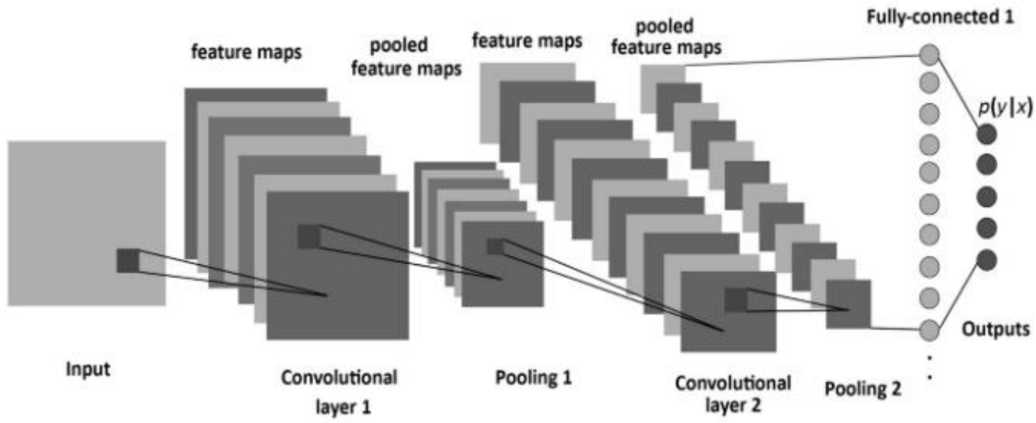


Figure 4 CNN architecture

2.4 How does CNNs work?

The Artificial Neural Network takes long time to train and as a replacement, CNN turned out better. It uses less parameters than the fully connected network. CNN can perform analysis in a specific area of an image rather than analyzing every pixel which is the way of fully connected network. With the less parameters and required fewer data, the CNN performs faster and better. The CNN is beneficial in case of image processing and NLP (Natural Language Processing). This project is based on image classification and CNN is trained for image classification and extract features from base level. It has variation of use cases like classify only focused feature like a human and ignore other objects like trees in a same image. It can perform thorough scan of a person's Facebook profile and analyze likes and style preference to help suggest advertisement according to the analysis (Missinglink, 2019).

2.5 Architecture in detail

There are different layers in CNN that performs specific task. The layers are summarized below:

2.5.1 Convolutional Layer

Convolutional layer used a smaller matrix which is called filter. The filter is then applied to over the image matrix and multiply. The multiplied value is summed up to one value. The matrix moves to right by a specified value. The filter is applied throughout the image matrix. The final produced output in much smaller than the original input (Upadhyay, 2019).

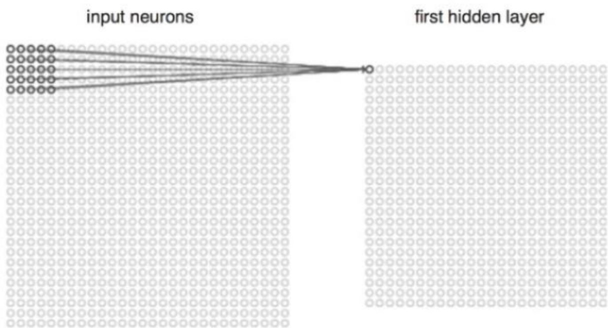


Figure 5 Convolution Operation

2.5.2 Nonlinear Layer

Nonlinear layer is an activation layer placed after each convolutional layer. It uses activation function to bring non-linearity in data. With nonlinearity, a complex mapping is done between the input and output of a network. This helps in learning and modeling complex data like image, audio, video. It helps of find pattern in data. Backpropagation is possible unlike with linear activation function. Some activation functions are Sigmoid, Tanh, ReLU, Leaky ReLU (Sharma, 2017).

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Figure 6 Activation Function Cheat sheet (Sharma, 2017)

2.5.3 Pooling Layer

Pooling layer is applied after a pooling layer. It works for further down sampling up to 75% and it also works to control overfitting. Because of down sampling, it also lessens computational power. There are several layer options for pooling layer which include:

- Max pooling : selects largest value in the matrix
- Min pooling : selects smallest value in the matrix
- Mean pooling : selects mean value of the sum of the matrix
- Average pooling: selects average value of the sum of the matrix (Deshpande, 2016)

2.5.4 Fully connected (FC) layers

Fully connected layers work to connect all the input layers connected to activation layers. It is usually placed in last few layers. In this layer, every neuron in one layer is connected to every layer in another layer. It is a time-consuming task, so it is not placed in every layer. Fully connected layer serves the purpose to classify the data in various classes (Singh, n.d.).

2.6 CNN based architecture

Many architectures for image classification are developed being based on CNN. Some important architectures are explained below:

2.6.1 AlexNet (2012)

AlexNet was the winner in 2012 ImageNet LSVRC-2012 competition. AlexNet uses ReLU activation function. It used overlapping polling and that's why the network in smaller in size. The layers include five convolutional layers and three fully connected layers and after every

Convolutional layer, ReLU activation is applied. There are 62.3 million parameters in the architecture that required 1.1 billion computation units (Krizhevsky, et al., 2017).

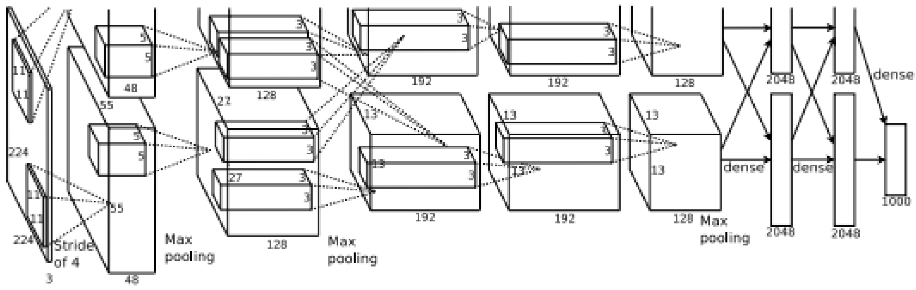


Figure 7 Architecture of AlexNet

2.6.2 VGGNet (2014)

VGGNet is the 1st runner-up in ILSVRC in 2014 which was invented in University of Oxford by VGG (Visual Geometry Group). The architecture has a uniform architecture with 13 convolutional and 3 fully connected layers using ReLU as activation layer. It works better for feature extraction in image data. 138M parameters are defined in the architecture that could take up space upto 500MB (Simonyan & Zisserman, 2015).

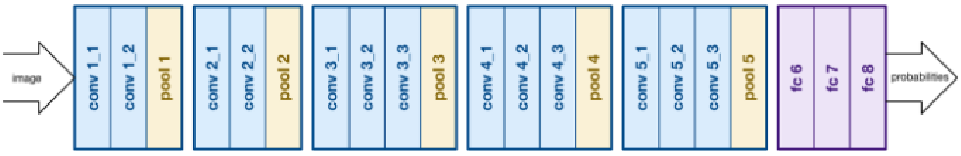


Figure 8 Architecture of VGGNet

2.6.3 ResNet (2015)

ResNet was introduced by Kaiming He and won the ILSVRC 2015 that outperformed human in a specific database. It features heavy batch normalization that used 152 layers. Moreover, it avoids vanishing gradients. The features of a image may degrade rapidly with high number of layers and cannot regain with backpropagation. So, this architecture prevents that from happening (Fung, 2017).

2.6.4 GoogLeNet/Inception

InceptionV3 which was primarily known as GoogLeNet is being used for this project. The GoogLeNet was submitted to ILSVRC 2014 that used 12 times less parameter that the architecture that won two years ago. GoogLeNet then won the ILSVRC 2014. It had significant improvement over the architectures that won the competition before 2014. With the naming, it is easy to say that it is from Google. With the additional versions, it is called Inception v1, and after that v2, v3 and v4. This architecture is quite different from others as it used 1×1 convolution in the middle and uses global average pooling instead of fully connected layers. Later the concept for Inception was for the architecture to go deeper rather than widening.

The features of Inception architecture include the 1×1 convolution that is used to reduce dimension in the module to lessen the computation. With the reduction, both depth and width can be increased. Looking after the prior architecture to the inception, they used fixed size for each layer of convolution but with this architecture, that changes. Inception uses different sizes of convolution along with different sizes of pooling which result is extraction of different kinds of features. For the replacement of fully connected (FC) layers, Inception used global average pooling at the end of the network. This resulted in the improved top-1 accuracy with 0.6%. The architecture goes 22 layers deep. Fig.9 shows the overall architecture of Inception and different softmax branches can be noticed in the middle as an intermediate. They are used only for training (Szegedy, et al., 2015).

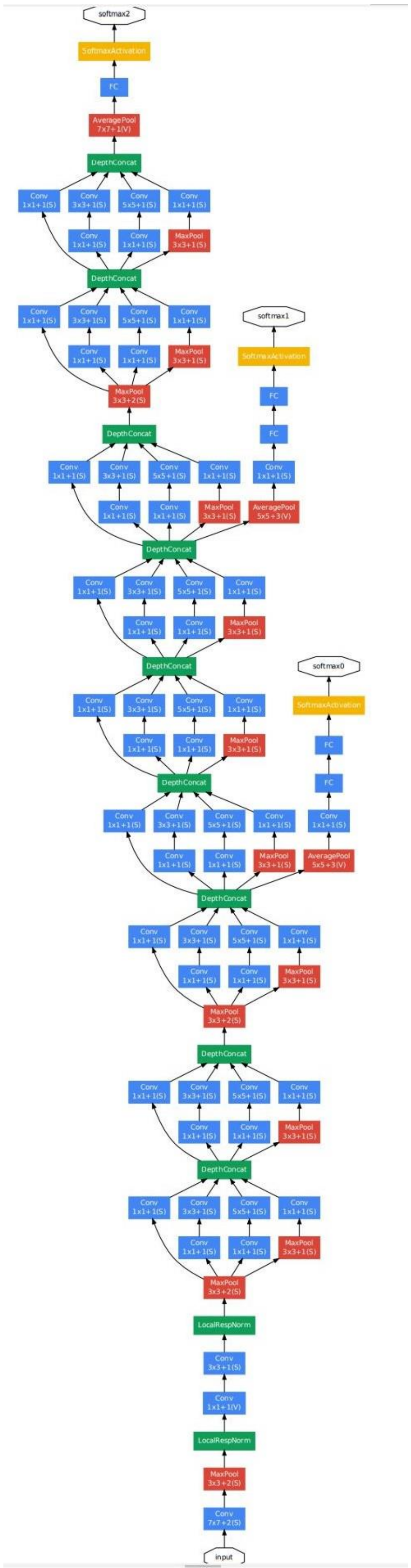


Figure 9 Inception Architecture

2.7 Similar Project

Food detection and recognition using Convolutional Neural Network

Food detection is a comparatively more difficult task because of wide varieties of food with inconsistent shapes, sizes and appearance. The proposed system uses Convolutional Neural Network to detect and recognize food items. The dataset used for the experiment has the food items that are commonly available. The idea is to detect the food and track the intake of nutrition that prioritizes on diet control. The image recognition can be alternative way to track, rather than keeping record manually. The images collected are identified with the region of food item and then isolated for dataset. Images in dataset is scaled to 80×80. Local Response Normalization is used after each pooling layer and after each normalization, a 6-fold validation was conducted. The accuracy of the model was 93.8% (Hokuto, et al., 2014).

CNN-Based Food Image Segmentation Without Pixel-Wise Annotation

The project is prioritized on DCNN which is proved to be effective for large-scale object recognition. The proposed system is a CNN-based food image segmentation that does not require pixel-wise annotation. Here, the selective search is applied by obtaining 2000 region boxes at most. Grouping is done for the obtained bounding boxes using clustering. Then the back propagation is performed over the pre-trained DCNN. Bounding box is based on the ratio of the intersection over union (IOU). There can be 20 bounding boxes groups at most and those which have less than 15 bounding boxes are removes. The selected groups are regarded as food region. Some bounding box groups can contain other bounding box groups inside them.

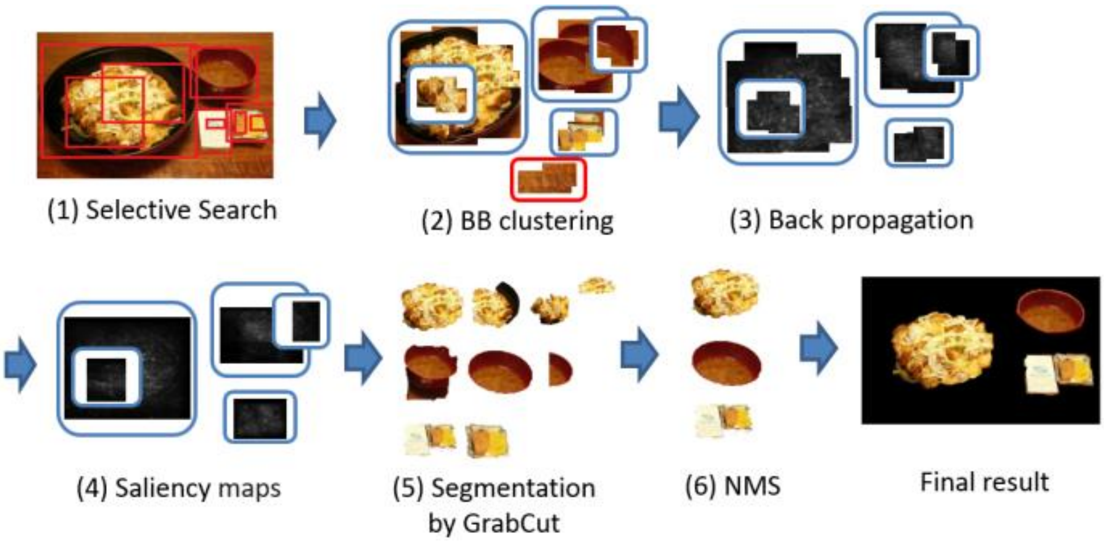


Figure 10 Image segmentation without pixel annotation

Then, using saliency maps, the rough position of the food regions is represented. It is done using back propagation on the pretrained DCNN. There is no need of pixel-wise annotation or bounding box annotation as part of training, the back propagation can be used for object region estimation. UEC-FOOD100 dataset is used in the experiment fitting with the AlexNet to estimate food categories and saliency maps. Then, with GrabCut, whole region around the food is extracted, because back propagation requires discriminative parts for the estimation. Finally, non-maximum suppression (NMS) is applied to obtain non-overlapped region. Each segmented region is provided with corresponding label so that they could be trained in CNN models (Shimoda & Yanai, 2015).

Simultaneous Estimation of Food Categories and Calories with Multi-task CNN

The paper is based on the multi-task CNN to determine the food categories and their corresponding calories. This proposal for this paper was to develop fully automatic system to classify the food items and also their corresponding calories, as there were no such applications for the CNN based where it could do all things automatically. There were applications that could classify the food items but to count the calories, those needed some variables as input from the user. Unlike that, this experiment used web image mining to create 15 category of calorie-annotated food images. The multi-task CNN is proposed over the single CNN because of its advantage to perform multiple tasks simultaneously. The applied multi-task CNN is based on VGG16 that simultaneously estimates food categories and calorie (Ege & Yanai, 2017).

CNN-based features for retrieval and classification of food images

In this paper, the deep Convolutional Neural Networks (CNNs) have been implemented as it can be more robust and expressive that the self-trained ones. It uses computer vision for food recognition with the help of food database 'Food-475' that includes 247,636 images of 475 classes. Different CNN architectures are analyzed and Residual Network (ResNet-50) turned out to perform the best. Fine-tuning with trained ResNet-50 is performed for food recognition. The result concludes that we need larger sets of data to increase the accuracy in food recognition (Aziz, et al., 2018).

CNN-based Features for Retrieval and Classification of Food Images

In this paper, a bunch of deep learning techniques are tested, that includes ResNet, Inception V3, MultiTask CNN in different databases like UECFOOD-100, UECFOOD-256, Food-101, Food-524, Food-475. One the other hand, it also tests the ResNet-50 performance drawn from the scratch. The ResNet-50 in comparison, was the best among other CNN architecture. Fine tuning with ResNet-50 are done from the scratch. It was tested with the available databases of food images. The feature extraction is done for the fine-tuned CNNs based on the classification and retrieval operation from UNICT-FD1200 and Food-475. The accuracy presented by the ResNet-50 is 95.45% whereas the same network implemented from scratch provides 91.01% accuracy (Ciocca, et al., 2017).

Food-101 – Mining Discriminative Components with Random Forests

The project focuses on the increasing interest of people in keeping track of their daily calories intake without having to consult with the medical expert. Random Forest (RF) method is proposed in order to carry out the task. It helps to mine the specific parts simultaneously from all the database. The dataset Food-101 is proposed of proposed to carry out the mining for food classification, that includes 101 food categories with a total of 101000 images. With the 50.76% of its accuracy, it performs better than the Fisher Vectors and other part mining algorithms by 11.88% and 8.13% respectively, but cannot outperform CNN (Bossard, et al., 2014).

Food Image Recognition by Using Convolutional Neural Networks (CNNs)

Convolutional neural network (CNN) have gained its popularity for image classification since it won the ImageNet Large-Scale Visual Recognition Challenge (ILLSVRC) in 2012. With the convolution layers and pooling layers, it can extract the features of a 2D image. In this experiment, the total number of 5822 images of 10 different classed of food item are collected from ImageNet, resolution fixed to 128*128. Bag-of-features model used along with Scale invariant feature transform to extract the features which is later processed with support vector machine. On the other hand, the CNN is applied with four hidden layers including three pooling layer and one fully connected layer.

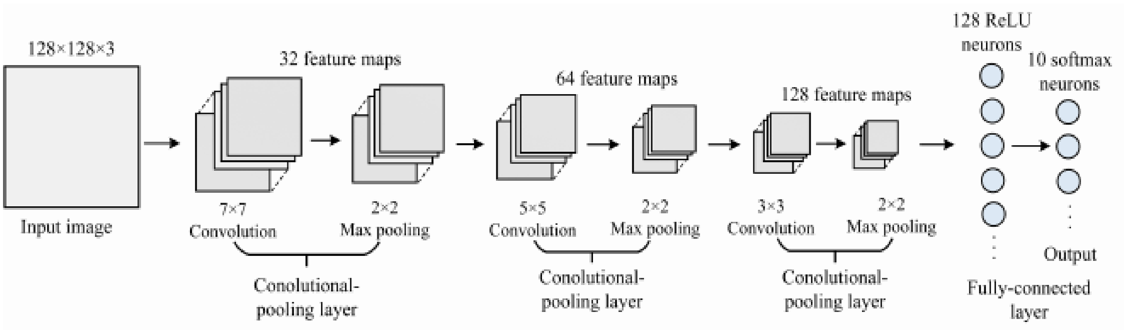


Figure 11. The architecture of a convolutional neural network model.

The CNN trained module showed 95% accuracy while Bag-of-features resulted 74% accuracy. Concluding, the CNN works better for image processing (Lu, 2019).

A Real-time Food Detection Mobile Application by Deep convolutional Neural Networks
With deep learning algorithm, the proposed system is smartphone based that used YOLO for live detection of food that is shown in bounding box with label. The model is well suited for even low-end devices and perform result in negligible time. The priority of this project is also to track calories in food item. While, others tend to detect single item in one image, they focus multiple food detection. This could be quite challenging just because there are numerous kinds of food depending upon the location. A deep convolutional neural network model is built upon Mobilenet, applied with the YOLOv2. Then, TensorFlow Java API is used to add-on with the mobile interface. Mobilenet used 3*3 filter with depth-wise separable convolutional splits. This can result in reduction in computational effort. Databases named UECFood100 that contains 100 categories of food items and UECFood256 which contains 256 categories of food items are used. Almost 40k images in total are used. There are 30 layers with 3.5 million parameters in the architecture if depth-wise and point wise layers are considered separate. And finally, for the nutrition analysis, NutritionixAPI that contains food information of more than 700k items with wide range. The DCNN model size is 8.1MB with the CPU time being 15ms and Wall clock time 75ms (Sun, et al., 2019).
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
Among many neural networks, MobileNets is considered to be a lightweight neural network. For this paper, the MobileNets network is tested among some applications including object detection. Two hyper parameters are introduced to define faster and efficient architecture. In real life scenario where augmented reality, self-driving car exist, speed can be a vital part. Large and heavy programs tend to be slower in speed. Standard convolutional filter is replaced with 1. Depthwise convolution and 2. Pointwise convolutional. The architecture use depthwise separable convolutions to apply single filter in each channel. This outputs two separate layers for filtering and adjoining. This results in less computation power and more speed in performance. Each layer of convolutional network is followed by ReLU and batchnorm. Softmax function is used for classification. On top of it, witdh multiplied is used to scale down the model even smaller. There is a tradeoff between speed and performance. The still performs better than GoogleNet despite being 2.5 times less computation (Howard, et al., 2017).

Xception: Deep Learning with Depthwise Separable Convolutions

The proposed Xception (stands for "Extreme Inception") model is modified form the classic Inception model to outperform the Inception model which was primarily built to work with ImageNet dataset. The Inception architecture was first introduced by Szegedy as GoogLeNet. Inception model is popular for its richer representation with less parameters. For this experiment, a map cross-channel will be used with a separate spatial correlation for every output channel. It will be completely based on depthwise separable convolutional layers with 36 convolutional layers subdivided into 14 modules. The dataset used for this experiment is JFT dataset that has large-scale image dataset with 17000 classes of 350 million high-resolution images. The model is compared to the Inception V3 model for having the similar scale. The top-5 accuracy for Inception V3 was 0.941 where as Xception was 0.945 (Chollet, 2017).

Exploring Better Food Detection via Transfer Learning

The experiment used transferred knowledge from pretrained food/non-food classification model. This improves the detection performance with better timing and quality. YOLOv2 framework is used for the application. And, three different neural networks are tested and find the quality of the transfer learning. It is considered as first approach to one-stage pipeline for in food detection field. The training is not required for the transfer learning as the features are already extracted in a pre-trained model. It results better because of its previous experiences and knowledge. The three neural networks used are MobileNet, MobileNetV2 and TLA-MobileNetV2 with the datasets UECFood100 where mean precision rates are 76.37%, 78.29% and 61.66% respectively whereas for UECFood256, mean precision rates are 75.01%, 76.01% and 54.77%. this concludes that small neural networks can also result good performance (Sun, et al., 2019).

Very Deep Convolutional Networks for Large-Scale Image Recognition

In this paper, a very small 3×3 convolutional filter is used for increasing depth of the architecture. It is based upon the ImageNet Challenge 2014. The input for the ConvNets is fixed to 224×224 RGB of image size. Group of multiple layers of convolution is followed by Fully Connected layers of three. It has five max-pooling layers of 2×2-pixel window. The final layer is the SoftMax layer. The training was done 370K iterations (74 epochs). The architecture resulted 7.0% test error which was better than of GoogLeNet with 7.9% test error (Simonyan & Zisserman, 2015).

Food Recognition for Dietary Assessment Using Deep Convolutional Neural Network

This paper is focused for the automated food intake assessment with the help of computer vision. Main motivation is to manage diet intake to prevent diet-related diseases. It uses 6-layer deep convolutional neural network to classify food image patches. The overlapping patches are considered food items. The used dataset consists of 573 food items. It resulted in the accuracy of 84.9% (Christodoulidis, et al., 2015).

Non-linear Convolution Filters for CNN-based Learning

Convolutional layers are typically linear systems. It was less expressive with functionalities. CNN has gained much popularity recently. Non-linear forms were experimented as it can result complex visualization of the data. Then the non-linearity has been used as activation functions and also many pooling strategies have been applied to bring non-linearity in the data. So, the CNN architecture is used with non-linear convolution. The datasets "CIFAR-10" and "CIFAR-100" are used and show better result with linear and non-linear filters outperforms the one with standard linear filter (Zoumpourlis, et al., 2017).

Calorie Mama

Calorie mama is a smartphone app that is available on Apple store and Google Store, which uses deep learning to track nutrition from food images. With a snap of food image, it can provide with the nutritional information about the meal. It is powered by food AI API. The food categories in this API includes global cuisine from Asia, Europe and South America. It improves as new food images are added to the database (Azumio, 2017).

2.8 Literature Summary

From all the research, it is clear that there are many ways to develop system. There are experiments that compares CNN with traditional ways like Fisher vector algorithm and Bag of Features. Then there is comparison between the architectures that are CNN based like ResNet, VGG-16 and those which are implemented from scratch. With all the research, it helped to decide the best option for our project.

Of course, it is clear that there is great importance of diet management in the society and more projects are leaning towards them. In addition, its server purpose in experimentation of the neural networks and other study and could also be implemented in system like the proposed one. There are variations in ways to develop the model and improve the performance. The project might seem small and simple, but it has a greater aspect in machine learning and AI. It is feasible to deploy a system like this in the mass like the Calorie mama which is popular for diet tracking.

Chapter 3 Main content

3.1 Fact Finding Technique

Fact finding can be important because it involves journals, research papers, questionnaires, interviews, prototyping, observations, experiments, etc. which is beneficial to carry out the project in quality manner. It is a very crucial part in System Development Life Cycle. It is a early stage process as it can provide the information how and about to carry out the project, what were the finding of the projects that were done in the past, what improvements can be made, which errors can be avoided. It would be easier to carry out the project and get enough information about the planning (UKESSAYS, 2016). The techniques we are going to user are project research and observation:

a. Research and site visit:

For a quality paper, it is must to have quality research. This involves gathering information from different journals, research papers, experiments, etc. with good citation and quality information. Some gathered information involves:

- Image Dataset: Datasets are hard to find, and it is ever more difficult incase of food dataset. There were multiple datasets having food images but with enough research, a usable dataset that was perfect for the project was found named FOOD-101.
- Types of CNN architecture: there are many award-winning CNN architectures available that includes LeNet-5, AlexNet, VGG-16, GoogLeNet, and for this project InceptionV3 (later version of GoogLeNet) was used.
- Programming language: Django framework of python was used for the project as it was best suited for AI based projects.

b. Observation:

Observation is another fact-finding technique that helped for this project. It involves learning about the business perspective of a website, how it handles customers, their requests and huge part of designing and efficiency. Also, the system analyst should have users' point of view to analyze the feasibility of the web interface. It also involves going through reports, documents of an existing system.

3.1.1 Functional decomposition diagram

The following figure shows functional decomposition diagram of the project that involves all the major functions and sub functions needs to available in the system.

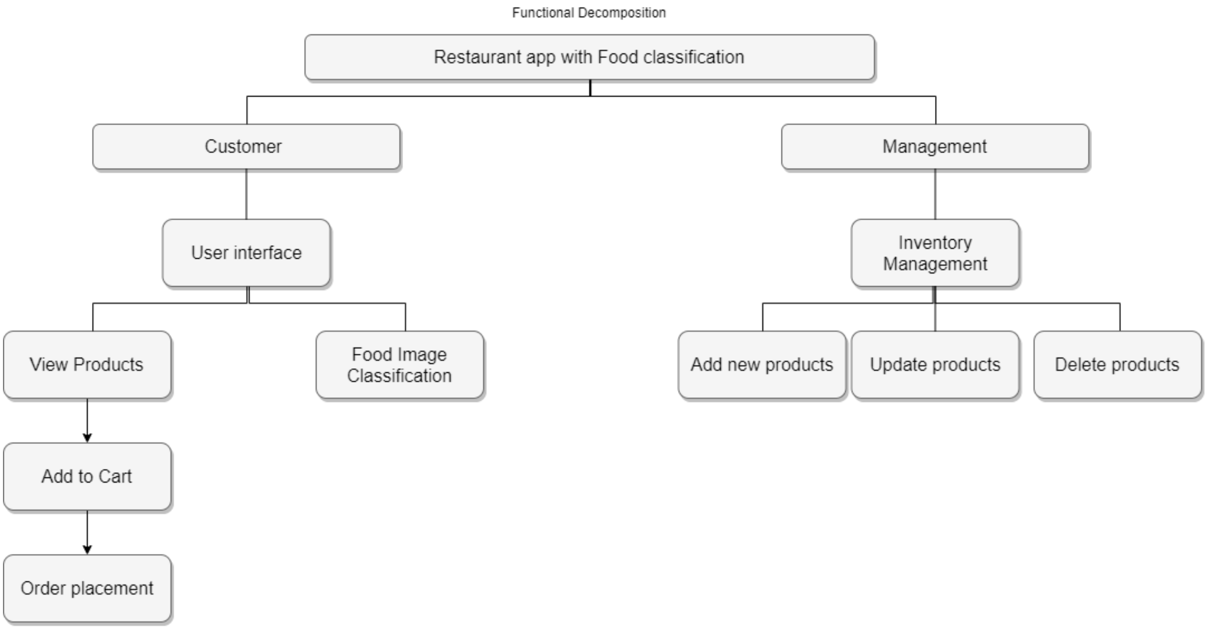


Figure 12 Functional Decomposition Diagram

3.1.2 Business Process Modeling

Business process modeling shows the workflow, or the processes carried out in a business. The following workflow show the better understanding of the system.

1. View products by customer

Customer is directly welcomed by the main page with the products.

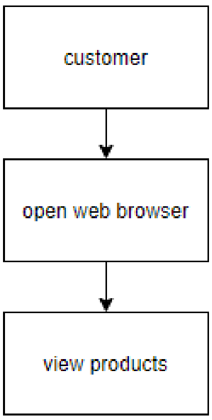


Figure 13 View products by customer

2. Add/remove products

Admin can login to the admin panel and add, remove or edit the product.

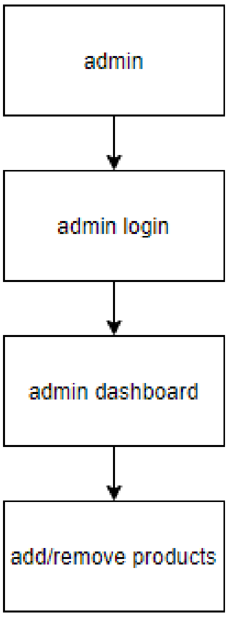


Figure 14 Add/remove products

3. Adding product to cart

Customer can view products in the main page and the add them to the cart then fill the form to confirm the order placement.

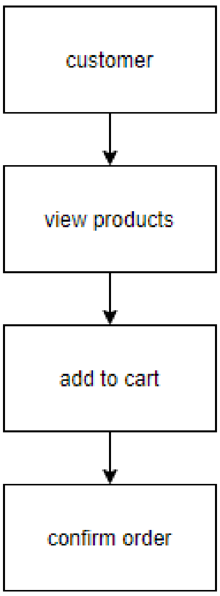


Figure 15 Add products to cart

4. Classify image to know about the food item

Customer can upload an image file in the designated place and then get a classified image as output.

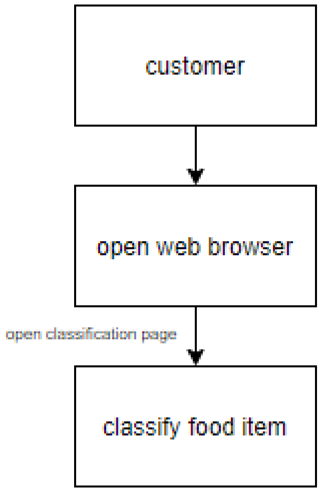
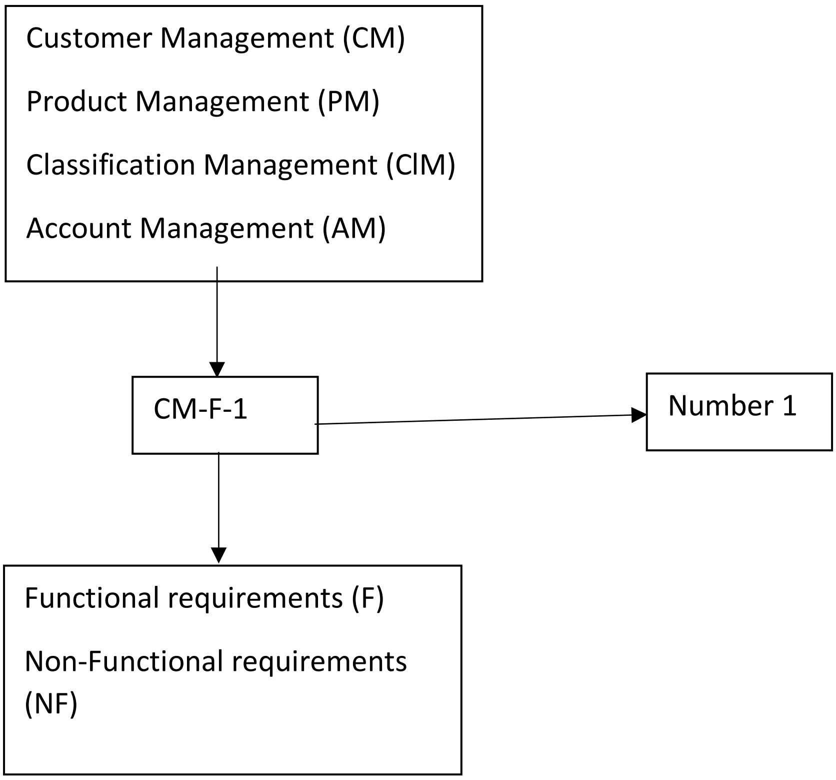


Figure 16 Classify food image

3.2 System Requirement Specification (SRS)



Requirement ID	Requirement Specification	Prioritization
CM-F-1.1	System must allow customers to view products	Must be
CM-NF-1.1.1	Anyone can visit the webpage without registration	Should be
CM-F-1.2	System must allow customers to add products in the cart	Must be
CM-NF-1.2.1	System must update the cart items every time new product id added	Must be
CM-F-1.3	System should show all the details of the products in add to cart page	Must be
CM-NF-1.3.1	System should use appropriate database to store all the information	Should be
CM-F-1.4	Customer should be able to increase or decrease order number from the cart page	Should be
CM-F-1.5	Customer must provide personal and shipping details while confirming order	Must be
PM-F-2.1	System must show all the items the database into the web browser	Must be
PM-NF-2.1.1	System should display the items in the home page	Should be
PM-F-2.2	System should allow admin to add a product	Must be
PM-NF-2.2.1	Admin has to enter all the details of the item	Should be

PM-F-2.3	System should allow admin to remove any item from database	Must be
PM-NF-2.3.1	System should ask confirmation before deleting	Could be
PM-F-2.4	System must allow admin to edit the items detail	Must be
PM-NF-2.4.1	All the details should be editable shown while editing item detail	Should be
ClM-F-3.1	Any customer should be allowed to use image classification	Should be
ClM-NF-3.1.1	Customer must upload an image file	Must be
ClM-NF-3.1.2	Customer should upload image of a food item	Should be
ClM-F-3.2	System must respond with a classified food name with the uploaded image	Must be
AM-F-4.1	Admin must provide correct username and password to login	Must be
AM-NF-4.1	Admin should know if username or password is wrong	Should be
AM-F-4.2	Only admin must be allowed to manipulate the products	Must be
AM-F-4.3	Admin must be allowed to view order details	Must be
AM-F-4.4	Admin must be allowed to view shipping details	Must be

3.3 Process diagram

3.3.1 Use Case Diagram

The following figure is the use case diagram for an ecommerce website of a restaurant with an integrated classification feature for food items. This includes customer and admin controls in the system. It shows the relation between system, actors and the use case of the application.

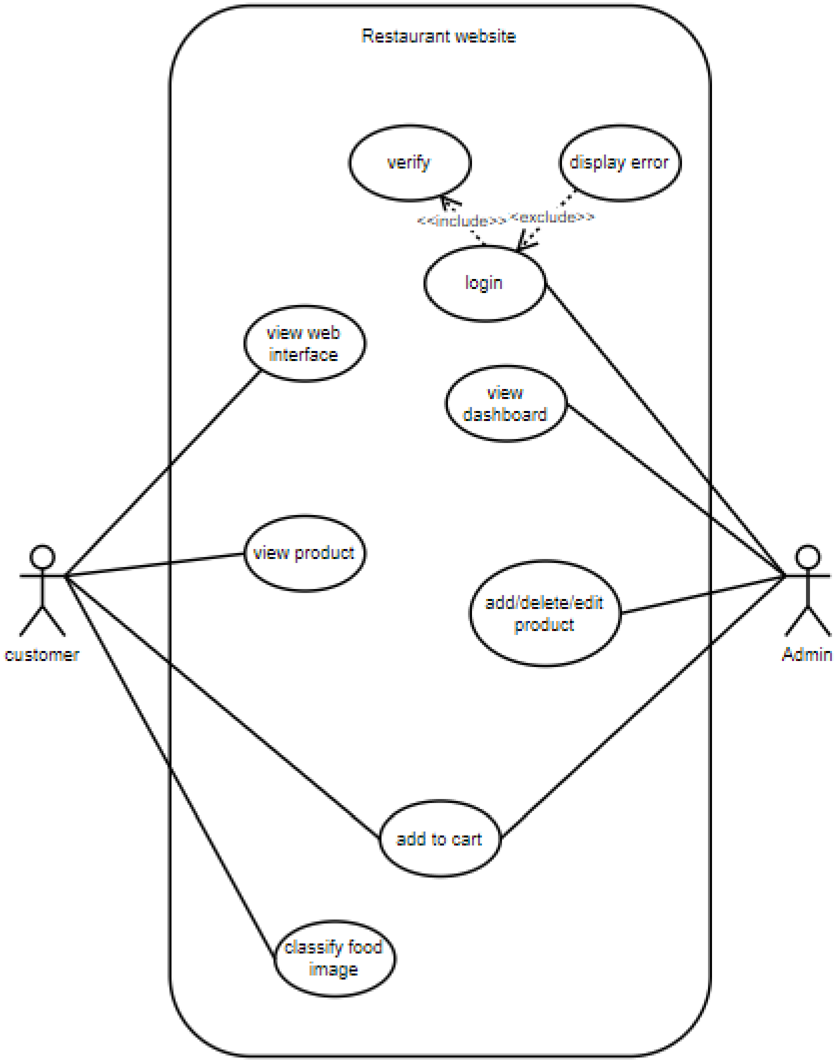


Figure 17 Use case diagram of a restaurant website

Use case body

Title: Restaurant website with image classification

Summary: The use case shows the relation of actors with the system and the interaction between them.

Actors: Customer, Admin

Event list:

Action of actors	Action of system
Customer opens web page	Show the home page with products listed
Customer adds product to the cart	System updates the cart data
Customer uploads image for classification	Systems shows the classified name with the same image
Admin logs in into the system	System displays admin panel
Admin add/remove/update products	System displays success message

3.3.2 Class diagram

The following diagram is the class diagram for the restaurant website. It includes classes, attributes and the relationship between the classes.

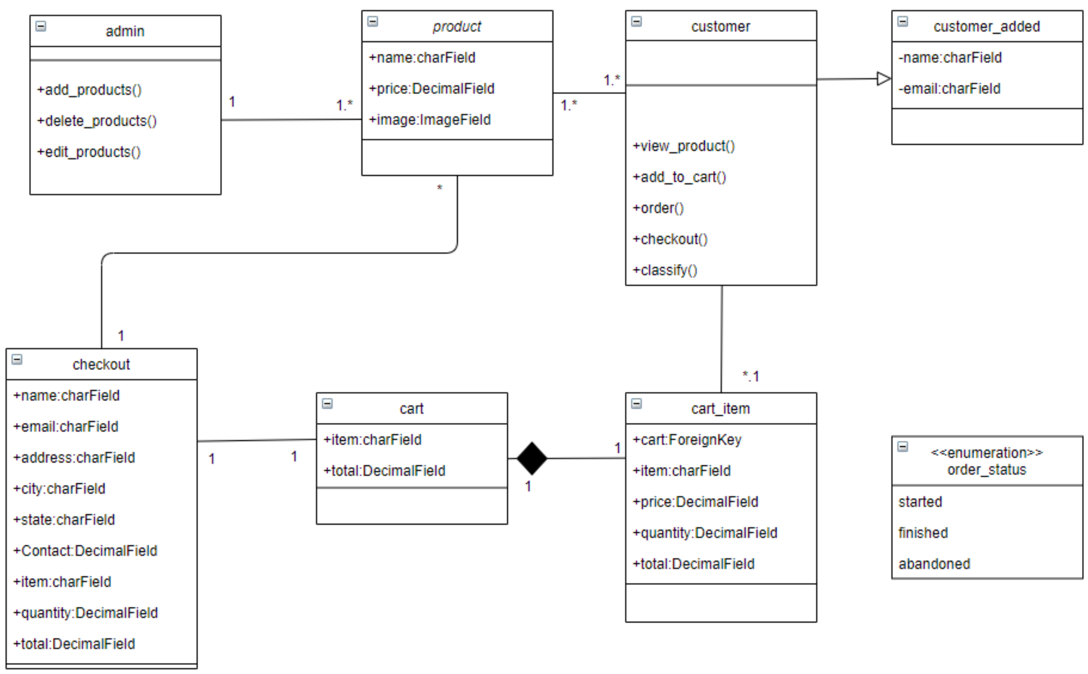


Figure 18 Class diagram of restaurant website

Here, the customer_added class is inheriting customer class because the customer is only added when he/she shall make a purchase. Cart and cart_item have a composition relationship whereas cart and checkout, product and customer, customer and cart_item have association relations.

3.3.3 Sequence Diagram

A sequence diagram show interaction between different objects in the system in a sequential order for caring out the operations. It shows the action is sequence with the input an output that involves actors and the system. It is time focused represents the time of action and messages.

1. System must allow customers of view products

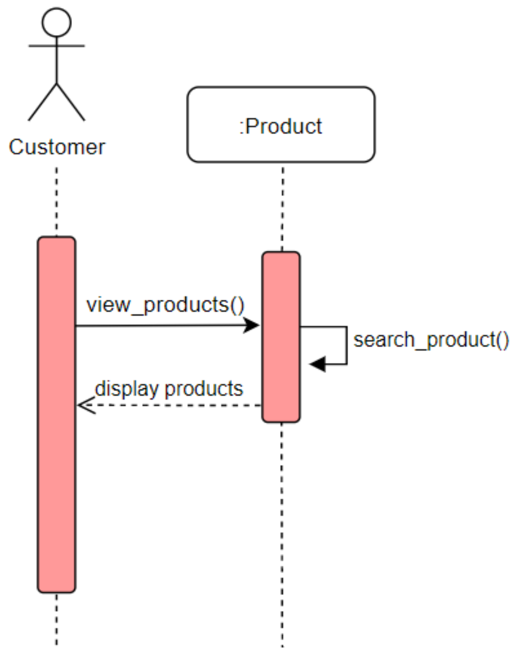


Figure 19 sequence diagram to view products

2. System must allow customers to add products in cart

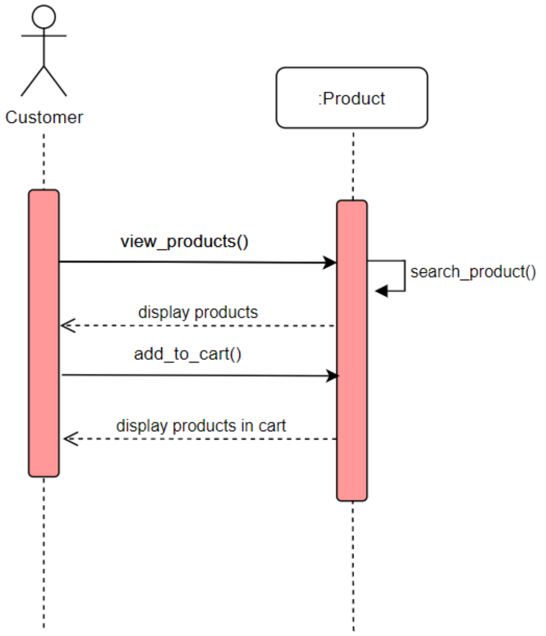


Figure 20 Sequence diagram to add products in cart

3. System must allow admin to add products

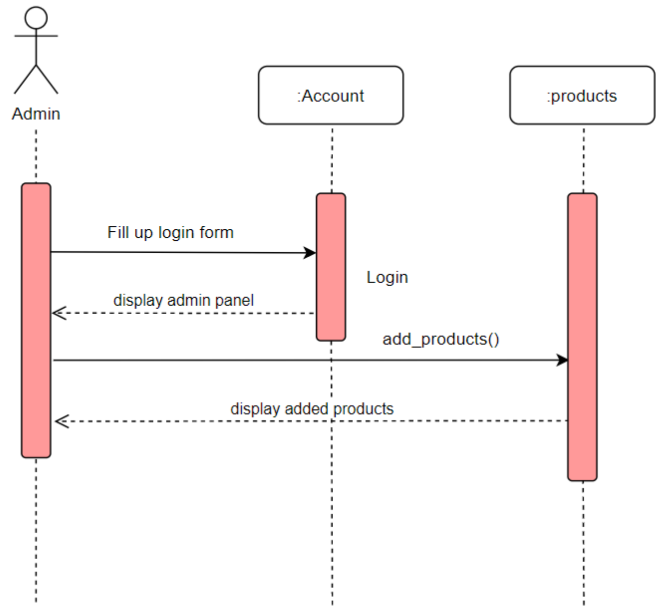


Figure 21 Sequence diagram to add products

4. System must allow admin to delete products from database

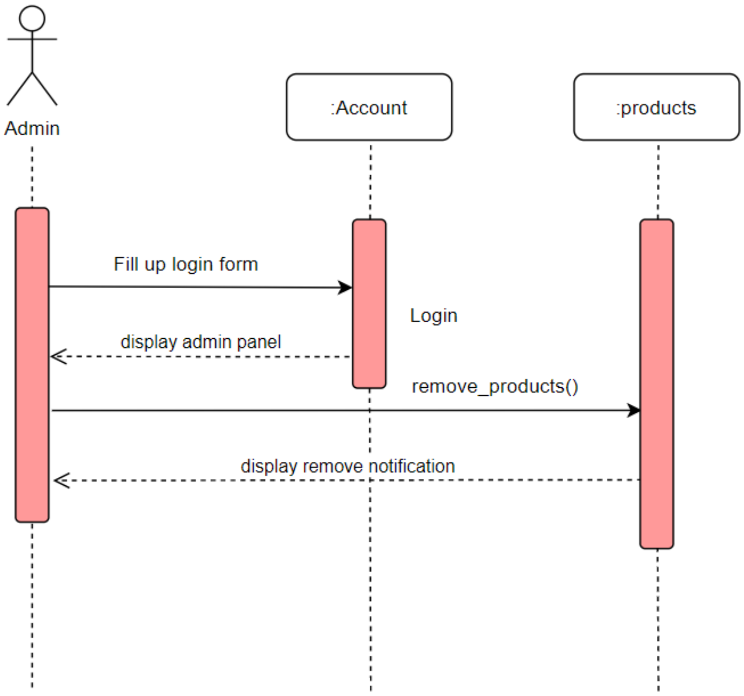


Figure 22 Sequence diagram to delete products

5. System must allow admin to edit product details

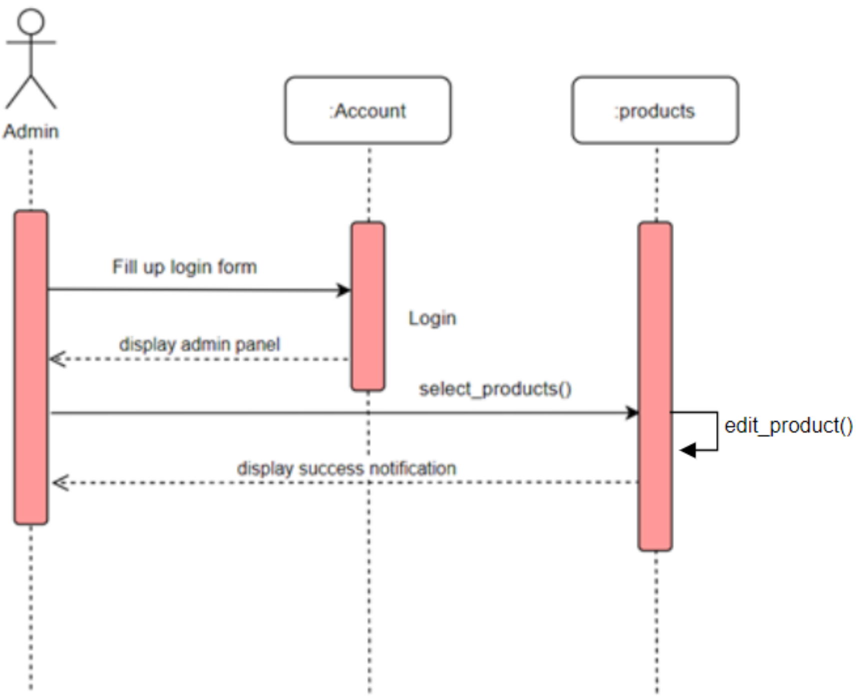


Figure 23 Sequence diagram to edit product details

6. System must allow customers to classify the image

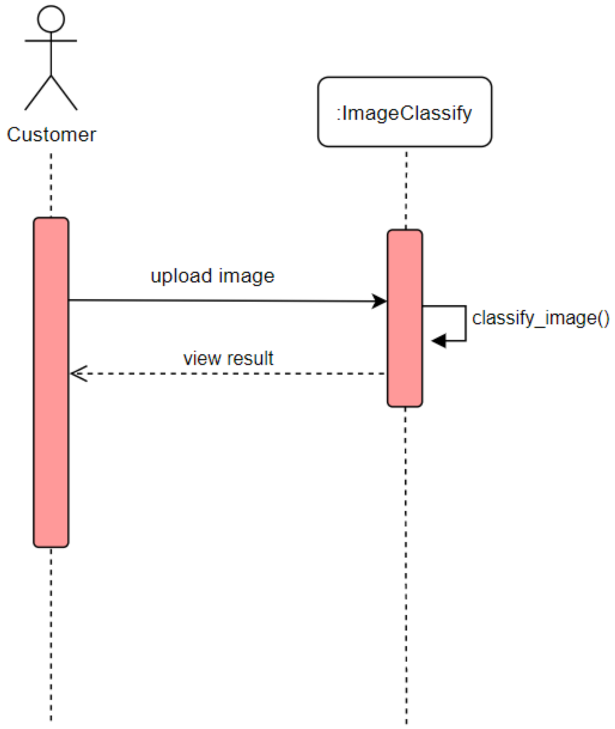


Figure 24 Sequence diagram to classify image data

3.3.4 Activity diagram

Activity diagram is a part of UML diagram. It helps to show dynamic aspect of a system. It shows model workflow between different use cases of the system. It is majorly used in business process modeling. The activity diagram always has a beginning and ending. Some belonging activity diagrams of the system are mentioned below:

1. **Customer view page:** The following diagram shows the initial view page by the customer. As there is no online payment through the system itself, customer directly goes to the home page with different products listed. The listed products are food item added by the admin.

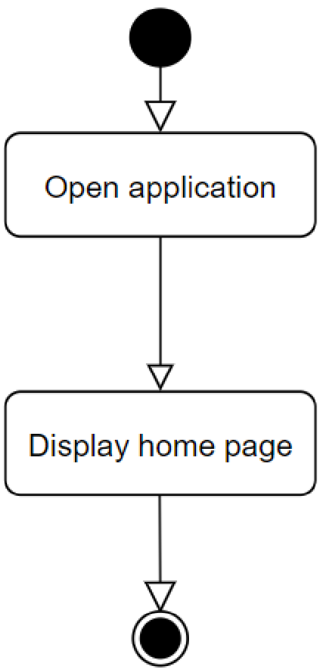


Figure 25 Activity diagram of customer in home page

2. **Admin login:** The following diagram shows the login process of an admin. Admin enters the username and password to the system and if validated, admin will be accessed to the admin panel. If the username or password is incorrect, system will show an error message.

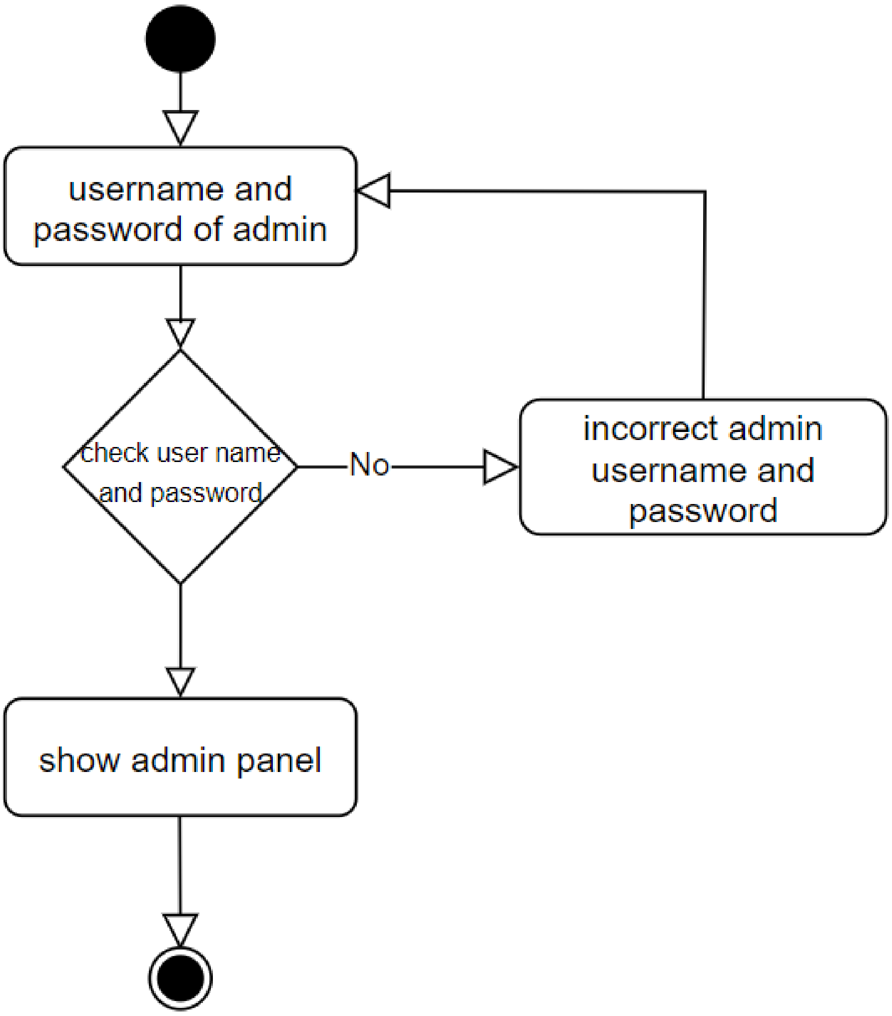


Figure 26 Activity diagram of admin login

3. **Order product:** The following diagram show the ordering process for any food item. At first, user sees product item in home page, there they can add any amount of item in the cart, in the cart they can increase, decrease or totally remove any product. Then they have to fill shipping address to confirm the order.

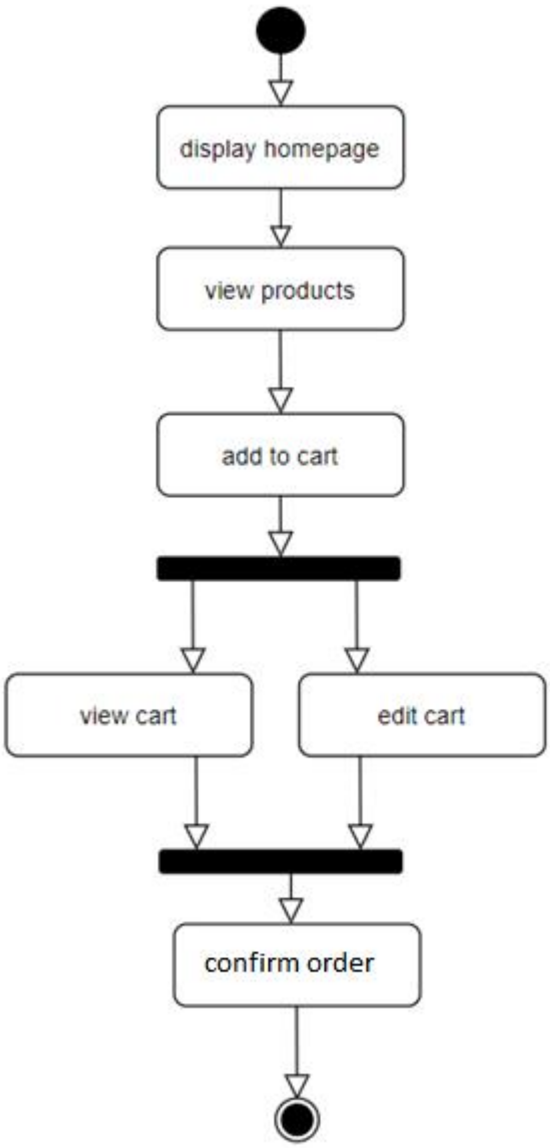


Figure 27 Activity diagram to add to cart

4. **Image classification:** The following diagram show the activity diagram to classify the food images. The user is in the home page and redirected to a page to upload image. There the user adds a image file in the designated place and gets back the result with an image with the specified result.

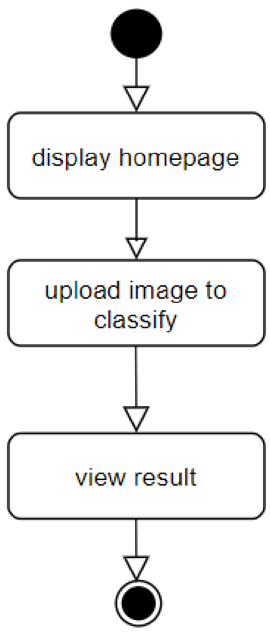


Figure 28 Activity diagram to classify an image

5. **Add/edit/remove product:** The following diagram show activity diagram to add, edit or remove any product in the database. The actions can only be performed by an admin so admin must login to the system with correct username and password. Then admin can choose the action weather to add, remove or edit product items.

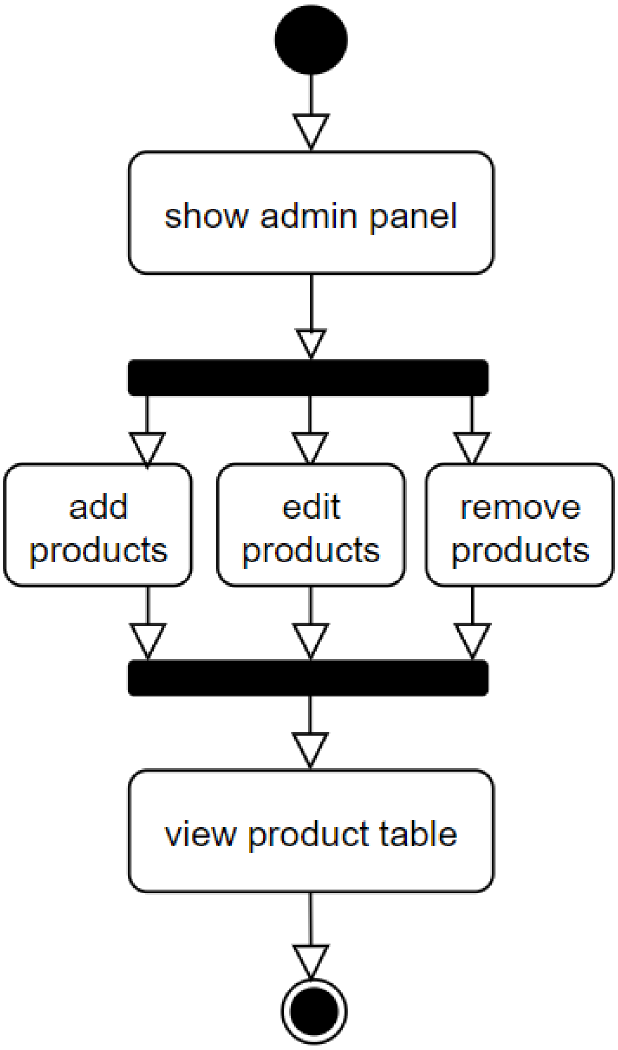


Figure 29 Activity diagram to add, edit or remove product item

3.4 Database design

3.4.1 Entity Relation Diagram (ERD)

The ERD is created to help analyze the requirement in the database to design and define the relation between different entity in the system. It is a primary requirement for database design in any system. It helps in initial visualization of the database requirements. The following diagram is the ERD of the system.

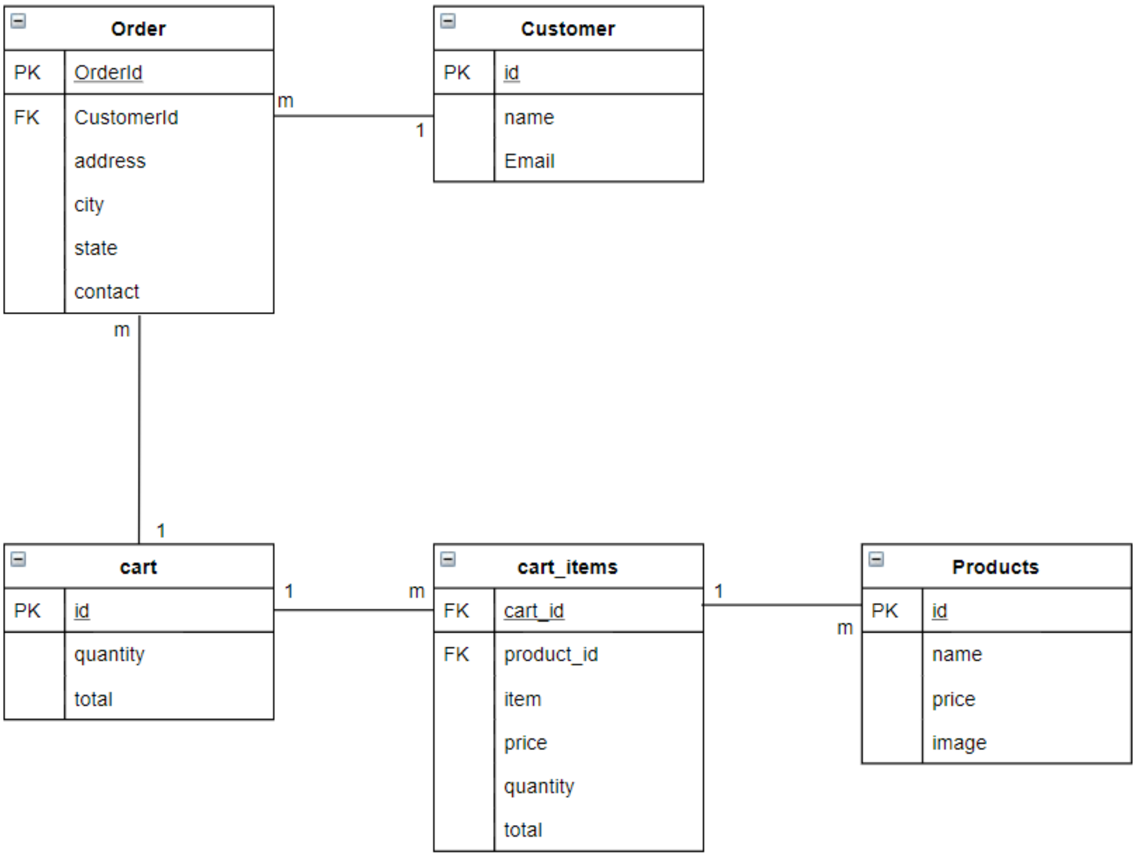


Figure 30 Entity Relation Diagram

Relationship between entities

- Customer and order have one to many relationships because one customer can have multiple orders
- Cart and order have one to many relationships because one cart can have multiple orders
- Cart_items and products have one to many relationships because cart_items can have multiple products.

3.4.2 Data dictionary

1. Customer

Field Name	Data Type	Field length	Constraint	Description
Customer_id	IntegerField	5	Primary key	Customer_id, auto generate
Name	CharField	20	Not null	Name of the customer
Email	CharField	20	Not null	Any email id

2. Products

Field Name	Data Type	Field length	Constraint	Description
Product_id	IntegerField	5	Primary key	Product_id, auto generate
Name	CharField	20	Not null	Name of the product
Price	DecimalField	5	Not null	Price of the product
Image	ImageField	-	Not null	Image of the product

3. Order

Field Name	Data Type	Field length	Constraint	Description
OrderId	IntegerField	5	Primary key	Order_id, auto generate
Customer_Id	IntegerField	5	Foreign key	Id of the customer who ordered the product
Address	CharField	20	Not null	Address of the customer for delivery
City	CharField	20	Not null	City of the customer for delivery
State	CharField	20	Not null	State of the customer for delivery
Contact	IntegerField	20	Not null	Contact number of the customer for delivery

4. Cart

Field Name	Data Type	Field length	Constraint	Description
Cart_id	IntegerField	5	Primary key	Cart_id, auto generate
Quantity	IntegerField	5	Not null	Total quantity of each ordered products
Total	IntegerField	5	Not null	Total price of overall products

5. Cart_item

Field Name	**Data Type**	**Field length**	**Constraint**	**Description**
Cart_id	IntegerField	5	Foreign key	Cart_id, auto generate
Product_id	IntegerField	5	Foreign key	Product id added with the cart item
Item	CharField	20	Not null	Name of food item
Price	Decimal Field	5	Not null	Price of each item
Quantity	Decimal Field	5	Not null	Quantity of each type of item
Total	IntegerField	5	Not null	Total price of overall products

3.5 User interface

3.5.1 Wireframe

By using wireframe, it was a great help to design and develop the website for both front end and backend.

1. Home page: this it the front page of the website that is shown at the first site when opened in the web browser.

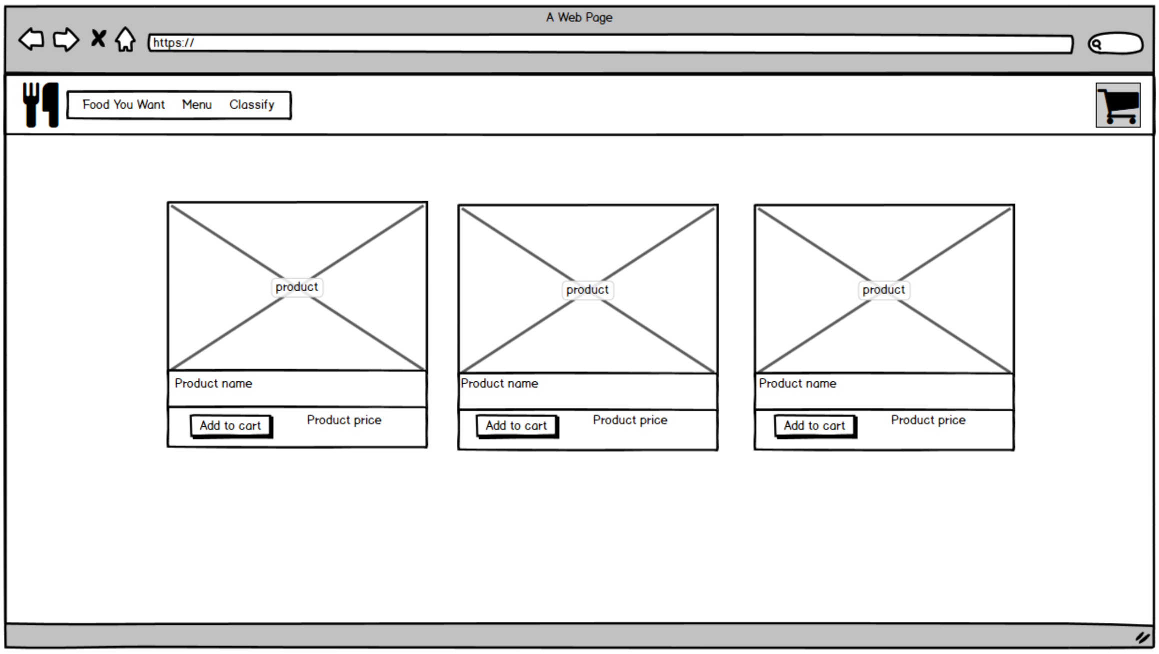


Figure 31 Wireframe of homepage

2. Cart page: this is the cart page where items are shown which are primarily added in the cart from home page. It shows all the items in the cart with their corresponding quantities which can be increased or decreased and see whole total.

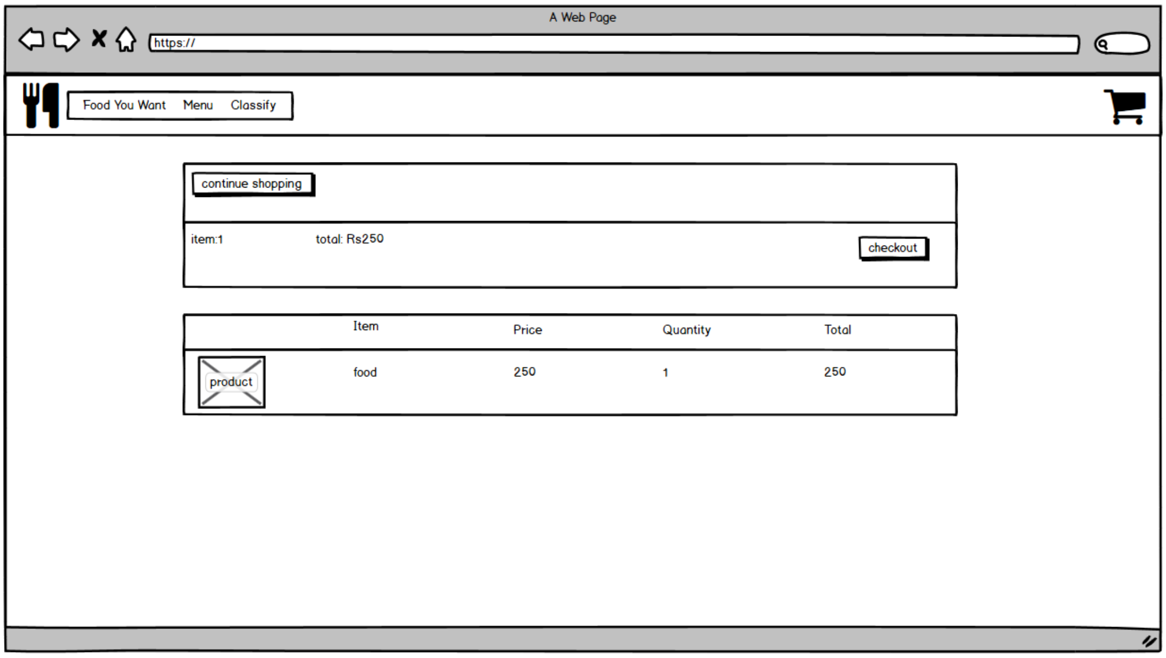


Figure 32 Wireframe of cart page

3. Checkout page: this is the checkout page where product in the cart is shown as a summary and includes a form to be filled before checkout for the delivery information.

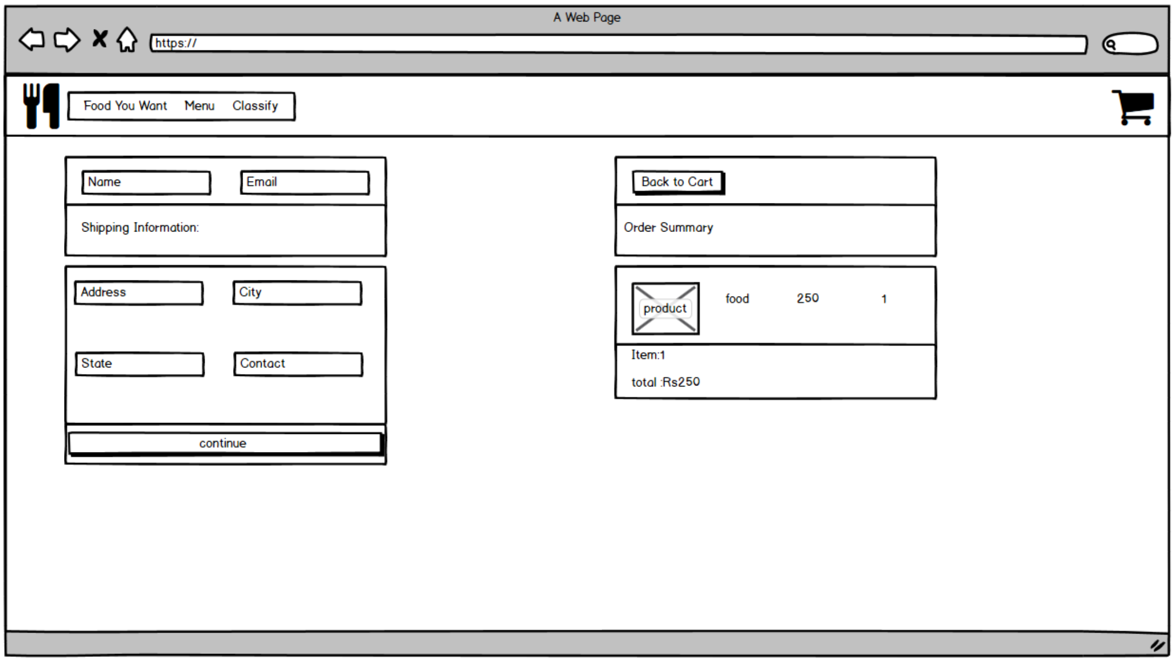


Figure 33 Wireframe of checkout page

4. Classify page: this page is for the classification of the images of the food items. Option for choosing a file is available and then it gets uploaded to be classified.

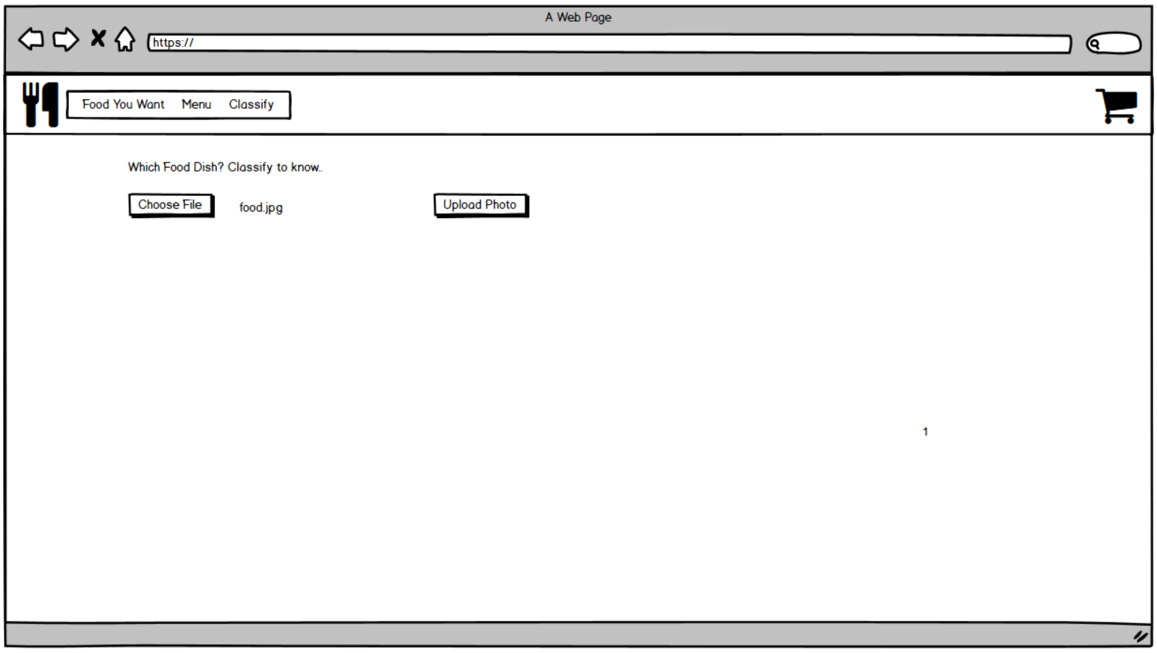


Figure 34 Wireframe of classification page

5. Classified page: when the image gets uploaded, it then gets processed and return the same image with the name of the food item.

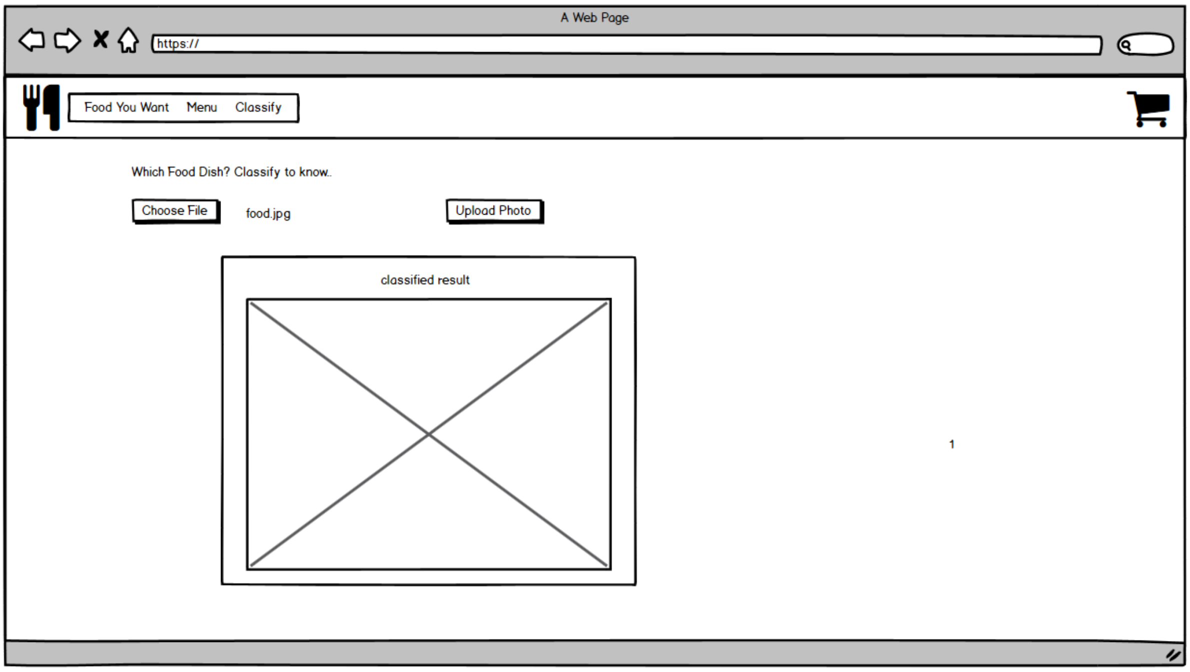


Figure 35 Wireframe of classified image

6. Login page: the following it the wireframe of the login page for the admin page.

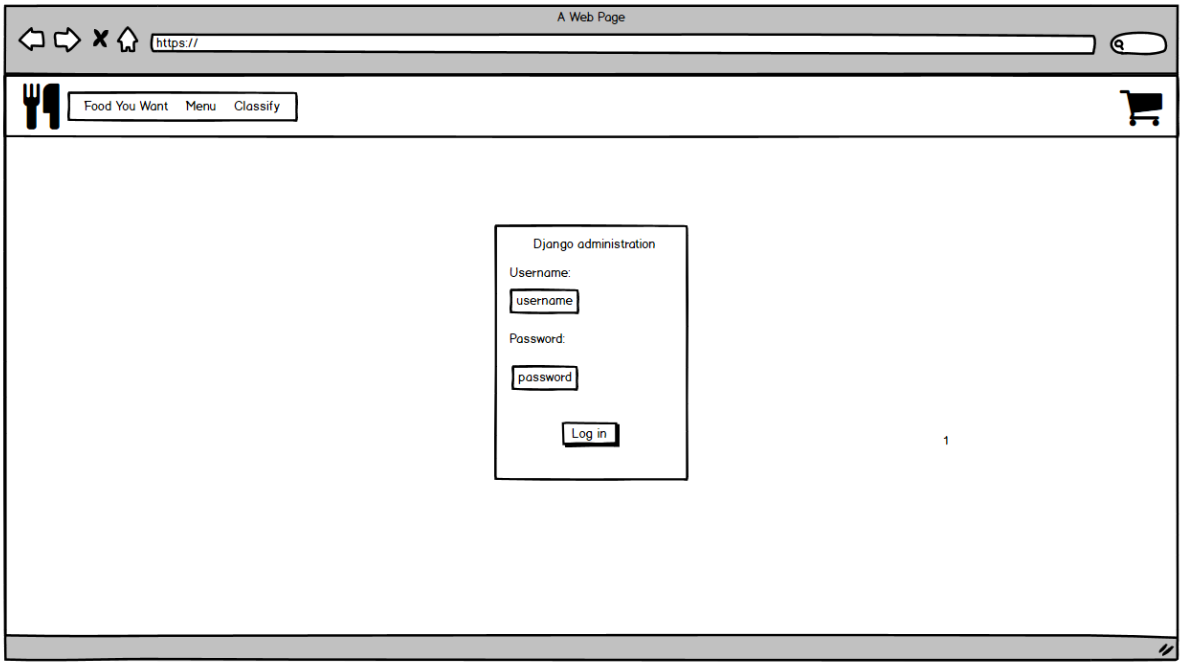


Figure 36 Wireframe for admin login

7. Admin panel: after logging in, admin gets to this page. Here, the admin can view, add or change customers, orders and products.

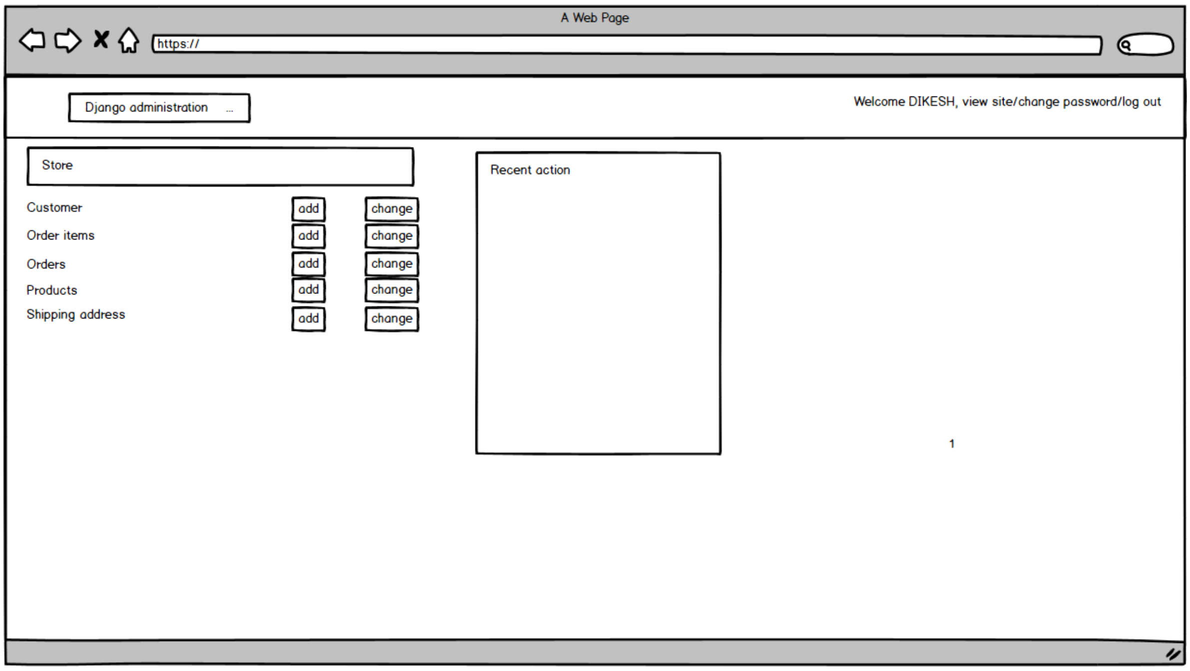


Figure 37 Wireframe of admin panel

8. Add product page: this page lets admin add a product with its name, price and image.

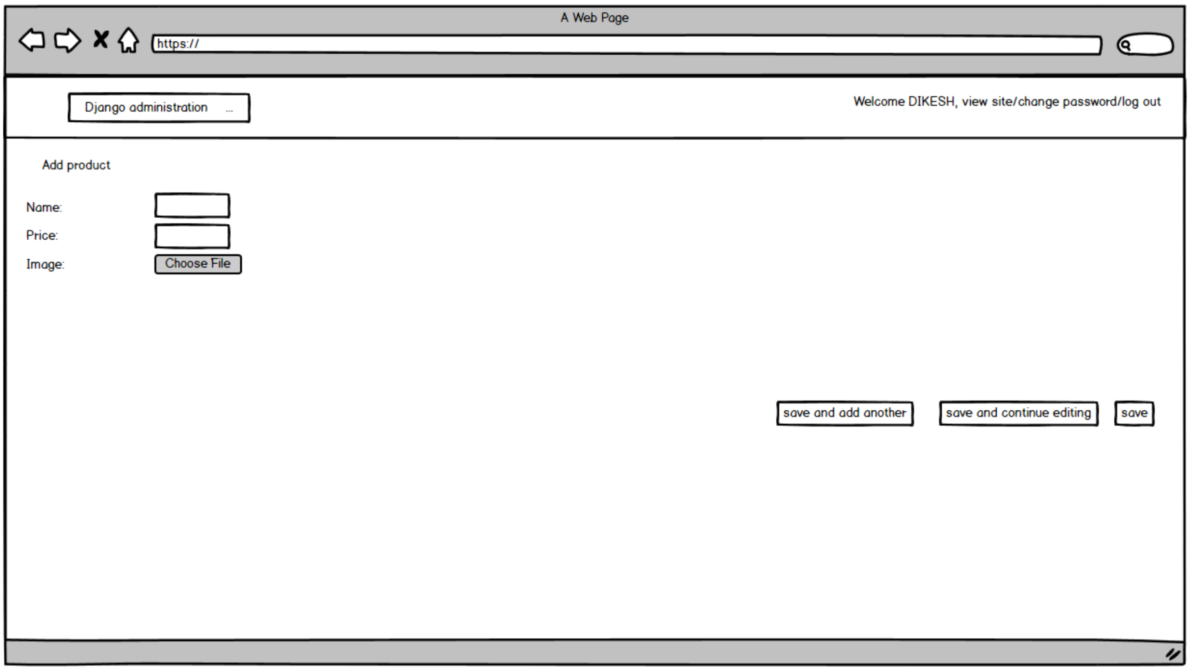


Figure 38 Wireframe to add a product

9. Product page: here, admin can view the list of available items if it needs to be changed.

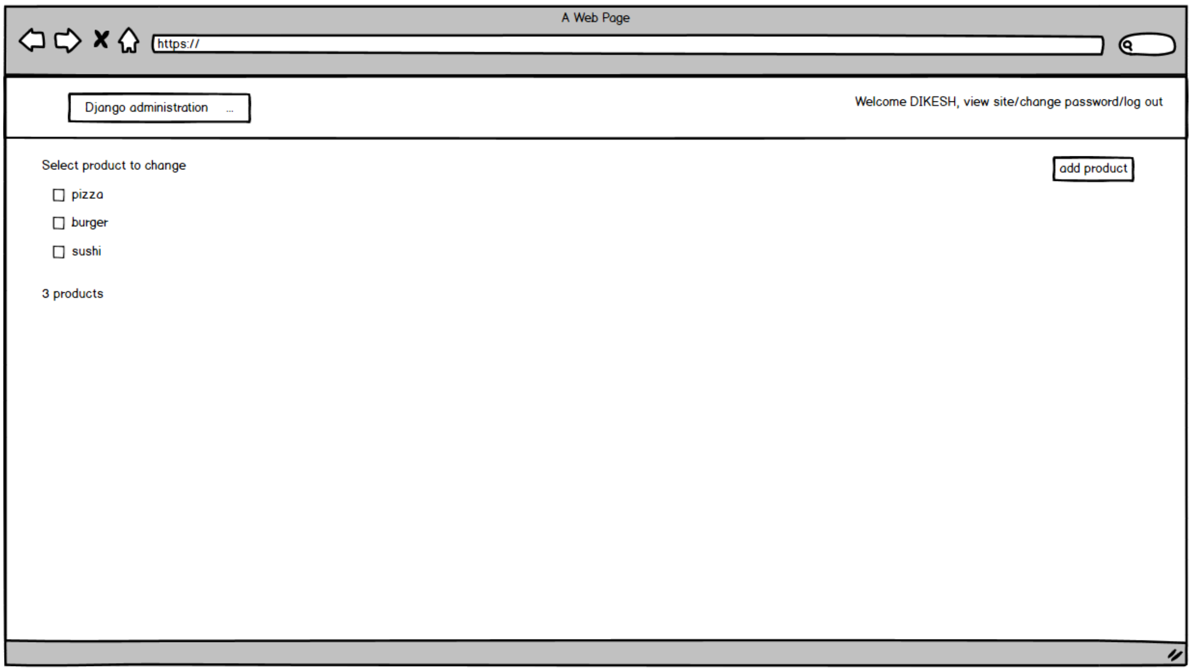


Figure 39 Wireframe to view list of available products

3.6 System architecture

System architecture can be divided in subgroups

3.6.1 Server

Server is the main component in the system architecture. It refers to the main storage of the system where all the data and information are saved and distributed whenever the demanded by an end user. There are different servers used for the project:

1. Database server: database server functions to serve data and information related to the application functionality like products and orders. The recorded data could be stored by customer and retrieved by admin and vice versa. Out system used default database of Django, SQLite as database server.
2. Web server: webserver will be required to serve and receive the data once the system in online in the internet. Since it is a web-based system, Apache HTTP server (apache tomcat) will be used as web server.

3.6.2 Clients:

Clients are divided into three types:

- Thick client
- Thin client
- Hybrid client

The designed system will have thin client because all the processing of application and database is done on the server and the customer relies on the server to do all the job.

3.6.3 Network

The system will use two types of networks to serve the data.

- Wide Area Network: since the system will the deployed in the internet, WAN will be used to make it reachable all over the internet
- Star network: all the clients are served from a single main server for data. So, it will be a centralized server.

3.6.4 Client/server architecture

The system uses three tier architecture that has presentation layer, application layer and database server. Presentation layer includes web interface in PC, developed with HTML, CSS and JS. Application layer includes the server for processing and serving the data to the end user. It also includes all the business logics needed in the application. Database server includes the main data base where all the data and information are stored. The benefits to this are it provides less access to the end user and protects applications and physical database.

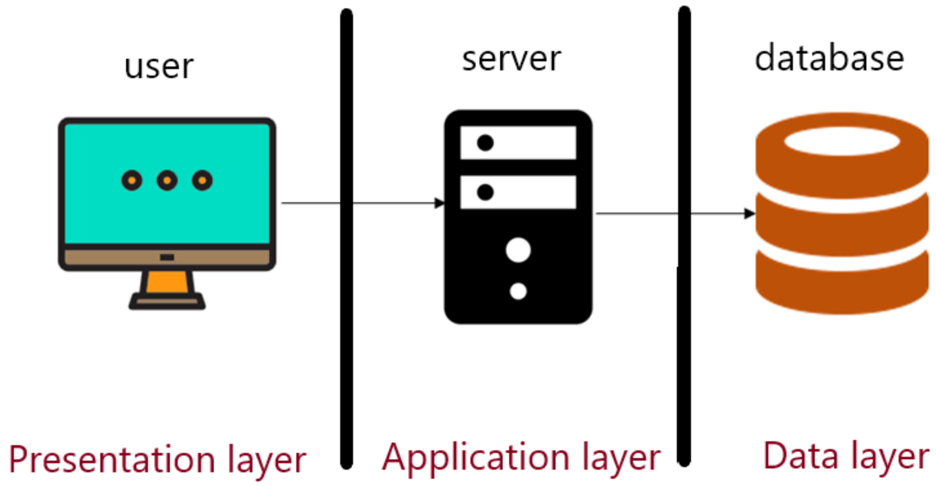
Why three tier architecture?

The three-tier architecture can be beneficial for the following reasons

- Easy upgradability in each layer if new technology is preferred
- Any kind of action in a layer cannot hamper another layer
- Flexibility in maintenance in hardware or software level

3.6.5 System architecture setup

System architecture for the application of website



3.7 Development of the food image classification

3.7.1 Data collection

Initial research was done on image classification and the feasibility of CNN in food classification through different papers and researches. There were many cases where people could not find good source of data with a proper collection. The classification needs a lot of data with wide variation of gain accuracy. After lots of research, dataset named "FOOD-101" was found which was exactly what this project needs. The dataset used can be found in this link: https://www.kaggle.com/kmader/food41. This is licensed by "K Scott Mader" in Kaggle. This dataset has 101000 images in total of 101 classes of unique fast food items that are most common in USA. Each food item has 1000 images that is maximum sized to 512 pixels, that is divided into 750 images for training samples and 250 images for testing samples. The entire database is sized 5GB.

Figure 40 One random image from each class from "FOOD-101" dataset

3.7.2 Importing necessary library

Some the important libraries used in the project are:

Numpy (Numerical Python): It is a python library that serves to provides array objects faster. For this project, Numpy is used to convert and retrieve array of the processed image data.

Keras: Keras is an open source library that runs on top of TensorFlow. It is majorly used for deep learning. It is user friendly and provides high-level APIs. It is used to load model and optimization tasks.

Matplotlib: It is an extension of NumPy and is used for the visualization of data and train/test progress.

```
from __future__ import absolute_import, division, print_function

import tensorflow as tf

import tensorflow.keras.backend as K
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras import regularizers
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import GlobalAveragePooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, CSVLogger
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.regularizers import l2

from tensorflow import keras
from tensorflow.keras import models
from tensorflow.keras.applications.inception_v3 import preprocess_input

import cv2
import os
import random
import collections
from collections import defaultdict

from shutil import copy
from shutil import copytree, rmtree

import numpy as np

import matplotlib.pyplot as plt
import matplotlib.image as img
```

Figure 41 Importing necessary libraries

3.7.3 Divide test and train data

The initial step is to divide the database before using it in the model. The data is divided into ¾ part for training purpose and ¼ for testing purpose. The prepare_data function helps take the train.txt or test.txt which has the list of divided images, then copies the listed item from source folder to destination folder to create train folder and test folder.

```
# Helper method to split dataset into train and test folders
def prepare_data(filepath, src,dest):
  classes_images = defaultdict(list)
  with open(filepath, 'r') as txt:
      paths = [read.strip() for read in txt.readlines()]
      for p in paths:
        food = p.split('/')
        classes_images[food[0]].append(food[1] + '.jpg')

  for food in classes_images.keys():
    print("\nCopying images into ",food)
    if not os.path.exists(os.path.join(dest,food)):
      os.makedirs(os.path.join(dest,food))
    for i in classes_images[food]:
      copy(os.path.join(src,food,i), os.path.join(dest,food,i))
  print("Copying Done!")
```

Figure 42 Dividing test data and train data

3.7.4 Train model method

The following figure shows the function for training the data. At first, it takes training data and testing data from designated folders. It preprocesses the image data to fit in the model. The image width and height are set to target size. Then with the InceptionV3, images are trained with the layers of pooling, activation and dropout. Then for the final prediction, softmax function was used. It saves checkpoints as it completes each epoch.

```
def train_model(n_classes,num_epochs, nb_train_samples,nb_validation_samples):
  K.clear_session()

  img_width, img_height = 299, 299
  train_data_dir = 'food-101/train_mini'
  validation_data_dir = 'food-101/test_mini'
  batch_size = 16
  bestmodel_path = 'bestmodel_'+str(n_classes)+'class.hdf5'
  trainedmodel_path = 'trainedmodel_'+str(n_classes)+'class.hdf5'
  history_path = 'history_'+str(n_classes)+'.log'

  train_datagen = ImageDataGenerator(
      preprocessing_function=preprocess_input,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True)

  test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)

  train_generator = train_datagen.flow_from_directory(
      train_data_dir,
      target_size=(img_height, img_width),
      batch_size=batch_size,
      class_mode='categorical')

  validation_generator = test_datagen.flow_from_directory(
      validation_data_dir,
      target_size=(img_height, img_width),
      batch_size=batch_size,
      class_mode='categorical')


  inception = InceptionV3(weights='imagenet', include_top=False)
  x = inception.output
  x = GlobalAveragePooling2D()(x)
  x = Dense(128,activation='relu')(x)
  x = Dropout(0.2)(x)

  predictions = Dense(n_classes,kernel_regularizer=regularizers.l2(0.005), activation='softmax')(x)

  model = Model(inputs=inception.input, outputs=predictions)
  model.compile(optimizer=SGD(lr=0.0001, momentum=0.9), loss='categorical_crossentropy', metrics=['accuracy'])
  checkpoint = ModelCheckpoint(filepath=bestmodel_path, verbose=1, save_best_only=True)
  csv_logger = CSVLogger(history_path)

  history = model.fit_generator(train_generator,
                      steps_per_epoch = nb_train_samples // batch_size,
                      validation_data=validation_generator,
                      validation_steps=nb_validation_samples // batch_size,
                      epochs=num_epochs,
                      verbose=1,
                      callbacks=[csv_logger, checkpoint])

  model.save(trainedmodel_path)
  class_map = train_generator.class_indices
  return history, class_map
```

Figure 43 Method for training model

3.7.5 Training data

The following figure shows the actual training of the data. The model was trained with all 101 classes for 40 epochs. The training process took almost 13 hours on my machine. The accuracy of the model is 0.95 which is considered good for image classification.

```
In [28]: # Train the model with data from 101 classes
         n_classes = 101
         epochs = 30
         nb_train_samples = train_files
         nb_validation_samples = test_files

         history, class_map_11 = train_model(n_classes,epochs, nb_train_samples,nb_validation_samples)
         print(class_map_11)
```

```
Epoch 27/30
4733/4734 [============================>.] - ETA: 0s - loss: 0.4916 - accuracy: 0.9365
Epoch 00027: val_loss did not improve from 0.94649
4734/4734 [==============================] - 1465s 310ms/step - loss: 0.4916 - accuracy: 0.9365 - val_loss: 0.9644 - val_accu
racy: 0.8119
Epoch 28/30
4733/4734 [============================>.] - ETA: 0s - loss: 0.4656 - accuracy: 0.9425
Epoch 00028: val_loss did not improve from 0.94649
4734/4734 [==============================] - 1553s 328ms/step - loss: 0.4655 - accuracy: 0.9425 - val_loss: 0.9674 - val_accu
racy: 0.8124
Epoch 29/30
4733/4734 [============================>.] - ETA: 0s - loss: 0.4426 - accuracy: 0.9479
Epoch 00029: val_loss did not improve from 0.94649
4734/4734 [==============================] - 1466s 310ms/step - loss: 0.4426 - accuracy: 0.9479 - val_loss: 0.9602 - val_accu
racy: 0.8141
Epoch 30/30
4733/4734 [============================>.] - ETA: 0s - loss: 0.4177 - accuracy: 0.9535
Epoch 00030: val_loss did not improve from 0.94649
4734/4734 [==============================] - 1463s 309ms/step - loss: 0.4177 - accuracy: 0.9535 - val_loss: 0.9569 - val_accu
racy: 0.8153
```

Figure 44 Model fitting

3.7.6 Predict image

In the following function, input image is preprocessed and with the passed-on model from load_model, it makes prediction of the image and produces predicted value and the image itself with the help of matplotlib.

```
def predict_class(model, images, show = True):
  for img in images:
    img = image.load_img(img, target_size=(299, 299))
    img = image.img_to_array(img)
    img = np.expand_dims(img, axis=0)
    img = preprocess_input(img)

    pred = model.predict(img)
    index = np.argmax(pred)
    food_list.sort()
    pred_value = food_list[index]
    #print(pred)
    if show:
        plt.imshow(img[0])
        plt.axis('off')
        plt.title(pred_value)
        plt.show()
```

Figure 45 Predict function

```
%%time
# Loading the best saved model to make predictions

K.clear_session()
model_best = load_model('bestmodel_101class.hdf5',compile = False)
```

```
Wall time: 3.58 s
```

Figure 46 Load saved model

```
images.append(imagepath+'omelette.jpg')
predict_class(model_best, images, True)
```

Figure 47 Call predict function by passing image

3.7.7 Graph plot for accuracy and loss

The blue line in first plot denoted train_accuracy tha improved over each epoch to reach as high as 0.94. the yellow line in the first plot denotes validation_accuracy that shows after about 5 epochs, it has been consistent, with not much improvements. The blue line in second plot denotes train_loss that decreased over each epoch which goes as low as 0.41. The yellow line in second plot denotes validation_loss which decreases up to 10 epochs and is almost consistent afterwards.

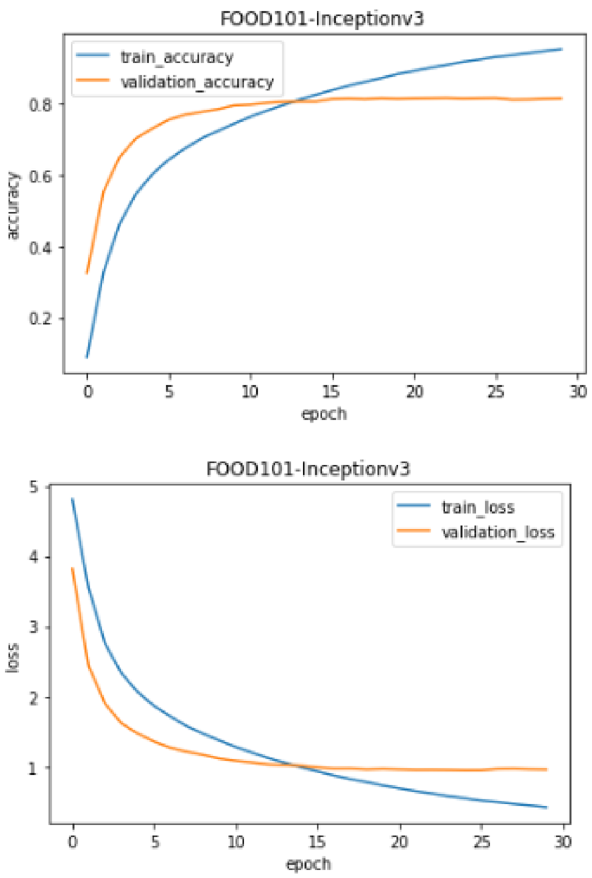


Figure 48 Visualization od accuracy and loss

3.8 Testing

Testing is done to test the system functionalities if working correctly or not. The main purpose of testing is to find out bugs, then debug it. It is an important phase before implementing the system. It involves checking the actual result with the expected result. This is done to ensure to make any system defect free.

Adding customer testing

Objective	To check if new customer gets created when new details gets added
Action	Details of customer filled up in the form
Expected result	Success message and redirect to home
Outcome	Message shown and redirected to home
Conclusion	Test successful

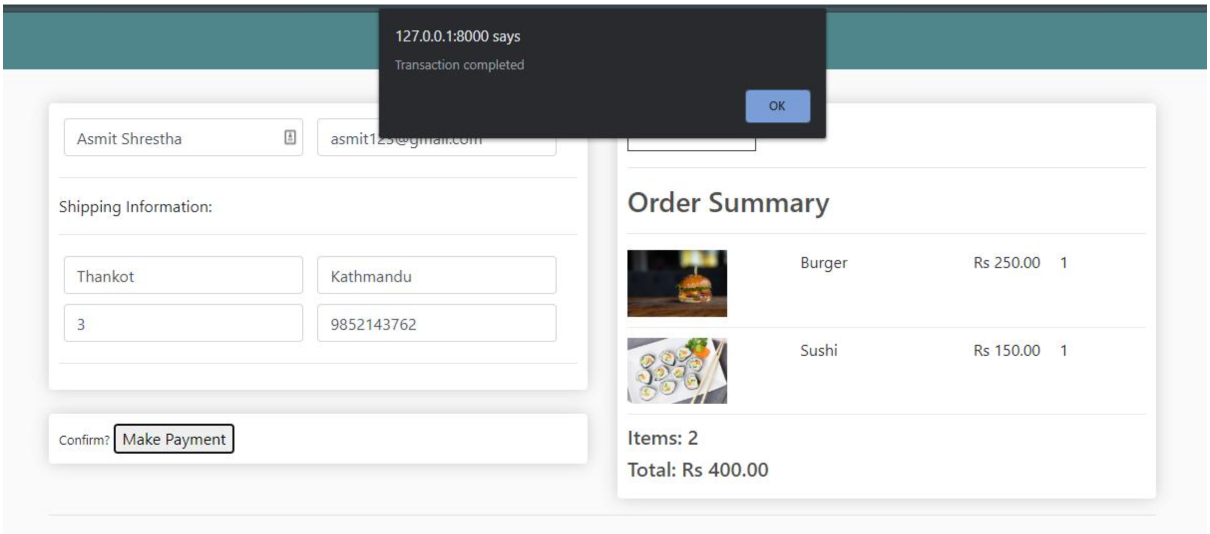


Figure 49 Making order with new user

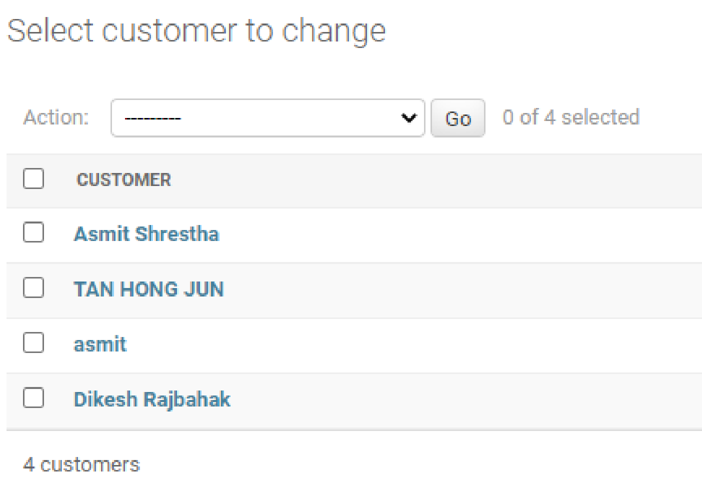


Figure 50 New user created

Add to cart test

Objective	To check if pressing add to cart button makes update on cart items
Action	Press add to cart button
Expected result	Number of products in cart increases
Outcome	The number in the cart increases
Conclusion	Test successful

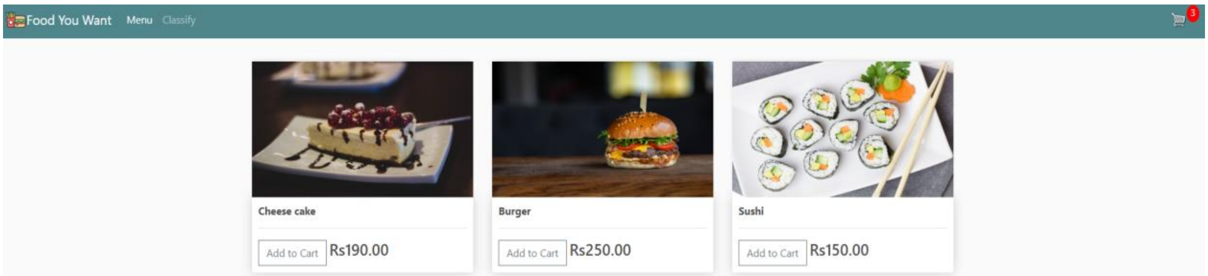


Figure 51 Cart value updated

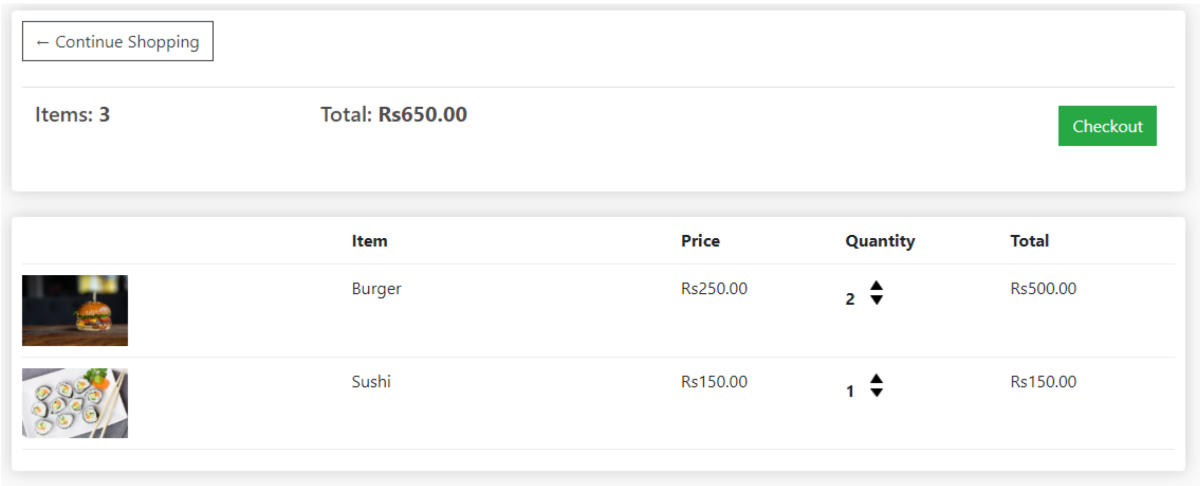


Figure 52 Cart gets updated

3.8.1 Turing test

Turing test was developed by "Alan Turing" in 1950 to test artificial intelligence if it could think as humans or not. So visual turing test in carried out to see if it can be distinguished from human.

To test this, 5 of my friends were to judge the final results published by myself and the AI model. They could easily distinguish the results between the AI model and me as a human. The model was unable to distinguish food in disturbed backgrounds and presented wrong answers for some food items which were easy for me to classify. To conclude, the system failed turing test.

3.8.2 Black box testing for image classification

Image	Expected result	Actual result
Image of Hummus	Hummus	Humus
Image of chocolate cake	Chocolate cake	Chocolate_cake
Image of samosa	Samosa	Samosa
Image of pizza	Pizza	Pizza
Image of dumpling	Dumpling	Frozen yogurt
Image of random text	No food detected	Chocolate_cake

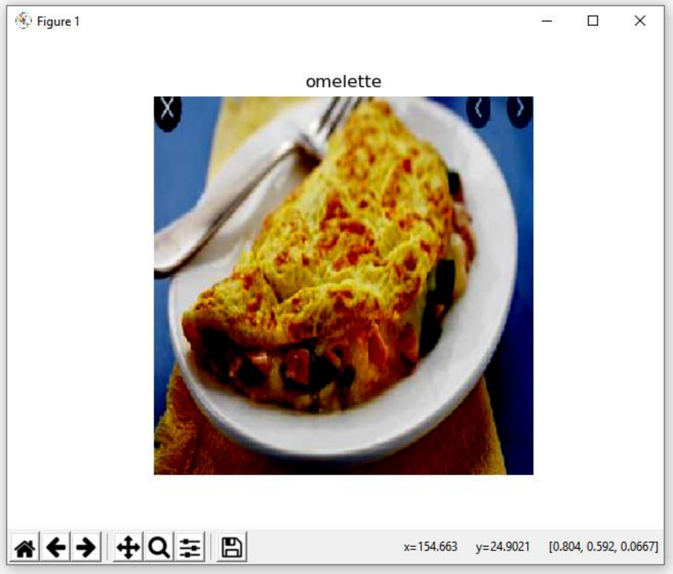


Figure 53 Classification done right

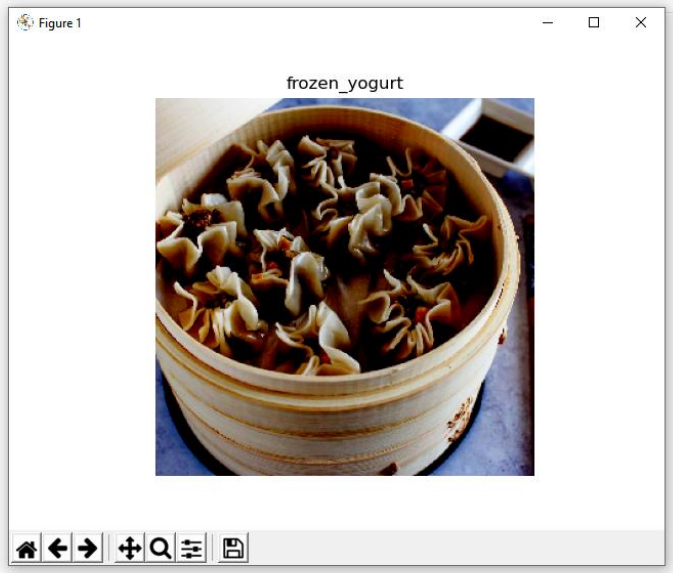


Figure 54 Classification done wrong

Chapter 4 Academic Question

How does CNN classify images, how is it better and which architecture is better?

CNN is a branch of deep learning that results more effectiveness in image processing and NLP. We used InceptionV3 for this system which is a CNN based architecture and it won ILSVRC 2014. It uses convolution layer of 1×1 which a small size but it helps to increase the model in both depth and width, hence improving the performance. It used global average pooling instead of fully connected layers because it showed better result. Primarily, it was named GoogLeNet but with the increase in depth, it renamed with Inception and it has versions from 1 to 4 till date. There are multiple softmax branches in the middle of the architecture to intermediate, as the architecture is deep. Total of 22 layers are present in this architecture. In basic, it trains the images of food where it only processes the portion of the image that is considered as food item. Then with the data collected from all the images, after training and testing, it creates a model of .hdf5 file. This file contains the parameters to classify any image with the trained data. So, after loading the model and feeding it with image, it classifies with food item.

Chapter 5 Conclusion

There has been a long history of computer vision. Humans have tried to replicate human intelligence into the machines so that it could learn and classify objects. The initial stage of computer vision was to detect edge from an image and see 3D features in a 2D image. It has come a long way since then. The machines can perform excellent result in image processing. With the improvement in technologies and the neural networks, some systems can show results similar to human and also outperform human. With the wide range of uses, computer vision is being implement in websites, mobile devices, industries etc. The systems working with computer vision is just going to be better over the course of time, with more data collected from the users themselves. With the advancement in technology, development in smartphones, it will be reached to a larger mass of people. Food image classification is achieved through InceptionV3 architecture that is based on CNN. It is trained with the dataset "FOOD-101" which is a dataset of food images with 101 classes, 1000 images each and 101000 images in total. The dataset includes common fast food items in USA. The model is trained with all the images for 30 epochs that took time duration of about 13 hours total. The accuracy of the model is 0.94 and the loss is 0.41. The model is implemented in a web interface which is developed using Django for backend and HTML, CSS and JS for frontend. The web interface is a restaurant app that functions to serve as an online food delivery service. So, it is mainly focused for food images, but it still has flaws and error during classification. The results of these models will obviously get better over time with more improvements. As I conclude, the computer vision has come so far with better results and will be certainly better in the future and it is an important part of technological advancement since it sets remarkable result that could be used for better purpose.

Chapter 6 Critical Evaluation

There are multiple algorithms that makes image processing possible. Traditional neural networks can also perform image processing but with the deep neural networks having deeper and wider architecture, it can detect more features, process faster with more data. It is already an impressive achievement to mimic the human capabilities of image classification. Now with CNN, the results have improved drastically. The model was trained with a limited data with minimal hardware and it results 0.94 in accuracy. The model can do better with higher volume of data and superior hardware. CNN helps to focus on only the feature areas in an image that lessens the computational power. With the non-linearity layer, different types features can be extracted. However, it still shows incorrect result when one food item looks too similar to some other item.

Chapter 7 Future Plan

There can be improvement in the existing system with the following ways:

- There could be more augmentation on test images
- Fine tuning could be done on entire dataset
- Threshold could be set for wrong classification
- The hyper parameters, their values could be changes for possible better result.

Chapter 8 References

Anon., n.d. s.l., s.n.

Anon., n.d. [Online].

Arnold, L., Rebecchi, S., Chevallier, S. & Moisy, H. P., 2011. An Introduction to Deep Learning. *Proceedings of the European Symposium of Artificial Neural Network,* Volume 1, pp. 477-488.

Aziz, F., Labbani-Igbida, O., Radgui, A. & Tamtaoui, A., 2018. Computer Vision and Image Understanding. *CNN-based features for retrieval and classification of food images,* pp. 70-77.

Azumio, 2017. *Introducing Calorie Mama.* [Online]
Available at: https://www.azumio.com/blog/azumio/calorie-mama-release
[Accessed 02 05 2020].

Bossard, L., Guillaumin, M. & Gool, V. L., 2014. *Food-101 - Mining Discriminative Components with Random Forests,* Switzerland: Springer International Publishing Switzerland.

Chollet, F., 2017. *Xception: Deep Learning with Depthwise Separable Convolutions,* s.l.: Google.

Christodoulidis, S., Anthimopoulos, M. & Mougiakakou, S., 2015. *Food Recognition for Dietary Assessment Using Deep Convolutional Neural Networks,* Bern: ARTORG Center for Biomedical Engineering Research,.

Ciocca, G., Napoletano, P. & Schettini, R., 2017. *CNN-based Features for Retrieval and Classification of Food Images,* Italy: Department of Informatics, Systems and Communication,.

Deshpande, A., 2016. *A Beginner's Guide to Understanding Convolutional Neural Network.* [Online]
Available at: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/
[Accessed 12 04 2020].

Ege, T. & Yanai, K., 2017. *Simultaneous Estimation of Food Categories and Calories.* Nagoya, Department of Informatics, The University of Electro-Communications, Tokyo, pp. 172-175.

Fung, V., 2017. *An Overview of ResNet and its Variants.* [Online]
Available at: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035
[Accessed 23 04 2020].

Geeksforgeeks, 2018. *Introduction deep learning.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-deep-learning/
[Accessed 28 05 2020].

Google, n.d. *Google Lens.* [Online]
Available at: https://lens.google.com/
[Accessed 28 05 2020].

Hokuto, K., Kiyohara, A. & Makoto, O., 2014. *Food Detection and Recognition Using Convolutional Neural Network.* Tokyo, s.n.

Howard, G. A. et al., 2017. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision,* s.l.: Google Inc..

Huang, T. S., n.d. *Computer Vision: Evolution and Promise,* illinois: University of Illinois at Urbana-Campaign.

Khan, S., Rahmani, H., Shah, S. A. A. & Bennamoun, M., 2018. *A Guide to Convolutional Neural Networks for Computer Vision.* s.l.:Morgan & Claypool Publishers.

Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2017. *ImageNet Classification with Deep Convolutional Neural Network,* s.l.: Communications of ACM.

Lu, Y., 2019. *Food Images Recognition by Using Convolutional Neural Networks (CNNs),* East Lansing: Department of Biosystems and Agricultural Engineering, Michigan State University.

McGinty, B., 2018. *Deep learning at scale.* [Online]
Available at: https://verneglobal.com/news/blog/deep-learning-at-scale
[Accessed 29 05 2020].

Mihajlovic, I., 2019. *Everything You Ever Wanted To Know About Computer Vision..* [Online]
Available at: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e
[Accessed 28 05 2020].

Missinglink, 2019. *Convolutional Neural Networks for Image Classification.* [Online]
Available at: https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-networks-image-classification/
[Accessed 29 05 2020].

Nigam, V., 2018. *Understanding Neural Networks from neuron to RNN, CNN, and Deep learning.* [Online]
Available at: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90
[Accessed 29 05 2020].

Roberts, L. G., 1963. *Machine Perception of Three-Dimensional Solids,* Cambridge: Massachusetts Institute of Technology.

Sharma, A., 2017. *Understanding Activation Functions in Neural Networks.* [Online]
Available at: https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0
[Accessed 13 05 2020].

Sharma, S., 2017. *Activation Function in Neural Networks.* [Online]
Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
[Accessed 11 05 2020].

Sharma, S., 2017. *Activation Functions in Neural Network.* [Online]
Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
[Accessed 11 05 2020].

Shimoda, W. & Yanai, K., 2015. *CNN-Based Food Image Segmentation Without Pixel-Wise Annotation,* Tokyo: Department of Informatics, The University of Electro-Communications.

Simonyan, K. & Zisserman, A., 2015. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* San Diego, ICLR 2015.

Simonyan, K. & Zisserman, A., 2015. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* San Diego, ICLR 2015.

Singh, S. P., n.d. *fully connected layer.* [Online]
Available at: https://iq.opengenus.org/fully-connected-layer/
[Accessed 24 05 2020].

Sun, J., Radecka, K. & Zilic, Z., 2019. *Exploring Better Food Detection via Transfer Learning.* Tokyo, 16th International Conference on Machine Vision Applications.

Sun, J., Zilic, Z. & Radecka, K., 2019. *FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks.* Tokyo, Department of Electrical and Computer Engineering, McGill University.

Szegedy, C. et al., 2015. *Going Deeper with Convolutions.* Boston, IEEE.

Szeliski, R., 2011. *Computer Vision: Algorithms and Applications.* 1st ed. New York: Springer.

UKESSAYS, 2016. *Definition Of Fact Finding Techniques Information Technology Essay.* [Online]
Available at: https://www.ukessays.com/essays/information-technology/definition-of-fact-finding-techniques-information-technology-essay.php#:~:text=Definition%20of%20Fact%2Dfinding%20Techniques,prototyping%20and%20joint%20requirements%20planning.
[Accessed 22 01 2020].

Upadhyay, Y., 2019. *introduction to computer vision.* [Online]
Available at: https://medium.com/alumnaiacademy/introduction-to-computer-vision-4fc2a2ba9dc
[Accessed 15 05 2020].

Verdict, 2020. *History of computer vision: Timeline.* [Online]
Available at: https://www.verdict.co.uk/computer-vision-timeline/
[Accessed 28 05 2020].

Zbigatron, 2018. *the early history of computer vision.* [Online]
Available at: https://zbigatron.com/the-early-history-of-computer-vision/
[Accessed 28 05 2020].

Zoumpourlis, G., Doumanoglou, A., Vretos, N. & Daras, P., 2017. *Non-linear Convolution Filters for CNN-based Learning,* Greece: Information Technologies Institute, Center for Research and Technology Hellas.

Chapter 9 Appendix

Model training:

https://drive.google.com/file/d/1Plk8q6TZSqhf1SMnVjXuMj3riIFliv71/view?usp=sharing