A Food Recommendation System Combining InceptionV3 and KNN Considering Calorie Estimated From Images

MSc Research Project
Data Analytics

Debratna Dhali
Student ID: 19201028

School of Computing
National College of Ireland

Supervisor: Dr. Majid Latifi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Debratna Dhali
Student ID:	19201028
Programme:	Data Analytics
Year:	2020
Module:	MSc. Research Project
Supervisor:	Dr. Majid Latifi
Submission Due Date:	16/08/2021
Project Title:	A Food Recommendation System Combining InceptionV3 and KNN Considering Calorie Estimated From Images
Word Count:	8684
Page Count:	28

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Debratna Dhali
Date:	22nd September 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Food Recommendation System Combining InceptionV3 and KNN Considering Calorie Estimated From Images

Debratna Dhali
19201028

Abstract

Dietitians and nutritionists have a major concern about the eating habits of people especially at a time when obesity and non-communicable diseases due to unhealthy eating habits are on the rise across the globe. The improvement of computational power provides an opportunity to automate the process of assessing the food intake of people. Rather than the regular monitoring practices, an automated process of doing this is more convenient and the same is proposed in this research. A computer vision based system is proposed in this research to detect the food item and estimate the overall calorie intake using the features extracted from pre-trained InceptionV3 algorithm, a process which in machine learning is referred to as transfer learning. Nutritionix database has been webscraped to construct the dataset containing nutritional information of the detected food items. To further enhance the system, K-nearest neighbors algorithm has been implemented to develop a recommender system that suggests alternatives of the detected food items. The proposed image classification model is mainly evaluated against metrics like training and validation accuracy and train and validation loss. These solutions are aimed at addressing the problems of time consumption, imprecision, under reporting etc. which can occur with traditional monitoring processes.

Keywords - Computer vision, Deep Learning, Image Classification, Food Detection, Web Scraping, Calorie Estimation, Recommender System, KNN

Contents

1 Introduction

Intake of food with high calorie content results in abnormal accumulation of fat in the body. This condition which is otherwise avoidable results in harmful health conditions like obesity, cholesterol, heart attacks, blood pressure and prostate cancer etc. (Munsell et al.; 2014). Unhealthy eating habits also has a direct and considerable impact considering that it leads to an increase in chronic diseases like diabetes, obesity, dental diseases and cardiovascular diseases etc.(Nishida et al.; 2004) After being significantly ignorant about this, people now, across various age groups have developed an inclination towards diet management. There is however a major challenge in diet management which is maintaining a balance between what the individual is consuming and how they are monitoring the overall intake. With the advent of deep learning in the field of image processing, researchers are aggressively employing image recognition models for various purposes such as cancer detection, video frame analysis etc. Similarly, researchers are also keen on estimating calories from food images using various machine learning and deep learning techniques (Darapaneni et al.; 2021). This study is concentrated on developing a system that effectively determine the calorie and nutritional value from food images thus allowing users to monitor their overall calorie intake in each meal and also provides them alternatives based on the detected items having similar nutritional content. This aims at having a more efficient diet management and a more balanced nutrition for the users.

1.1 Motivation

An individual's health depend heavily on their lifestyle in which food plays an extremely crucial part. It has been proved that diet and lifestyle is majorly responsible for the growing chronic epidemics in some countries. Chronic diseases were reported to be responsible for 60% of the 56.5 million deaths reported and 46% of the global diseases (Allegra et al.; 2020). This figure is estimated to increase by over 50% by 2025. Harris and Guten (1979) in their study grouped lifestyle in five different categories out of which two of them correlate to food such as having healthy eating habits or avoiding harmful substances like junk etc. In a study conducted by Livingstone et al. (2004) on approaches to understand eating habits, it was recommended that the amount and kind of food an individual intakes in a meal should be recorded to further analyze the overall calorie and nutrient intake of that individual.

Recognizing objects from images is a particularly complex task for machines since for machines, images are merely numbers without a meaning. Several approaches have been proposed to leverage this task among which deep learning techniques have been found to provide more state-of-the-art performance. Convolutional Neural Networks (CNNs) are particularly suited for the tasks that require object detection. Hence, most of the researches done in this field involved using a CNN model to facilitate image detection. However, researchers often encounter several problems with image detection. For example, Ayon et al. (2021) in their future work reported that their model had ambiguity issues for similar looking food items. Also, the angle at which the photo was taken impacted image detection. In this case, the model seemed to work inaccurately for images taken from a top view. On the other hand, there are various applications, both web based and android based that track calorie intake by monitoring the food intake of the user. However, this can be a tedious and inaccurate process because the user needs to manually input the portion size which can lead to inaccurate results (Darapaneni et al.; 2021).

Hence, the main motivation was to build an user friendly and efficient system that eradicated these issues. Accordingly, a system has been built that not only determines food items accurately but also shows their nutritional content. In addition to this, a recommender system is built to suggest different food items to users having similar nutritional content. In this research paper, the proposed model trains the dataset on the weights of the pre-trained InceptionV3 model. The dataset used in this research has 101 different food categories out of which 72 has been chosen with each category having 1000 images in total. The process of using an existing model to train another dataset is called transfer learning in machine learning terminology and the same has been used here.The output of the model was mapped with a dataset containing the nutrient information of several food items. This dataset was prepared by web scraping the Nutritionix database using their API key. Further, a KNN model was implemented on the basis of the calorie information derived from the previous steps to build the recommender system. The output of the recommender system suggests 3 food items to the user that has similar nutritional value as the detected one.

1.2 Research Question

Dietary management can be a daunting task because people are often unaware about the kind of portions or calories they are in taking. Hence, food detection and calorie estimation is extremely significant to help individuals with better and efficient diet management so that problems like obesity, cholesterol which are raging issues in today's world can be significantly reduced.

RQ:*"How can a combination of InceptionV3 and KNN algorithm be used to develop a system that not only estimates the nutritional information from food images but also suggests similar alternatives to the users?"*

To solve the research question, the objectives mentioned below are specified and implemented.

1.3 Research Objectives

Table 1 lists the research objectives and intended goals of the research that has been done here.

Table 1: Research Objectives

S No.	Objective	Description
1	Literature Review	A critical review and evaluation of peer reviewed literature on work done in the field of food identification and calorie estimation using Computer Vision techniques.
2	Dataset Identification and pre-processing	Identification of food image dataset to perform image classification and calorie estimation

2.1	Resizing Image	The images in the dataset are available in different sizes. They need to be resized so that it is in accordance to the model requirement.
2.2	Rescaling Images	The pixel values of the images should be normalized. They generally range from 0-255 but they will be rescaled to 0-1 as preferred by neural nets.
2.3	Splitting dataset	The dataset is splitted into training and testing dataset based on the train.txt and test.txt files.
2.4	Feature Extraction	Weights of the Pre-trained models like InceptionV3 can be used to extract features from the images.
3	Image classification model implementation	Implementing neural network algorithms on the chosen dataset to detect food items from the supplied image.
4.1	Calorie Estimation dataset	Constructing a calorie information dataset to get the nutritional information of detected food item.
4.2	Calculating Calorie of detected food	Mapping the calorie dataset with the detected food label to calculate calorie information.
5	Recommender System	Building a recommender system to suggest alternative food items to users.
6	Evaluation of applied algorithm	The applied algorithms were evaluated against parameters like accuracy and valuation loss.

1.4 Structure of Research Paper

The research paper is further subdivided into different sections. Section 2 consists of the Literature Review in which previous work done in this field is investigated. In Section 3, Research Methodology in which the different steps leading to the implementation and evaluation of the model following the data mining methodology is explained. The blueprint of the research work and algorithms applied to achieve the research objectives is discussed in Section 4. Model implementation is discussed in Section 5 and the evaluation of the applied model along with additional experiments performed to achieve the final result is discussed in Section 6. Section 7 concludes the research paper and focuses on the future scope of the research.

2 Related Work

In this section, some of the previous works done in the field of image processing and detection using Computer Vision and food recommendation system has been reviewed to understand and critically examine the challenges that researchers have faced in the past. The possible approaches to overcome the underlying problems has been investigated in this section. Section 2.1 discusses the various image segmentation techniques and work done on the respective techniques. Since transfer learning forms the basis of this research, it is imperative that a comparative review of the different models and their effectiveness is necessary to evaluate which model to use. This is discussed in Section 2.2. Previous works on calorie estimation techniques are discussed in Section 2.3. Building a recommender system is the final stage of the project and previous work done in this field has been critically reviewed in Section 2.4. Section 2.5 provides the comparative analysis of the different techniques used by other researchers in this research domain which is followed by the conclusion.

2.1 Image Segmentation Techniques

One of the most fundamental task of Computer Vision is image segmentation. Image segmentation is rather complex given its requirement of low level spatial information. Image segmentation is of two types, namely, semantic segmentation and instance segmentation (Sultana et al.; 2020).

In semantic segmentation every pixel in the image is given a separate class label (Wei et al.; 2016). There have been many successful semantic segmentation techniques after the success of AlexNet in 2012. Among them, the Fully Convolutional Network (FCN) was the most successful network.This model proposed by Long et al. used AlexNet, VGGNet and GoogleNet, all of which were pre-trained on the ILSVRC data. While FCN successfully achieved state of the art result but it had major drawbacks as well. The use of local information in FCN resulted in ambiguity of the semantic segmentation process. To reduce the ambiguity, the contextual information from the image was required (Sultana et al.; 2020).

Similar to semantic segmentation, instance segmentation techniques also employed CNN for better accuracy. The sole difference being in instance segmentation, an additional segmentation mask is added to layer where object detection is performed. This enhances the accuracy of the model and also reduces test time (Sultana et al.; 2020).

Mask R-CNN is an extension of Faster R-CNN which is mainly used for object detection. A branch for binary mask prediction is extended to the Faster R-CNN network to facilitate instance segmentation. This model was found to be computationally cost effective because it generates bounding boxes which in turn is extremely beneficial for object detection (He et al.; 2017). However, this process also involves expensive computational alignment procedures. Also, since bounding boxes are created, separate masks are created for every instance and to overcome this problem researchers have suggested the generation of segmentation mask instead of bonding box. Even though, segmentation masks help models have better accuracy but it lacks the power of capturing instances of objects having varying scales. Hence, on a comparative basis Mask R-CNN seems to be a fairly effective algorithm which can be used for performing image segmentation.

2.2 Review of Food Image Classification using Transfer Learning

Transfer learning is a handy tool that solves the problem of limited training data. Deep transfer learning tries to utilize the knowledge from one domain which is the source domain to the target domain. There are various techniques of deep transfer learning and in this research, Network based transfer learning is used. In this technique a part of the network is pre-trained in the source domain which includes the network structure and its parameters. The front layers work as feature extractors and these are used in the target domain to derive versatile features.Food detection and classification is an extremely complex task and researchers often resort to transfer learning to achieve this task and this has led to decent accuracy in the previous research works. Some of them are discussed below.

To test the efficiency of pre trained models, Özsert Yiğit and Özyildirim (2018) in their research compared the performance of a Deep Convolutional Neural Network (DCNN) algorithm which was built from scratch against that of pre-trained models like AlexNet and Caffenet on a combination of two datasets namely, Food 11 and Food 101. In this research, three different DCNN models with different gradient descent optimization methods were applied. Since, the DCNN models were newly built, it required a larger dataset than the collection of Food 11 and Food 101. It was seen that the pre-trained models showed better accuracy and performance as compared to the newly built DCNN models. As reported in this research, the advantage of using transfer learning methods is since they are already trained on a larger dataset they provide much higher accuracy even when trained on a relatively smaller dataset like Food-101. In this research, I am using the Food-101 dataset, hence it is ascertained that pre-trained models should be a good starting point to achieve image detection.

In another research performed by Memiş et al. (2020), a comparative study was performed to analyze the performance of various deep learning methods.This study was performed on the UEC Food-100 dataset using deep learning methods like ResNet-18, Inception-V3, Resnet-50, Densenet-121, Wide Resnet-50 and ResNext-50. The comparative study achieved the highest accuracy on InceptionV3 and ResNext-50 models. In researches performed by Yanai and Kawano (2015), it was seen that deep learning methods achieved state of the art performance with the help of transfer learning on CaffeNet which is a DCNN network. In other researches done by Szegedy et al. (2016), Inception-V3 model was fine tuned to achieve nearly 82% accuracy on the UEC dataset. Hence, it is seen that the InceptionV3 model trained on ImageNet weights achieve over 80% accuracy in some of the previous researches done in this field.

2.3 Investigation of Calorie Estimation Techniques

Accurately building a dietary management system involves estimating the overall calorie intake. This can be done by estimating the portion size from the food image supplied as input. This is rather a ill poised problem and can be extremely complicated to execute accurately. Hence, other methods for calorie estimation needs to be investigated to develop a easier and more user friendly approach.

Lo et al. (2020) in their research paper, presented a detailed review of the various processes researchers have proposed from 2009-2019. The process, advantages and disadvantages of five approaches namely, Stereo-based approach, Model-based approach, Depth Camera based approach and Perspective transformation approach has been discussed in this paper. However, the efficacy of most these approaches are directly proportional to the kind of dataset they are applied on. Hence, there is no one rule fits all scenario here. For example, Stereo-based approach involves using at least two images which are captured by a moving camera to recreate a 3D structure of the item. Another method in this approach involves using a binocular camera to construct a 3D image. Similarly, depth camera based approach involves the usage of a depth camera in which a 3D sensor system is used to capture an aerial view of the food item thereby computing the final volume. Such approaches even though provide more spatial information about the item but requires specialised hardware and an integrated system which is not often available for an end user.

To overcome this problem, model based approaches are suggested in which pre-trained models or templates are used to compute the final volume. The limitation is that if a particular image is not available in the dataset the model is trained on, then that image will not be recognized and a large estimation error will be introduced. Perspective transformation based approach also involves having an integrated system in wearable devices or mobile phones so that volume can be estimated from the RGB images captured by them. This involves a cost and power constraint which can be a ineffective for the end users. Hence, a simple to use, scalable system is necessary enabling end users to get calorie information effortlessly.

Researchers also explored the use of Monocular Depth-Prediction Networks to estimate the food volume from images which is instrumental in determining the calorie. Graikos et al. (2020) in their research, employed a state of the art, monocular depth prediction network that was primarily trained on videos obtained from EPIC-KITCHENS dataset.A depth estimation network was proposed which was trained on monocular sequences of a video. The depth estimation output is fed into a depth map and eventually a 3D point cloud representation is created to estimate the volume of the item. This method works only on images which are placed on a plate and the images are taken from an aerial view.

Several other researchers such as Yue et al. (2020) and Aladem et al. (2019) have proposed various methodologies based on monocular depth networks that include creation of a point cloud representation to estimate the food volume. Hassannejad et al. (2017) proposed that user takes short videos of their meals and an automatic extraction of key frames from those videos would be combined to create a point cloud representation. A major drawback of such a method is that users need to constantly carry items for calibration and for taking images from different angles. To overcome this drawback, Meyers et al. (2015) proposed the use of DCN networks in which each pixel of images captured by users would be projected to a point in 3D space. Subsequently, a rough estimation of the food item would be constructed to determine the volume. But this process once again has a major drawback because it requires a lot of ground work and images need to be captured in a particular way which can be troublesome for users.

Since,this research aims at creating an easy solution for the end users and keeping these drawbacks in mind, a more simplistic approach to determine the calorie content of detected food images was required. One such simplistic approach was undertaken by (Ayon et al.; 2021). In this study, researchers focused on creating a web based application in which the food image uploaded by the user was detected and their equivalent calories were displayed as output. To detect the food from inputted image, a model focused on CNN network was proposed.The feature extraction was performed using Inception V3 model. Another dataset with the base calories of various food items were also loaded simultaneously. After which the detected food item was looked up in the calorie table and eventually based on portion size, the overall calorie of the detected food item is displayed to the users. This is a simple yet accurate method of calorie estimation. The major drawback of this system is that calorie estimation is mainly dependent on the secondary dataset which firstly needs to be constantly updated and secondly the data needs to be highly accurate otherwise can lead to misleading results.

2.4 Investigation of Food Recommendation System

A part of this research also focuses on building a recommendation system which aims to provide the users with alternatives based on the detected food items having similar or comparable calorie content. There are various algorithms that can be deployed to achieve this and some of them are discussed below. Food recommendation system has emerged to be an active area of research because customized alternatives can be provided based on the health profile of the user.

People suffering from diseases like thyroid or diabetes often require specialised diet and nutrition depending on their health profiles. To automate the calorie intake and to keep a track of the same, systems are being designed by researchers that would tell the user if that particular food item should be consumed or not. Vairale and Shukla (2021) in their research paper proposed the use of a Content-based KNN method to achieve recommendation of food items for thyroid patients. In this research, a framework is proposed that identifies various food items based on content based features. Unique food characteristics are studied with the help of a privately built model using this framework. KNN algorithm has been used to generate food recommendation list for thyroid patients using food ratings and similarity score. The proposed model is reported to achieve 93% accuracy and outperforms the traditional KNN model.

Similar research was performed by Sowah et al. (2020) to build a diabetes management system for users. Once again KNN algorithm was used on the images detected with Tensorflow. In this research, researchers further used cognitive sciences to build diabetes related question and built a chatbot for the same. The model reportedly performed well with high accuracy and successfully answered personalised questions in human like way. However, the key takeaway from the researches is the implementation of KNN algorithm to design the recommender system and the same can be implemented in this research to generate general recommendation of food items that user can try based on the detected food item.

Other methods include using Self Organising Maps (SOMs), K- means clustering or Associate Rule Mining (ARM) algorithms to develop a food recommender system. Premasundari and Yamini (2019) in their research used K-means clustering and ARM techniques to develop food and therapy recommendation system for autistic patients. The K-means algorithm has been implemented to group the syndromes based on the type of disorder of the patient. After clustering, ARM has been implemented for the recommendation system of food based on these syndromes. The proposed system reportedly achieved satisfactory results. However, the researchers report a potential scalability issue with this recommendation system as the algorithm needs to be applied separately to each cluster.

2.5 Comparison of Reviewed Techniques Used in Food Image Classification

Results of various techniques employed by other researchers were compared and tabulated as shown in Table 2. The application of Inception-V3 model as done by Rajayogi et al. (2019) achieved the highest accuracy of 87.9%. Similar methods used by Memiş et al. (2020) also achieved nearly equal accuracy with ResNext-50 model. Even though both are comparable models, for this research transfer learning based on InceptionV3 model will be used as it has proven accuracy on the food-101 dataset. Other researches like that of Fahira et al. (2019) show that traditional classification algorithms can also provide high accuracy in image classification given that it is backed by excellent feature extraction. Kernel learning done by Joutou and Yanai (2009) achieves 61.34% accuracy but it paves a new path for food image classification.

Through the review of previous work done in this field and some identified gaps in those works, it is clearly evident that there is a need to develop a system which not only detects food labels from images but also determines their calorie content and recommends alternate products to the user. Thereby, answering the research question in section 1.2 and research objectives in section 1.3. This section also satisfies research objective 1 which was to critically evaluate some previous work done in this domain.

Table 2: Comparative Analysis of Literature Review

Authors	**Classifiers and Techniques**	**Advantages**	**Disadvantages**
Rajayogi et al. (2019)	InceptionV3, VGG16,VGG19, ResNet	Achieved high accuracy of 87.9% with relatively smaller dataset. Less computational time. No overfitting of data.	Data processing to be done to remove image noise. Can be trained on larger dataset to reduce loss rate.
Şengür et al. (2019)	Pre trained AlexNet and VGG16	Combination of AlexNet and VGG16 led to better feature extraction leading to accuracy as high as 79.86%.	This model is evaluated on relatively smaller datasets and there is no proof that it can be equally effective on larger datasets as well.
Özsert Yiğit and Özyildirim (2018)	Combination of AlexNet and CaffeNet	Higher accuracy of pre-trained models when compared to other methods developed from scratch.	The research does not explore the performance of the model developed from scratch with higher number of images.
Fahira et al. (2019)	LDA, Logistic Regression, Decision Tree and Random Forest	Traditional classification algorithm especially Decision Tree backed with histogram feature and Gabor filters produced accuracy of 98.85%	Suitable only for small dataset
Memiş et al. (2020)	ResNet-18, Inception-V3, Resnet-50, Densenet-121	InceptionV3 and ResNext-50 outperformed other models with over 80% accuracy. Less computational efficiency and limited hardware required	Suitable for very small dataset
(He et al.; 2017)	R-Mask CNN	Has better performance than existing single model entries on various tasks. It is easy to train and also reduces test time.	Involves expensive computational alignment procedures.

3 Methodology Approach Used

3.1 Introduction

This section focuses on the research methodology which in this research follows KDD methodology. The main motivation of this research is to develop an efficient diet management system for which a 3-tier architecture is proposed and implemented in the subsequent sections. A modified KDD approach has been undertaken in this research.

3.2 Food Detection and Calorie Estimation Research Methodology Approach

The research methodology is shown in Figure 1. It consists of the following stages.

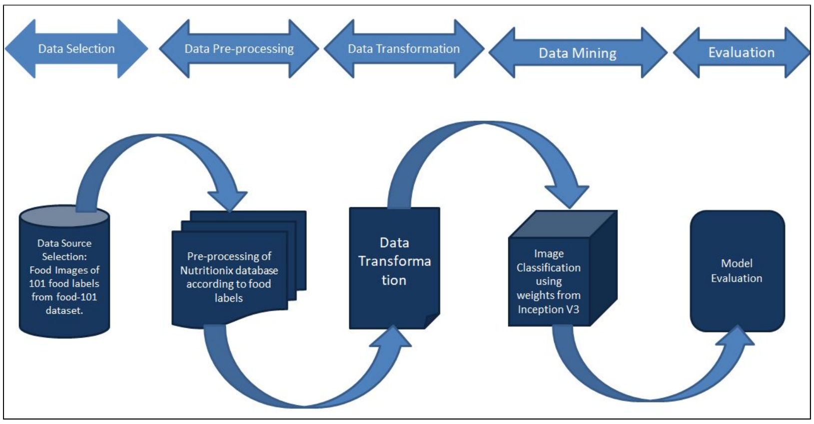


Figure 1: Research Methodology Architecture

3.2.1 Data Collection

Data selection which in this case is food images were collected from the food-101 repository which was created by Bossard et al. (2014) and is publicly available. The data is divided into various class labels and the images are in .jpeg format. There are 101,000 images across 101 class labels. To estimate the calorie of detected food items another dataset with calorie information of various food labels were collected from Nutritionix database. The Nutritionix database has an API through which the data was scraped and stored in a csv file. For this purpose the nutritionix library was installed and used.

3.2.2 Data Pre-processing

The food-101 data repository mainly consists of food images. Hence, no extensive data cleaning and pre-processing of the data was required. The Nutritionix database is highly extensive and contains a wide variety of food labels. Owing to computational restrictions and in order to reduce complexity, food labels which are not useful for our purpose was removed. Also, the name of the items were renamed according to the detected food labels so that they can be easily mapped together. This was done using pandas. The csv file was read such that it can be treated as a dataframe which made working on the file easier.

3.2.3 Data Transformation

The authors of the food-101 dataset have divided the data into 75,750 training images and 25,250 test images but this is in the form of txt files. Hence, the os library was used to make the directory and a helper function was created to divide the data into train and test folders according to the txt files. Owing to their different sizes, the food images in the dataset were also resized to 299X299 and rescaled to pixels ranging between [0-1] using keras so as to bring them to a constant size. In the calorie dataset some of the items had missing values in terms of nutritional information. Since these are critical information, they were not imputed. Instead, the values were filled in from other sources from the internet.

3.2.4 Data Mining

Transfer learning using the weights from the pre-trained Inception V3 model was applied on the food images to facilitate multi class classification of food labels. Tensorflow was used to train the final model on the food dataset. The recommender system to suggest alternatives to the user was built using KNN algorithm.

3.2.5 Data Interpretation and Evaluation

After determining the class label it was mapped with the calorie dataframe so as to retrieve the calorie information. The recommender system suggests food items to the user which has similar nutritional content as that of the detected food item. Training accuracy, validation accuracy, training loss and validation loss are the main metrics on which the efficacy of the applied InceptionV3 model was evaluated.

3.3 Project Design Process Flow

The design process of the project is depicted in Figure 2. The design process has been divided into i) decomposition layer, ii) logic layer and iii) evaluation layer. In the decomposition layer, the dataset collected from the various sources are stored in google drive. In the logic layer, the implementations of various models are done and the quantitative analysis, evaluation and visualization of this layer is done in the evaluation layer.

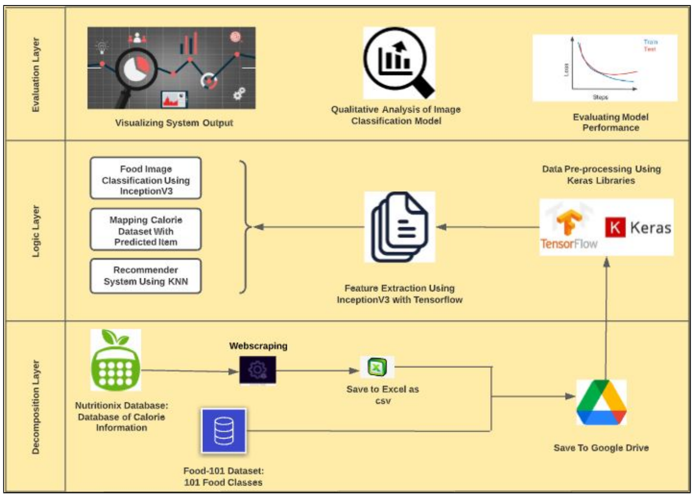


Figure 2: Project Design

3.4 Conclusion

KDD methodology has been used in this research based on which the project flow has been designed. The food images has been collected from the food-101 dataset and the Nutritionix database has been used to generate calorie estimation of detected food labels. The blueprint of the research and implementation of classification models to detect food items and their calories and their subsequent results and evaluation are discussed in the following sections.

4 Design Specification

4.1 Introduction

In this section the blueprint of the implementation of the research is discussed. The libraries required and steps undertaken to answer the research question is discussed in details in the subsequent sections. The project is also implemented following the flow chart as shown in fig 3

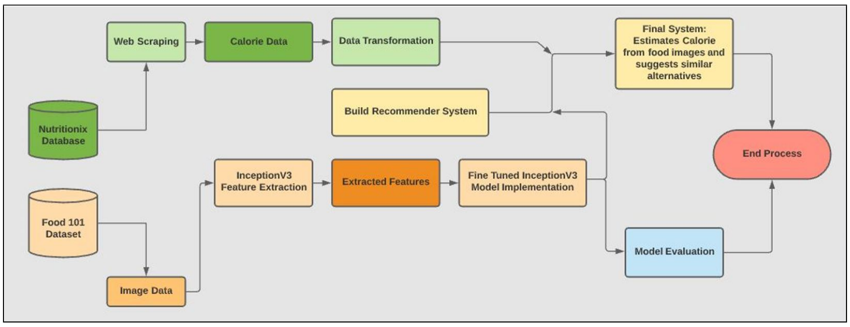


Figure 3: Design Blueprint

4.2 Tool Used for implementation

This project has been fully implemented on the Google Colaboratory. Google Colab is an excellent tool that hosts a Jupyter notebook and does not require any setup. It is tailor made for performing deep learning tasks because tensorflow and keras is pre-installed in this and can simply be imported. I have used the regular colab version which gives access to GPU on a standard RAM to meet the extensive computational requirements of neural networks.

4.3 Tools for Model Creation

TensorFlow is an open source Python library which helps in fast numerical computing. It is an artificial intelligence library that uses data flow graphs to build a model and it is mainly used for tasks like classification, prediction creation etc.

In this project tensorflow has been extensively used to leverage various tasks. Firstly, the load_model library from the tensorflow.keras.models package is used to load the best model. The model with the highest accuracy is saved as a hd5 file. Using the load_model, the configuration of the best model can be accessed anytime. Other libraries such as InceptionV3, Sequential, Model, Dense, Dropout, Activation, Flatten, Convolution2D, MaxPooling2D, ZeroPadding2D, GlobalAveragePooling2D and AveragePooling2D are also imported to retrieve the weights and the different layers of the already trained Inception model. These libraries are important to facilitate transfer learning and the food dataset can be trained on these pre-trained weights. ImageDataGenerator is imported from the tensorflow package to work on pre-processing of the image data like resizing and rescaling of the image. Finally ModelCheckpoint and CSVLogger libraries are imported to set up checkpoints and maintain logs of the model. This helps in keeping a track of model execution so that in case of interruptions like runtime disconnect, the model can be retrieved from its last execution and further execution can resume.

4.4 Convolutional Neural Network

Convolutional Neural Network or CNN is a deep learning technique that is mainly used in image classification tasks. CNN uses pooling to reduce the dimensions so that less data needs to be processed, thereby, reducing the processing time and power required to do the task. There are multiple layers in a CNN network and the output of the previous layer is feeded as input of the next layer. The images are made suitable for multilayer perceptron and after that CNN flattens the images into column vectors. The flattened images are fed into the neural network which is called a feed forward network. Back propagation techniques are used to trace errors and to analyze the errors that might occur in the modelling process. With every iteration of the training data, this process is repeated. Back propagation goes back to the hidden or inner layers after calculating the total error so that the weights can be adjusted to decrease the error. Until the desired accuracy is achieved, this process is repeated by back propagation.

4.5 Image Pre-processing with Keras

Image pre-processing in this research is done with the help of ImageDataGenerator. It is a class under keras that allows data augmentation while the model is in execution and in real time. It generates batches of tensor image data. Any transformation is directly applied on each of the training images as they are directly passed to the model. This reduces the overhead memory and makes the model more robust. Augmentations such as resizing and rescaling the images were done using this class.

4.6 Feature Extraction Using InceptionV3 model

Convolutional Neural Networks form the core of most state-of-the art computer vision solutions for various different tasks. Rethinking ways on scaling up the network to achieve more computational efficiency by utilizing additional resources efficiently led to the development of InceptionV3 network (Szegedy et al.; 2016). The goal was to modify the existing inception network by reducing the computational complexity. InceptionV3 is an extension of GoogleNet that achieved excellent classification accuracy for biomedical applications. It is a 48 layer network that outruns VGGNet in terms of computational efficiency. Changes in the inception network are registered only when computational changes are not lost. Factorized convolution, dimensionality reduction etc. are some of the techniques that has been included in to optimize the inception model due to which the constraints are lost and an easier model is obtained.

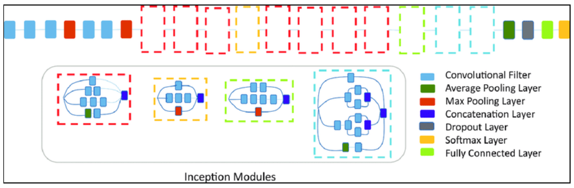


Figure 4: Inception Modules Ding et al. (2019)

4.7 KNN Algorithm For Building Recommender System

K-nearest neighbour or KNN is extensively used as a classification algorithm. It follows the principle of nearest neighbors in which to judge or classify an object, similar objects are searched from the entire training set that has comparable attributes with the object to be classified. KNN algorithm can refer to attributes of multiple objects while determining the category of an object. This value depends on the value of K which is set while training the algorithm.

In this research, KNN algorithm has been used to build the recommender system that suggests alternate food items to the user having similar nutritional content as that of the detected food item. Ball tree algorithm has been used and the euclidean distance has been set as the metric while building the recommender system. The ball tree or metric tree is a data structure that partitions space so that various points can be organized in a multi dimensional space. The name comes from the fact that it partitions data points into a set of hyperspheres or "balls". The data structure which results from this is useful in a variety of applications, most notably, neares neighbor search which is the basis of the KNN algorithm.

The subsequent section explains how these tools and techniques were implemented.

5 Implementation

5.1 Introduction

In this chapter, implementation of the project to achieve the research objectives and answer the research question is discussed. The chapter also sheds light on the chosen algorithm, feature extraction and data extraction techniques that were employed supported with subsequent visualizations.

5.2 Creation of Food Dataset

In this subsection, research objective 2 will be resolved. The first step to answer this objective was to collect a food dataset containing images from multiple food categories. The dataset was constructed from food-101 dataset with 72000 images in total from 72 food classes. Both the training and validation dataset was thoroughly checked to ensure that the food classes corresponded to the correct food images. The 72 food classes used include:

- Apple Pie, Baklava, Beef Tartare, Beet Salad, Bread Pudding, Breakfast Burrito, Caesar Salad, Caprese Salad, Carrot Cake
- Cheese Plate , Cheesecake , Chicken Curry , Chicken Quesadilla , Chicken Wings , Chocolate cake , Chocolate mousse , Churros , Club sandwich

- Crab cakes, Cup cakes, Deviled eggs, Donuts, Dumplings, Eggs benedict, Falafel, Filet mignon, Fish and Chips
- Foie Gras, French fries, French onion soup, French toast , Fried Calamari, Fried rice, Frozen yogurt, Garlic Bread, Gnocchi
- Greek Salad, Grilled Cheese Sandwich, Grilled Salmon, Hamburger, Hot and Sour soup,Hot Dog, Hummus, Ice cream, Lasagna
- lobster roll sandwich, macaroni and cheese, macarons, mussels, Nachos, Omelette, Onion Rings, Oysters, Pad Thai
- Pancakes, Panna Cotta, Pizza, Pork Chop, Pulled Pork Sandwich, Ramen, Ravioli, Red Velvet Cake, Risotto
- Samosa, Spaghetti Bolognese, Spring Rolls, Steak, Strawberry shortcake, Sushi, Tacos, Tiramisu, Waffles

5.3 Data Preprocessing

In this subsection, data pre-processing stages implemented before extracting features from the food images is discussed. The necessary libraries were imported and the directory was set for further implementation. This was followed by resizing all the images to a common size because different images had different sizes in the original dataset. Tensorflow, os and numpy as np, matplotlib.pyplot as plt etc. are some of the libraries that were imported to facilitate data pre processing. The next step involved setting the directory so that I could access the images and prepare them for feature extraction. The initial dataset was stored in Google Drive under food-101 folder. I used Google Colab for implementing the project. The notebook was mounted on the drive and two folders, train and test was created to split the dataset for training and testing purpose. The final step involved resizing amd rescaling the images using ImageDataGenerator which is a keras library. The images were resized to 299X299, that is, Image Width = 299 and Image height = 299 and rescaled or normalized from a pixel range of [0-255] which is quite sparse to a pixel range [0-1].In this pre-processing stage research objective 2.1 and 2.2 was resolved.

The overall dataset was split into training and testing purposes. The authors provided both the sets in the form of ṫxt files. Appropriate coding was done so as to create separate folders to store the train and test data. The training dataset had 72000 images in total while the test dataset had 14400 images to facilitate validation thereby meeting research objective 2.3

5.4 Feature Extraction and Model Implementation

Initially, the basic Inception model with 'imagenets' weight was downloaded on which a global spatial average pooling layer, a fully connected layer and a logistic layer with 120 classes were added. Post this the base model was fine tuned. A dense layer was added along with a L2 regularization function with a lambda hyper-parameter value of 0.005 and softmax activation.. This was done to reduce overfitting of the model. Finally, the model was compiled with an SGD optimizer having a learning rate of 0.0001 and momentum of 0.9. "Categorical Crossentropy" was set as loss and the metric for evaluating the model was accuracy.

The features extracted from the images were stored as weights in the InceptionV3 model. These weights were further used to train my CNN model on the training dataset. Dense layer was used for predicting the class. For the dense layers, ReLu activation function was used so that the model learns faster and performs better. The best model was saved for future work and in case of any interruptions that could occur during model execution. Batch size was set to 75 and the model was executed for 45 epochs.

5.5 Calorie Dataset Collection

The next step of the research involved determining the calorie content of the detected food item. To facilitate this, a calorie dataset was constructed containing the nutritional information of all the food labels available in the food dataset. USDA dataset provides the most extensive nutritional information. Similarly, Nutritionix database is another such database which is built on the USDA dataset. The application also provides an API key with which information regarding any food class can be downloaded. I applied for the API key which was later provided to me.

Nutritionix package was installed as the first step. The API key was further used to webscrape data from the Nutritionix database for the different labels available in the food dataset. Information like serving size, Calorie, Fat, Cholesterol, Carbohydrate, Sodium and Protein is collected and the same is stored in a csv file so that they can be mapped with the detected food item. With this research objective 4.1 in Chapter 1, section 1.3 was fulfilled.

5.6 Implementing Recommender System

The recommender system was built to suggest alternative food items to the users having comparable nutritional value with that of the detected item. K Nearest Neighbours (KNN) algorithm has been implemented to achieve this.

Initially, the required libraries were imported which include, distance from scipy.spatial package, NearestNeighbors from sklearn.neighbors package and joblib from sklearn.externals package. The number of neighbors were set to 5, ball_tree algorithm was used to compute the neighbors and metric was set as eucledian. The KNN algorithm was set up up such that 3 neighbors or food items are returned as output or recommended products for the user.

5.7 Deploying Model With Calorie Dataset and Recommender System

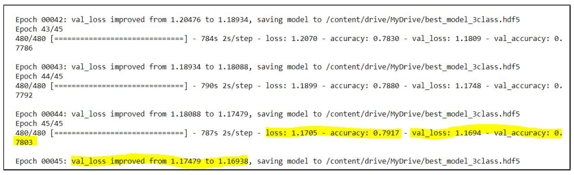
A helper function, "final_system" was created so as to bring the calorie dataset and recommender system together with the image classification output. The predicted class label from the inputted image was retrieved and stored in a variable. The calorie dataset csv was accessed with the pandas library and was stored as a dataframe. The predicted class label was looked up in the calorie dataframe and the nutritional information was retrieved. This information was further fed into the recommendation system so that the neighbors or products that share the same cluster of nutritional information with that of the predicted item can be retrieved as output. Next step involved printing the output in a tabulated format. The output was designed such that the inputted image and the detected class label was shown along with the nutritional information in tabulated format. The 3 recommended products were also shown as output as shown in figure 7.

5.8 Uploading Image for Validation

To validate the entire implementation, random images were chosen from the internet for different food items using wget that was later uploaded and fed into the helper system to analyze the output.

6 Evaluation

This section focuses on the evaluation of the implemented method. Accuracy and loss are the main metrics on which the implemented model is evaluated. The model was executed for 45 epochs on 72 classes of food. While fine tuning, the Inception model was compiled with the metrics set to accuracy. So with the execution of each epoch the accuracy and loss is recorded. The final model was executed for 45 epochs. The model had a training accuracy of 79.17% and validation accuracy of 78.03%. While the training loss and validation loss was recorded to be 1.1705 and 1.1694 respectively.


```
Epoch 00042: val_loss improved from 1.20476 to 1.18934, saving model to /content/drive/MyDrive/best_model_3class.hdf5
Epoch 43/45
480/480 [==============================] - 784s 2s/step - loss: 1.2070 - accuracy: 0.7830 - val_loss: 1.1809 - val_accuracy: 0.
7786

Epoch 00043: val_loss improved from 1.18934 to 1.18088, saving model to /content/drive/MyDrive/best_model_3class.hdf5
Epoch 44/45
480/480 [==============================] - 790s 2s/step - loss: 1.1899 - accuracy: 0.7880 - val_loss: 1.1748 - val_accuracy: 0.
7792

Epoch 00044: val_loss improved from 1.18088 to 1.17479, saving model to /content/drive/MyDrive/best_model_3class.hdf5
Epoch 45/45
480/480 [==============================] - 787s 2s/step - loss: 1.1705 - accuracy: 0.7917 - val_loss: 1.1694 - val_accuracy: 0.
7803

Epoch 00045: val_loss improved from 1.17479 to 1.16938, saving model to /content/drive/MyDrive/best_model_3class.hdf5
```

Figure 5: Loss and Accuracy on 45 Epochs for 72 Categories

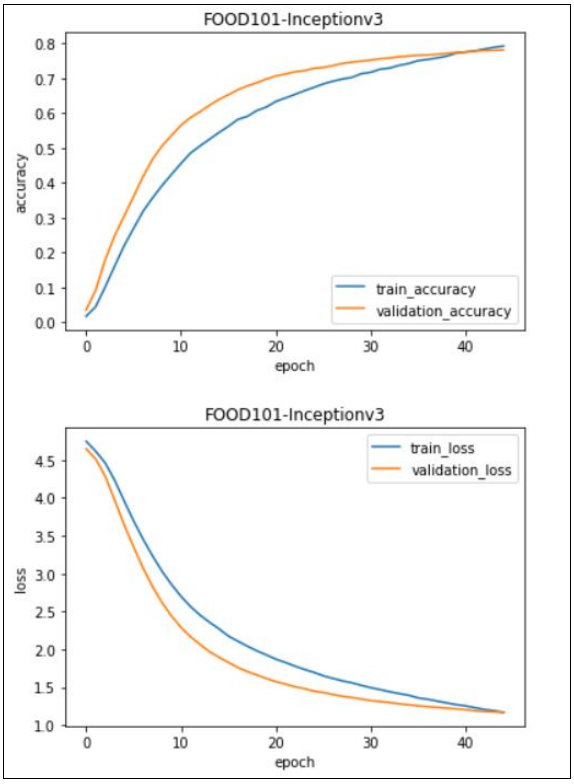


Figure 6: Loss and Accuracy Plots on 45 Epochs for 72 Categories

6.1 Analyzing Outputs

Multiple images were selected since this is a multi-class food identification system. The images were saved in a list and the list was fed into the helper function to compute the output. The output of the system is shown in figure 7

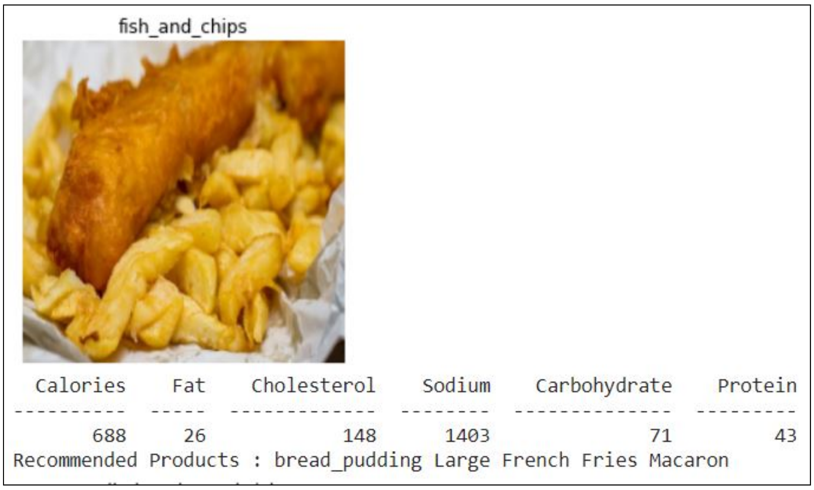


Figure 7: System detecting fish and chips

It is visible from figure 7 that the image inputted for validation was fish and chips which the system determines correctly. The nutritional information for fish and chips is given which shows Calories: 688, Fat: 26, Cholesterol: 148, Sodium: 1403, Carbohydrate: 71, Protein: 43. 3 recommended products are also shown as output that shows the user can also try bread pudding or large french fries or macarons as they have similar nutritional content as that of fish and chips.

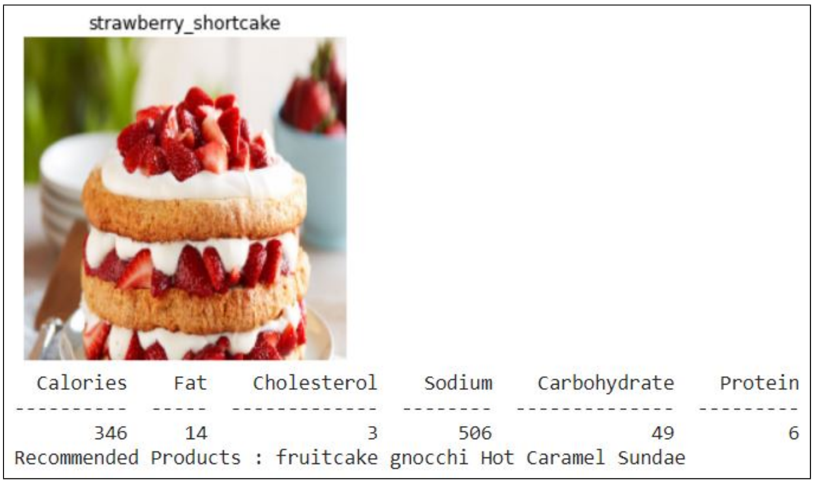


Figure 8: System detecting strawberry shortcake

Figure 8 correctly identifies the food label strawberry shortcake and the equivalent calorie information is also printed. Strawberry shortcake has Calories: 346, Fat: 14, Cholesterol: 3, Sodium: 506, Carbohydrate: 49 and Protein: 6. Users having shortcake can also go for fruitcake, gnocchi or hot caramel sundae which is quite understandable and realistic given that they fall in the same category like that of the detected class label. The above outputs show the development of a system which is a culmination of food image classification along with calorie estimation and recommender system. Therefore, objectives 4.2 and 5 set out in section 1.3 were accomplished.

In the subsequent sections, the experiments and results of experiments which were done prior to reaching the final model has been discussed in details. The results of the final model has also been discussed in details in Section 6.6. Table 3 shows a comparative analysis of the various experiments performed that gives a concise understanding of the output.

Table 3: Comparison of Results

Number of Classes	Number of Epochs	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
101	10	24.37%	32.64%	4.1220	3.8289
72	15	52.94	64.46	2.33	1.88
72	45	79.17%	78.03%	1.1705	1.1694

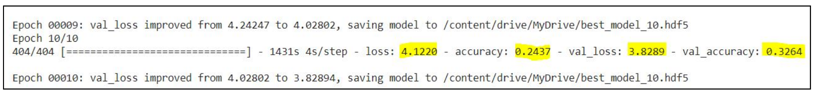
6.2 Case Study 1

The first experiment was performed with the R-Mask CNN model. In recent researches, some researchers have found this model to be particularly effective in food detection and calorie estimation. However, as reported in 2 this algorithm is computationally very exhaustive.

The same happened when I tried to execute it. Each epoch took nearly 12 hours on GPU connection in google colab and the model would invariably fail after a couple of epochs due to runtime disconnect. I tried changing the batch size only to have the same result. Hence, due to computational restrictions implementation of the model was dissolved and transfer learning using InceptionV3 model was undertaken.

6.3 Case Study 2

During the first run of the InceptionV3 model, all the 101 categories of the Food-101 dataset was used to train the model. Due to the high volume of data and computational restrictions, the dataset was downsampled to 500 train images and 200 test images per class. The model was executed for 10 epochs and the accuracy and loss values were noted as shown in figure 9. The accuracy and loss plots were also plotted as in figure 10 to have a better idea about the model. In this run, the model had a training accuracy of 24.37% and validation accuracy of 32.64%. The training and validation loss were also quite high at about 4.1220 and 3.8289 respectively.


```
Epoch 00009: val_loss improved from 4.24247 to 4.02802, saving model to /content/drive/MyDrive/best_model_10.hdf5
Epoch 10/10
404/404 [==============================] - 1431s 4s/step - loss: 4.1220 - accuracy: 0.2437 - val_loss: 3.8289 - val_accuracy: 0.3264

Epoch 00010: val_loss improved from 4.02802 to 3.82894, saving model to /content/drive/MyDrive/best_model_10.hdf5
```

Figure 9: Loss and Accuracy on 10 Epochs for 101 Categories

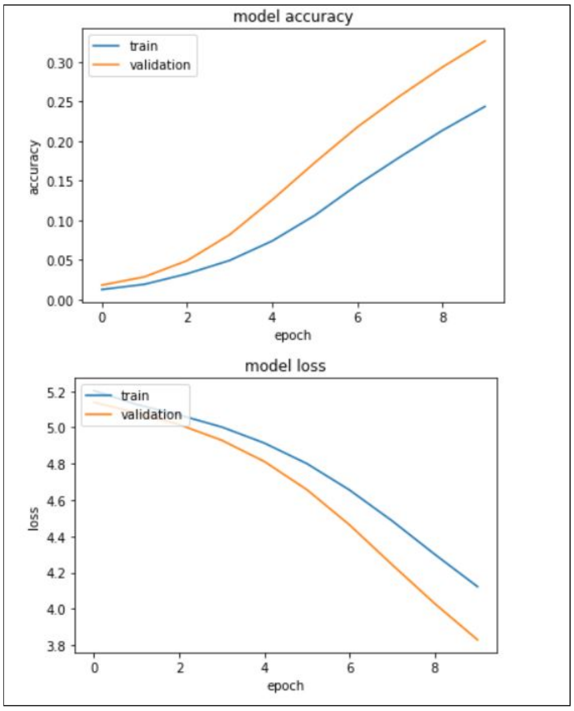


Figure 10: Loss and Accuracy Plots on 10 Epochs for 101 Categories

6.4 Case Study 3

Given the significantly low accuracy of the model with 10 epochs, all the 101 categories were once again trained for 40 epochs. Unfortunately, due to the high number of classes and corresponding volume of data, the model kept on terminating by throwing an error. Since, with regular colab version there is no provision of getting a higher RAM , I tried to reduce the data volume by downsampling. Once again, the training and testing data was sampled by 500 and 200 images respectively for each class. I tried to execute the model again with multiple epochs such as 20,25,30 only for it to end up into a runtime exception.

6.5 Case Study 4

With three unsuccessful attempts at working with all the categories of food in the dataset, I chose to reduce the number of categories. In this process, randomly a list of 72 categories of food were chosen on which further research progressed. Once again, a base model was executed with 15 epochs and downsampled data to check the model performance. The model attained training accuracy of nearly 53% while the validation accuracy was much higher at 64% as shown in figure 11. For 15 epochs this model performed pretty well because not only it correctly classified existing images but also new images. Figure 12 shows the accuracy plots and proves the efficacy of the model. The loss graphs however seem to be sloping which goes to show that the model might have a slow learning rate. Nevertheless, to get a better accuracy, the model was further executed for 45 epochs.

```
Epoch 00014: val_loss improved from 2.06707 to 1.98278, saving model to /content/food-101/best_model_3class.hdf5
Epoch 15/15
480/480 [==============================] - 837s 2s/step - loss: 2.3318 - accuracy: 0.5294 - val_loss: 1.8887 - val_accuracy: 0.6446

Epoch 00015: val_loss improved from 1.98278 to 1.88866, saving model to /content/food-101/best_model_3class.hdf5
```

Figure 11: Loss and Accuracy Plots on 15 Epochs for 72 Categories

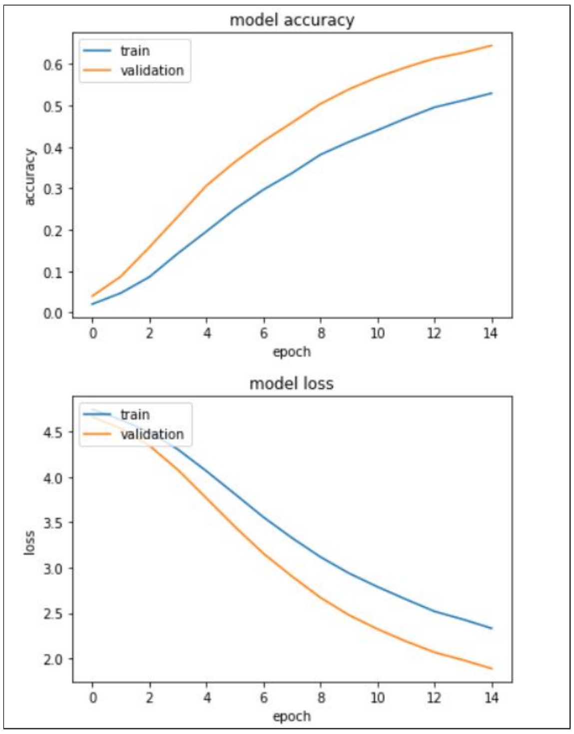


Figure 12: Loss and Accuracy Plots on 15 Epochs for 72 Categories

6.6 Discussion

In this research, majority of the focus has been cast on the performance of the image classification model, that is, the fine tuned inceptionv3 model because due to computational restrictions or incapability I was unable to execute the R-mask CNN model. During the case study 6.3, when the model was executed for 10 epochs for all the 101 categories, the validation accuracy of 32.64% was significantly higher than the training accuracy of 24.37%. This seems to be a decent model which goes to imply that the model does decently in classifying objects it hasn't been trained on. Similarly, from the loss plot it can be inferred that the validation loss was lesser than the training loss which concludes the same understanding. However, the overall accuracy of the model was still low and an improvement was tried by increasing the number of epochs even though it is not the most ideal way to improve accuracy. Unfortunately, due to computational restrictions once again I was unable to perform further experiments using the same configuration.

After reducing the number of categories to 72 and downsampling the train and test dataset, experiment **??** was performed. This was a pretty good model and it was able to classify or validate objects with a high level of accuracy in comparison to training accuracy. However, since the overall accuracy was low, it was presumed that with more number of epochs the accuracy would increase.

In line with this thought, the model was finally executed for 45 epochs. In this experiment, the model was seen to slightly overfit the data. A model overfits when it performs extremely well with the training data but the accuracy dips during validation of the data. In this model, the training accuracy is marginally higher than the validation accuracy as shown in Table 3. Also, if we look at the figure 6, it is clearly visible that the training and validation accuracy intersects each other at around 40 epochs. The loss curve depicts that the model had a good learning rate as it has some sort of exponential decrease. Overall, it can be concluded that for 72 classes, with 36000 training and 14400 test imagess, the fine tuned inceptionv3 model should be executed for about 35 epochs to have a good model with an accuracy around 75% because anything more than that will slightly render the model as overfit for the data.

From the above explanation, it is clear that executing the model for 45 epochs was not a good choice. Rather experiments should have been performed with lower number of epochs. Also, more number of images can be introduced in the classes so as to increase the number of training images rendering better learning of the model. Memiş et al. (2020) in their research reported a high accuracy of the InceptionV3 model which can be achieved for this project provided more number of images are supplied for training and validation. Therefore, even though the goal of developing a system that identifies food items from images and estimates the calorie to provide equivalent alternatives is met, the model can be re-executed with revised parameters to achieve even better accuracy.

7 Conclusion and Future Work

One of the major objective of this research was to develop a simplistic user friendly system that would make dietary management easier. The objective was to build a system that can easily detect the calorie content of a dish and recommend similar alternatives to the user. From this perspective, the following research question had to be answered. *"How can a combination of InceptionV3 and KNN algorithm be used to develop a system that not only estimates the nutritional information from food images but also suggests similar alternatives to the users?"* . Through this research I have been successful in answering this research question wherein, I have used pre-trained InceptionV3 and KNN model to achieve a system that correctly classifies food and its related calorie along with other nutritional information. It also, recommends alternative food items to the user based on the detected food item. This system works for 101 categories of food and a more extensive dataset can also be used if available.

The KNN algorithm that has been used here can be modified such that specialized nutritional information is provided into it. This will ensure that the system can recommend healthier alternatives of the ingredients detected by the food item. Also, personalised user profiles can be integrated so that the recommender system recommends products according to the user profile. This has not been implemented due to lack of computational capabilities but it can be attainable if it is advanced by another student. This system can also be integrated into android or ios, making it a mobile application which would enable the system to be commercialised. The datasets can be further enriched with a wider variety of food labels and hence, the application can be beneficial for a wide audience as it would render efficient dietary management as easy as taking a selfie.

The algorithm used in this research achieves decent results in terms of image detection and food recommendation. However, tests have revealed that the model overfits the data. A wide variety of techniques involving data processing, implementation of deep neural network, web scraping data and implementation of supervised learning techniques were exercised during the course of this research. Feature extraction using pre-trained model weights and applying them on the existing dataset results in a excellent accuracy and this utilization of existing models and the fact that they can be incorporated along with supervised algorithms like KNN are some of the key findings of this project. Having said that, the model also lacks in a particular sector which is determining the exact portion size of the detected food as this would require dedicated hardware. If this problem is overcome, then the calorie estimation would become even more accurate and the overall system would become much more efficient.

8 Acknowledgements

I would like to convey my sincere thanks and regards to my supervisor Dr. Majid Latifi for his constant and relentless support and guidance. longtable

References

Aladem, M., Chennupati, S., El-Shair, Z. and Rawashdeh, S. A. (2019). A comparative study of different cnn encoders for monocular depth prediction, *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pp. 328–331.

Allegra, D., Battiato, S., Ortis, A., Urso, S. and Polosa, R. (2020). A review on food recognition technology for health applications, *Health Psychology Research* **8**(3).

Ayon, S. A., Mashrafi, C. Z., Yousuf, A. B., Hossain, F. and Hossain, M. I. (2021). Foodiecal: A convolutional neural network based food detection and calorie estimation system, *2021 National Computing Colleges Conference (NCCC)*, pp. 1–6.

Bossard, L., Guillaumin, M. and Van Gool, L. (2014). Food-101–mining discriminative components with random forests, *European conference on computer vision*, Springer, pp. 446–461.

Darapaneni, N., Singh, V., Tarkar, Y. S., Kataria, S., Bansal, N., Kharade, A. and Paduri, A. R. (2021). Food image recognition and calorie prediction, *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–6.

Ding, Y., Sohn, J. H., Kawczynski, M. G., Trivedi, H., Harnish, R., Jenkins, N. W., Lituiev, D., Copeland, T. P., Aboian, M. S., Mari Aparici, C. et al. (2019). A deep learning model to predict a diagnosis of alzheimer disease by using 18f-fdg pet of the brain, *Radiology* **290**(2): 456–464.

Fahira, P. K., Wibisono, A., Wisesa, H. A., Rahmadhani, Z. P., Mursanto, P. and Nurhadiyatna, A. (2019). Sumatra traditional food image classification using classical machine learning, *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, pp. 1–5.

Graikos, A., Charisis, V., Iakovakis, D., Hadjidimitriou, S. and Hadjileontiadis, L. (2020). Single image-based food volume estimation using monocular depth-prediction networks, *International Conference on Human-Computer Interaction*, Springer, pp. 532–543.

Harris, D. M. and Guten, S. (1979). Health-protective behavior: An exploratory study, *Journal of health and social behavior* pp. 17–29.

Hassannejad, H., Matrella, G., Ciampolini, P., Munari, I. D., Mordonini, M. and Cagnoni, S. (2017). A new approach to image-based estimation of food volume, *Algorithms* **10**(2): 66.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017). Mask r-cnn, *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Joutou, T. and Yanai, K. (2009). A food image recognition system with multiple kernel learning, *2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 285–288.

Livingstone, M. B. E., Robson, P. and Wallace, J. (2004). Issues in dietary intake assessment of children and adolescents, *British Journal of Nutrition* **92**(S2): S213–S222.

Lo, F. P. W., Sun, Y., Qiu, J. and Lo, B. (2020). Image-based food classification and volume estimation for dietary assessment: A review, *IEEE Journal of Biomedical and Health Informatics* **24**(7): 1926–1939.

Memiş, S., Arslan, B., Batur, O. Z. and Sönmez, E. B. (2020). A comparative study of deep learning methods on food classification problem, *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, pp. 1–4.

Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J. and Murphy, K. P. (2015). Im2calories: towards an automated mobile vision food diary, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1233–1241.

Munsell, M. F., Sprague, B. L., Berry, D. A., Chisholm, G. and Trentham-Dietz, A. (2014). Body mass index and breast cancer risk according to postmenopausal estrogen-progestin use and hormone receptor status, *Epidemiologic reviews* **36**(1): 114–136.

Nishida, C., Uauy, R., Kumanyika, S. and Shetty, P. (2004). The joint who/fao expert consultation on diet, nutrition and the prevention of chronic diseases: process, product and policy implications, *Public health nutrition* **7**(1a): 245–250.

Özsert Yiğit, G. and Özyildirim, B. M. (2018). Comparison of convolutional neural network models for food image classification, *Journal of Information and Telecommunication* **2**(3): 347–357.

Premasundari, M. and Yamini, C. (2019). Food and therapy recommendation system for autistic syndrome using machine learning techniques, *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, pp. 1–6.

Rajayogi, J., Manjunath, G. and Shobha, G. (2019). Indian food image classification with transfer learning, *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, Vol. 4, IEEE, pp. 1–4.

Şengür, A., Akbulut, Y. and Budak, Ü. (2019). Food image classification with deep features, *2019 international artificial intelligence and data processing symposium (IDAP)*, IEEE, pp. 1–6.

Sowah, R. A., Bampoe-Addo, A. A., Armoo, S. K., Saalia, F. K., Gatsi, F. and Sarkodie-Mensah, B. (2020). Design and development of diabetes management system using machine learning, *International journal of telemedicine and applications* **2020**.

Sultana, F., Sufian, A. and Dutta, P. (2020). Evolution of image segmentation using deep convolutional neural network: a survey, *Knowledge-Based Systems* **201**: 106062.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016). Rethinking the inception architecture for computer vision, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Vairale, V. S. and Shukla, S. (2021). Recommendation of food items for thyroid patients using content-based knn method, *Data Science and Security*, Springer, pp. 71–77.

Wei, Y., Liang, X., Chen, Y., Jie, Z., Xiao, Y., Zhao, Y. and Yan, S. (2016). Learning to segment with image-level annotations, *Pattern Recognition* **59**: 234–244.

Yanai, K. and Kawano, Y. (2015). Food image recognition using deep convolutional network with pre-training and fine-tuning, *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, pp. 1–6.

Yue, H., Zhang, J., Wu, X., Wang, J. and Chen, W. (2020). Edge enhancement in monocular depth prediction, *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1594–1599.