T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

QUALITY ANALYSIS OF FOOD AND VEGETABLES WITH IMAGE PROCESSING

MASTER'S THESIS

Abdul Khalique BALOCH

Department of Artificial Intelligence and Data Science

Artificial Intelligence and Data Science Program

AUGUST 2022

T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

QUALITY ANALYSIS OF FOOD AND VEGETABLES WITH IMAGE PROCESSING

MASTER'S THESIS

Abdul Khalique BALOCH
(Y1913.140008)

Department of Artificial Intelligence and Data Science
Artificial Intelligence and Data Science Program

Thesis Advisor: Prof. Dr. Ali OKATAN

AUGUST 2022

ONAY FORMU

DECLARATION

The studies make up in this thesis were carried out in Department of Artificial Intelligence and Data Science, Institute of Computer Engineering, Istanbul Aydin University, and were supervised by Prof. Dr. Ali OKATAN. I hereby declare that all the information given in this thesis document has been achieved and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results. The thesis is carried out solely in the Istanbul Aydin University and has not been presented to any other institution.

Abdul Khalique BALOCH

FOREWORD

Initially I am feeling very grateful and thank my teachers and supervisors, Prof. Dr. Ali OKATAN for their valuable guidance and knowledge, time, and support throughout the period of this course and research work. This research work would not be possible without their help, supervision, and support. Moreover, I would like to thank teachers and colleagues at Department of Artificial Intelligence and Data Science, Institute of Computer Engineering, Istanbul Aydin University who helped me by providing their valuable suggestions.

Finally, it is due on me to say special thanks to my whole family members who have been supporting throughout my life, what I am today to undertake the research and write my thesis.

August 2022 Abdul Khalique Baloch

QUALITY ANALYSIS OF FOOD AND VEGETABLES WITH IMAGE PROCESSING

ABSTRACT

The quality analysis of vegetable and food from image is hot topic now a day, where researchers make them better then pervious findings through different technique and methods. In this research we have review the literature, and find gape from them, and suggest better proposed approach, design the algorithm, developed a software to measure the quality from images, where accuracy of image show better results, and compare the results with Perouse work done so for. The Application we uses an open-source dataset and python language with tensor flow lite framework. In this research we focus to sort food and vegetable from image, in the images, the application can sorts and make them grading after process the images, it could create less errors them human base sorting errors by manual grading. Digital pictures datasets were created. The collected images arranged by classes. The classification accuracy of the system was about 94%. As fruits and vegetables play main role in day-to-day life, the quality of fruits and vegetables is necessary in evaluating agricultural produce, customer always buy good quality fruits and vegetables. This document is about quality detection of fruit and vegetables using images. Most of customers suffering due to unhealthy foods and vegetables by suppliers so there is no proper quality measurement level followed by hotel managements. I have developed software to determine the quality of the vegetables and fruits by using images, it will tell you how your fruits and vegetables are fresh or rotten. Some algorithms reviewed in this thesis, including digital images, ResNet, VGG16, CNN and Transfer Learning grading feature extraction. This application used an open-source dataset of images and language used python, and designs a framework of system.

Keywords: Deep Learning, Computer Vision, Image Processing, Rotten Fruit Detection, Rotten Vegetables Detection, Fruits Quality Criteria, Vegetables Quality Criteria

GÖRÜNTÜ İŞLEME İLE GIDA VE SEBZELERIN KALİTE ANALİZİ

ÖZET

Görüntü işleme ile gıda ve sebzelerin kalite analizi, araştırmacıların farklı yöntem ve teknikler kullanarak araştırdıkları popüler konulardan biridir. Yapılan literatür araştırması sonucunda görüntü işleme ile gıda ve sebzelerin kalite analizine daha iyi bir yaklaşım ile oluşturulan algoritma ve daha kaliteli görüntüler kullanarak geliştirilen kalite analizi yazılımı sayesinde çalışma sonuçlarını Perouse çalışması ile karşılaştırdık. Araştırma analizinde kullanılmak üzere tasarlanan yazılım Python programlama dili ile tensor flow lite framework'ü kullanarak açık kaynak veri kümesi ile geliştirildi. Bu araştırmada yiyecek ve sebzeleri görüntüden sıralamaya odaklanarak onları sıralar ve derecelendirir böylece insanlar tarafından oluşabilecek hataları en aza indirgenmektedir. Dijital resim kümesi oluşturuldu ve yaklaşık %94 doğru sınıflama ile geliştirilen yazılım ile sınıflandırıldı. Meyve ve sebzeler günlük hayatın önemli bir parçası olduğundan insanlar tarafından kaliteli olanları seçilmektedir. Bu yüzden mevye ve sebzelerin kalitesinin değerlendirilmesi önemlidir. Bu amaçla araştırmanın ana konusu görüntü işleme ile gıda ve sebzelerin kalite analizi olarak belirlenmiştir. Otellerde sebze ve meyvelerin kalitesini ölçmek için kullanılan herhangi bir uygulama olmaması otel müşterilerinin sağlıksız yiyecek ve sebzeler nedeniyle acı çekmesine neden olmaktadır. Bu nedenle görüntüler kullanılarak sebze ve meyvelerin kalitesini belirlemek amacıyla bir yazılım geliştirildi ve böylelikle meyve ve sebzelerin en kadar taze ve çürük olduğunu belirle amaçlandı. Bu araştırmada dijital görüntüler, ResNet, VGG16, CNN ve Transfer Learning derecelendirme özelliği çıkarma dahil olmak üzere bazı algoritmalar kullanılmıştır. Bu araştırma Python programlama dili ile açık kaynaklı bir veri kümesi kullanarak tasarlanmıştır.

Anahtar Kelimeler: Derin Öğrenme, Bilgisayarla Görme, Görüntü İşleme, Çürük Meyve Tespiti, Çürük Sebze Tespiti, Meyve Kalite Kriterleri, Sebze Kalite Kriterleri

TABLE OF CONTENT

Page

ABBREVIATIONS

ML	**:** Machine Learning
AI	**:** Artificial Intelligence
SVM	**:** Support Vector Machines

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

This document is about analyzing the quality of fruit and vegetables with image processing. Leftover fruits and vegetables and many markets have expired and unhealthy fruit and vegetables. It's hard to know about the quality and health so we will design a solution with an image processing approach to measure the quality of fruit and vegetable, to find out which of them is fresh or rotten. The leftover food is wasted very much nowadays and is hard to measure for a common person because of hybrid quality of food. Peoples are unable to find the quality of food. People need to know what they have bought and what they are eating, how much nutrition, hygiene and minerals it contains. Also, people should be aware of the food contains essential amount of nutrients according to their age group and physical needs, identifying these elements is very hard nowadays without a proper hardware device and knowledge, most of the people can't afford to buy a device. In the results they are targeted by low and unhealthy food supply. The poor quality food and nutrition can contribute to stress, tiredness and low capacity of working; it is also risk of developing some illnesses and other health problems.

Our project works as an AI image processing using tensor flow library and trained machine learning models and algorithms. That detects the nutrients of food as well as the quality of food. Sometimes people buy the food without noticing its quality of food and do not analyse the quality of food due to lack of knowledge about healthy and unhealthy food. We have designed our project based on large data sets of images, that we have collected so many pictures of foods for image detection. So many times, it may create a conflict to recognize the food through pictures, it can happens because some foods having the same shape and color, in order to avoid this conflict, we have done too much research to collect large number of images.

Through tensor flow library we can import it into our python project and pass an image for processing from the dataset. This project is an offline desktop-based application, to compile images and measure the health of food. This project takes an image from inventory and processes it with our datasets to measure its colors, shapes and texture properties to identify the required results, then it checks the health according to our datasets. The results are returned as an array list to view the possible results of the food images. We can also improve our image datasets by saving the images which are being analysed by the software. This technique will speed up our image data collection but for this we must take our software online with a cloud server so that it can process our required results.

A. Background

The quality analysis of fruits and vegetable from image is hot topic now a day, where researchers make them better then pervious findings through different technique and methods. In this research we have review the literature, and find gape from them, and suggest better proposed approach, design the algorithm, developed a software to measure the quality from images, where accuracy of image show better results, and compare the results with Perouse work done so for.

The images also take importance in the agricultural sciences, where images are used to measure the product quality and other things. To produce the reports and analysis from such images, where some different kinds of method and mathematical formula are used.

Digital image processing technique helps for evolutions or grading to images, there are some tools and techniques which will improving images results. There are several works done so for in the area of image-processing application and some of developed for fruits and vegetable quality analysis. One by one we have defined of them and try to find gape from them and compare them with results of accuracy.

There are well known imaging applications are available, where just press keys and get results. In other part it is hard to products better results than pervious methods. In this research we review the literature and try our best propose a solution which addresses these problems. Recognition system is a challenge as compared to people's recognition level. The grading of food and vegetable would be useful in markets by a client. Food and vegetable grading using images can also be useful in computer vision. Picking out different kinds of vegetables is a routine task in markets, where identification of (i.e., banana, apple, pear) to determine price of them. The problem addressed, but most of the time consumers want to pick by themselves, but proposed approach presents some memorization of codes, which failing to result, this approach reduces over the time-consuming of users.

We have used several datasets of images to address the problem, to recognize food and vegetables from images basis of color and texture cues and generate a list of possible types of food or vegetable. The acquired images put in the system in single variety and random position, the given type of variety of vegetables done on site, the system achieve accuracy using some methods. In this section we high light some points which are related with the topic, issues and challenging with foods and vegetable identification from images

B. Research Problems

To analysis the quality of fruits and vegetable from images is hard task, where the freshness grading is one step assessments, such issues to handling of noisy image (M. K. Prem Kumar, 2020).

The common problems in digital pictures that are too light or too dark, these problems are usually caused by the system exposure and reading the background (Treadaway, 2007). Some challenges in image classification discussed like Intra-Class, Viewpoint, and Scale Variation (Gilani, 2020).

A detailed literature review proposed a compression summary for various facets of organically and normally grown vegetables and of post-harvest fruit quality grading (Mditshwa, 2016).

Another research evaluated fruit conditions on post-harvest condition, also works on temperature, humidity, and the impacts on fruit (Sivakumar, 2019).

In this research gives the freshness grading, first of this kind of research work with using different technique and better than other to our knowledge.

- Some of existed work done to addressed classifications, our approach for freshness grading.
- The best of our understanding there is no current research work done so for on topic freshness grading.

Also new thing in our research is the development of an application that reduces error rate and divided images into segment.

C. Research Contributions

Our findings address the issues above discussed by providing the support of the infrastructure of image processing proposed approach; there are three steps for the application. The key contributions of this hypothesis are two-fold.

1. The algorithm for the system has been designed.
2. Design, implement and evolution of the proposed approach which provides a better solution to the relevant area of the market.
3. Collection of datasets.

D. Outline of the Thesis

This thesis separated into bellowing chapters, chapter 2 describes the literature review, the research model development and hypothesis formation presented in chapter 3, proposed approach in chapter 4, followed by the implementation and compression results of proposed system in chapter 5, finally chapter 6 concludes the thesis and contributions of this research and a discussion of the future directions for the work undertaken.

II. LITERATURE REVIEW

In chapter two, we analysis the work done so for image classification, item recognition and disease-identification from images, vegetable and, food classification and disease-identification through image classification.

The previous work done on recognized colour and texture properties, others are investigated ne images, but here restrict to review this proposed for the classification of vegetables and foods.

A. Visualization of Images

The sorting between vegetables has been explored by researchers, the stages taken to supervised model by using image classification, the reprocessing feature which are followed by the majority of the researchers sorting.

The features of sorting of shape from images and its color with texture. The most of shape by size or area, the state of art comes in the deep learning techniques which apply in the images processing. Still, there is sufficient work done to improve the research.

B. Collection of Data

Collection of data is carried out from various sources: the Fruits Fresh and Rotten, the Vegetables Fresh and Rotten dataset from Kaggle and self-collected images. Manually collected data involved screenshot from stock image websites, as well as photos taken on our smartphones for later testing.

These transforms were chosen as they retain color information, saturation and brightness, which can do predictions of type and state. The augmentation resulted in approximately52,000 usable images, our model reached high final accuracies and we had the option of increasing the dataset if needed. We used a train/validation/test split of 64%, 16%, 20%.

C. Issues and Challenges with Image processing

The literature used to recognize vegetables and identify them with diseases, the issues and challenges serve as on basis to evaluate methods. The images contain more variety, some of variations in shape depending on the ripeness. Just sort fruits and vegetables from images are not sufficient therefore it is essential to cutting and chain those are usefull to generate specular reflections.

D. Recognition and Classification

The name 'Veggie-Vision' coined by Bolle et al. in the year 1996 to recognition of the fruits and the vegetable from images through color, and texture, The accuracy results nearly 95%. (Bolle, 1996). Some research has been done on image categorization. On it to develop to recognize the items.

The method of color coherence vector (CCV) and border interior classification (BIC) obtained poor results (Selvaraj, 2013). The 86% accuracy achieved using minimum-distance classifier over their dataset with different types of items and some framework presented to classify automatic recognition of product. Some research recognized some vegetables by utilizing color texture features and histogram, where accuracy of up to 96.55%.

E. Classification and Identification Food and Vegetable

Unconscious recognition of food unwanted point of diseases increase time to time therefore it is necessary to detect immediately the symptoms appear on the food. It can develop after arrangement of the food by quality. Then determine to measures those and avoid losses, the recognize is critical to the symptoms of fruit diseases.

I have followed open research to how systems are going to work; the images are sorted by color and texture, the images seem similar in shape and colors, so analysis and identify images. There are many approaches are used to differentiate items using machine learning.

F. Machine Learning

In machine learning a computer program is work like human experience for a target. The metric describes accuracy when categorizing a data sample, theseveral types of evaluation the error and gap between in it, in this research, we mainly focus on image information, classification and machine learning is sorted into some major classes under defined:

- Supervised Learning
- Unsupervised Learning.
- Reinforcement Learning.

1 . Reinforcement Learning

It explains class of challenges in which an agent works in a given environment and must learn to function based on input. The usage of an environment implies that there is no preset training dataset, but rather a goal or collection of objectives that an agent must attain, actions that they may take, and reply on performance towards goal.

2 . Deep Learning

Deep learning is a machine learning approach that trains computers to accomplish things that humans do instinctively. Artificial neural networks are an area of machine learning that is involved with algorithms motivated by the structure and function of the brain.

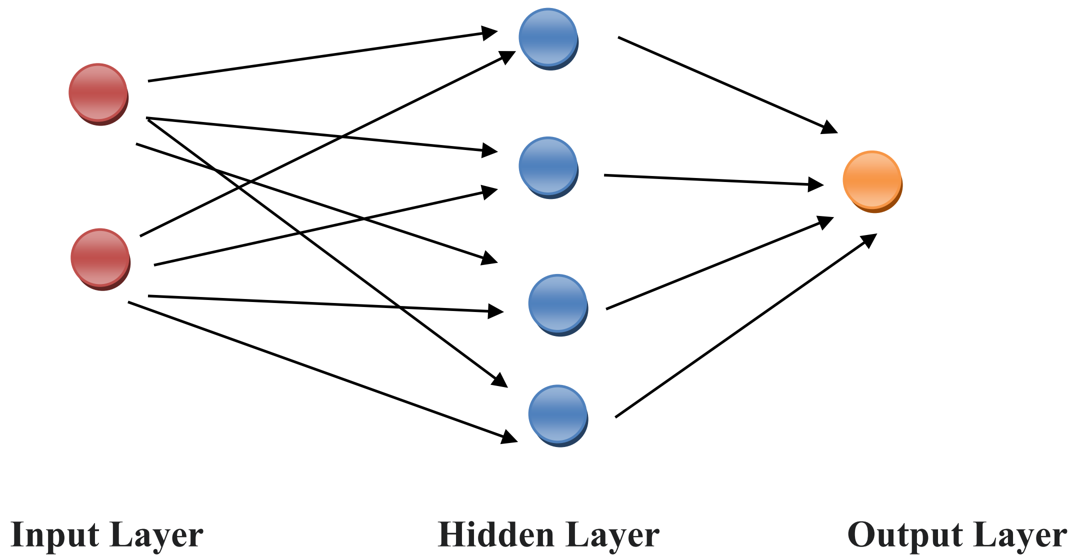


Figure 1 Deep learning and neural network (NN)

3 . Artificial Neural Networks (ANNs)

Artificial neural networks are presented as working system brains, with algorithms capable of learning from given events or samples.

Artificial Neural Networks are defined by (Singh and Chauhan) as a mathematical model that is established on organic neural networks and so is an simulation of biological neural system.

4 . Convolutional Neural Networks (CNNs)

Convolutional neural network (CNN) made up of convolutional layers combined plus sub sampling, the outputs of which are retrieved. CNN have diverse topologies, but the basic remains unchanged.

5 . VGGNet

VGGNet features standardised convolution and fully connected layers. VGGNet is a set of convolutional neural networks with well-designed kernels and layers with high accuracy.

VGG-19	VGG-16	VGG-16 (Conv1)	VGG-13	VGG-11 (LRN)	VGG-11
Image	Image	Image	Image	Image	Image
Conv3-64	Conv3-64	Conv3-64	Conv3-64	Conv3-64	Conv3-64
Conv3-64	Conv3-64	Conv3-64	Conv3-64	LRN	Max pool
Max pool	Max pool	Max pool	Max pool	Max pool	
Conv3-128	Conv3-128	Conv3-128	Conv3-128	Conv3-128	Conv3-128
Conv3-128	Conv3-128	Conv3-128	Conv3-128	Max pool	Max pool
Max pool	Max pool	Max pool	Max pool		
Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256
Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256
Conv3-256	Conv3-256	Conv1-256	Max pool	Max pool	Max pool
Conv3-256	Max pool	Max pool			
Max pool					
Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
Conv3-512	Conv3-512	Conv1-512	Max pool	Max pool	Max pool
Conv3-512	Max pool	Max pool			
Max pool					
Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
Conv3-512	Conv3-512	Conv1-512	Max pool	Max pool	Max pool
Conv3-512	Max pool	Max pool			
Max pool					
FC-4096	FC-4096	FC-4096	FC-4096	FC-4096	FC-4096
FC-4096	FC-4096	FC-4096	FC-4096	FC-4096	FC-4096
FC-1000	FC-1000	FC-1000	FC-1000	FC-1000	FC-1000
Soft-max	Soft-max	Soft-max	Soft-max	Soft-max	Soft-max

	VGG-19	VGG-16	VGG-16 (Conv1)	VGG-13	VGG-11 (LRN)	VGG-11
Number of Parameters (millions)	144	138	134	133	133	133
Top-5 Error Rate(%)	9.0	8.8	9.4	9.9	10.5	10.4

Figure 2 VGGNet Parameterson.

6 . ResNet

ResNet is the name of a CNN structure that permits information through cross-layer information. This approach addresses vanishing gradient difficulties, which occur as a network gets deep, causing propagated information to take a long time to converge and the derivatives of propagated errors to evaporate.

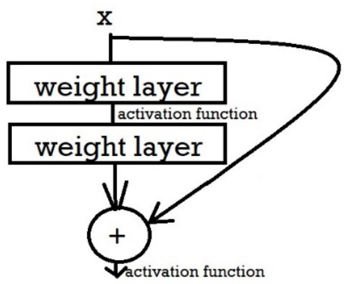


Figure 3 ResNet Layer module

Figure 3 shows a ResNet module, where input x is fed into two weight layers and x duplicate is summed with the two weight outputs, then go through another activation function.

ResNet comes in a variety of versions, the finest of which is ResNet152 (He K., Zhang, Ren, & Sun, 2015), which contains 152 layers. According to this study, increasing network depth can improve accuracy; in our experiments, we used the deepest network for grading fruit freshness.

layer name	conv1	conv2_x		conv3_x	conv4_x	conv5_x	
output size	112x112	56x56		28x28	14x14	7x7	1x1
ResNet18-layer	7x7, 64, stride2	3x3 max pool stride2	$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}$ x2	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}$ x2	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix}$ x2	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix}$ x2	average pool, 1000-d fc, softmax
ResNet34-layer	7x7, 64, stride2	3x3 max pool stride2	$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}$ x3	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}$ x4	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix}$ x6	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix}$ x3	average pool, 1000-d fc, softmax
ResNet50-layer	7x7, 64, stride2	3x3 max pool stride2	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}$ x3	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}$ x4	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}$ x6	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}$ x3	average pool, 1000-d fc, softmax
ResNet101-layer	7x7, 64, stride2	3x3 max pool stride2	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}$ x3	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}$ x4	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}$ x23	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}$ x3	average pool, 1000-d fc, softmax
ResNet152-layer	7x7, 64, stride2	3x3 max pool stride2	$\begin{bmatrix}1x1,64\\3x3,64\\1x1,256\end{bmatrix}$ x3	$\begin{bmatrix}1x1,128\\3x3,128\\1x1,512\end{bmatrix}$ x8	$\begin{bmatrix}1x1,256\\3x3,256\\1x1,1024\end{bmatrix}$ x36	$\begin{bmatrix}1x1,512\\3x3,512\\1x1,2048\end{bmatrix}$ x3	average pool, 1000-d fc, softmax

Figure 4 ResNet Structure of Family

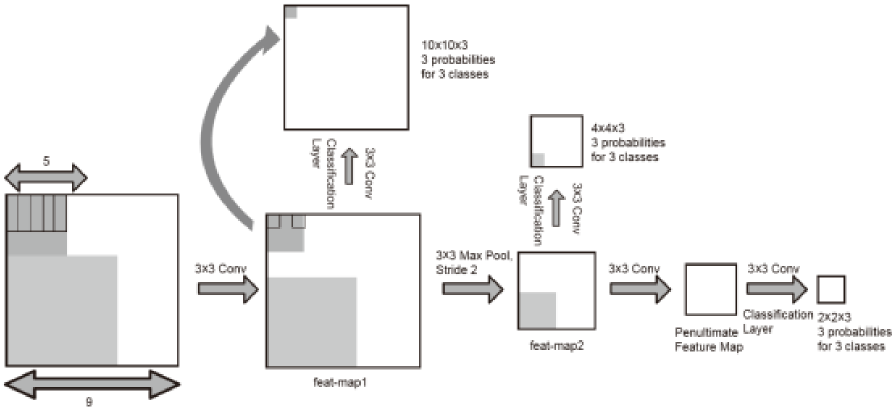


Figure 5 SSD Convolutional layers

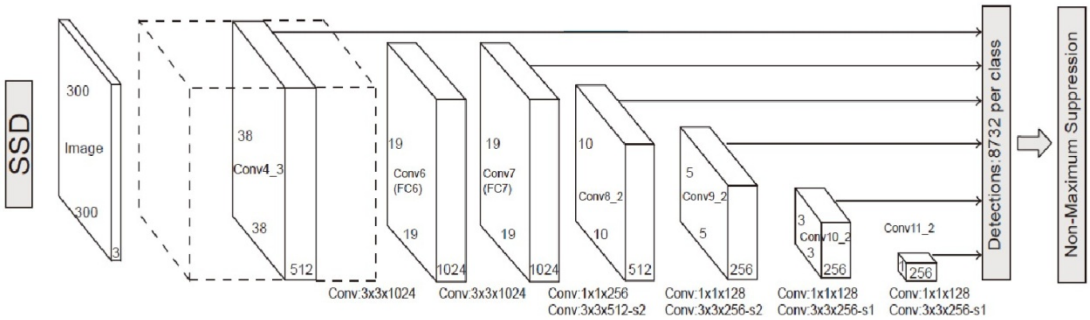


Figure 6 SSD structures

III. RESEARCH MODEL DEVELOPMENT AND HYPOTHESES FORMULATİON

Chapter three we provide description of our research and how data was collected before the execution of algorithms.

A. Limitation

The limitation of the deep learning, leftover vegetables and food are hard to analyze for every-one as everybody is not an expert in quality control, or they cant pay for highly advanced gear (kta Sonwani, 2022).

B. Data Collection

Our project data comes from a variety of sources, including the Food Fresh and Rotten dataset from Kaggle, the Vegetables Fresh and Rotten dataset from Kaggle, and self-collected photos. Manual data collection included screenshots from stock image websites and photographs captured on our smartphones for further assessment. These transformations were chosen because they maintain colour information, saturation, and brightness, allowing for type and state predictions. The augmentation produced roughly 32,000 acceptable photos, our model achieved excellent final accuracies, and we had the possibility of expanding the dataset if necessary. These photos were used to train, validate, and test the model.

The gathered dataset includes six varieties of fresh and rotting fruits and vegetables, such as apple, banana, cucumber, okra, potato, and tomato, with 12 classes shown in Figure 3.1, obtained from a range of locations with varying ambient sounds, irrelevant neighbouring items, and light conditions. In all, about 8280 training photos were gathered with each variety of fruit and vegetable, with the entire dataset provided in Table 3.1. The dataset was divided into training and validation sets at a 1:8 ratio (80 percent for training and 20 percent for validation). The freshness grade is tiered from fresh to rotten.

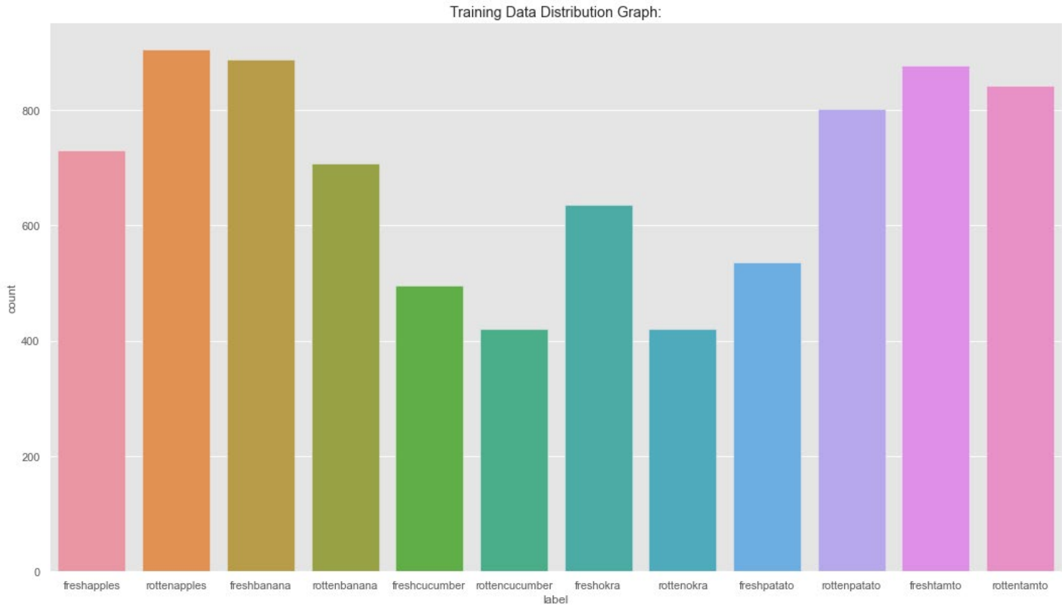


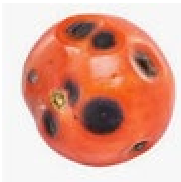
Figure 7 Training Data Distribution Graph

We have to classify fruit and vegetables items to collect datasets and train models accordingly. Classification will take part over the food class, shape, color and background. **GTM** is an open source and efficient teachable framework for tens or flow so I preferred to use that instead to do some deep learning on my own because that would have taken too much time.

The Kaggle dataset (BALOCH, n.d.) is thought one of the most valuable datasets for Vegetables and Fruits Fresh and Stale (BALOCH, n.d.) it includes more than 8180 in public available images captured by different angles, dimensions and multiple cameras. The division of dataset for training and testing is 80% and 20% respectively, and several googled and captured the images from cameras. Therefore, dataset contains different levels of image quality.

Table 1 Samples for each class in the dataset

Food Name		**Images**		
Fruit	Fresh Apple			
	Rotten Apple			
	Fresh Bananna			
	Rotten Bananna			
	Fresh Cucumber			
	Rotten Cucumber			

Vegetable	Fresh Potato			
	Rotten Potato			
	Fresh Tomato			
	Rotten Tomato			
	Fresh Okra			
	Rotten Okra			

C. Key Concepts and Definitions

The concept of the Image data collection is to measure the quality or freshness of the food with image processing. The collected images and then categorized it into from each family is interest take and put to formulated with the processing algorithm.

1 . Exporting Tensor flow

Tensor Flow is an end on open-source platform used for machine learning. Once we finish training the model then we have two options: either we can host our model live over the network or download a tensor flow file to integrate into any programming language.

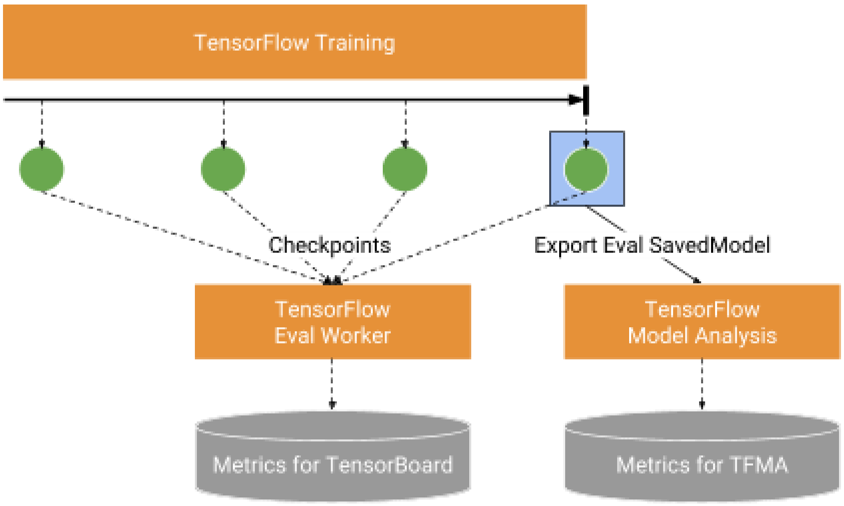


Figure 8 Training model

2 . Image processing

Food quality can be measured in multiple ways, but we are going to use image processing because it's cheap and it has 80% confidence. It is a simple and effective approach to measure the quality of food. In this approach images will be taken and processed by a trained machine learning model and the algorithms then the whole classification will be done.

Classification plays a major role because many foods, foods and vegetables have similar properties like color and shapes which does make confusion sometimes.

So, it's clear that more data and images of each food will bring more accuracy to classification and identification of food in order to find quality and nutrition.

Once we created tensor-flow model then we can develop short algorithm to interact with it. We got a tensor flow file and labes.txt file in which we can see the class name that we have created. It's like an array with indexes. The tensor flow will return us the array of results after processing the image which will identify the health of food according to the data we have provided.

More data we will use to create the dataset than more accuracy will come. Currently this program is not that accurate; the total accuracy rate of this program is 55% in some cases because it was too difficult to collect a huge amount of data for quality results. So, first we have to, take an image as input then tensor flow tells us about the health in percentage.

```
Image = file()
Results = tensorflow.analyze (image)
For (results as result)
      Print (result)
```

This will print the class name of food and percentage of the health according to our datasets. According to the name, image processing addresses to processing of image and this can include much different technologies until and unless we get our results. The final result may be either in the shape of image or a list of features of that image. This will be used for further decision making and analysis.

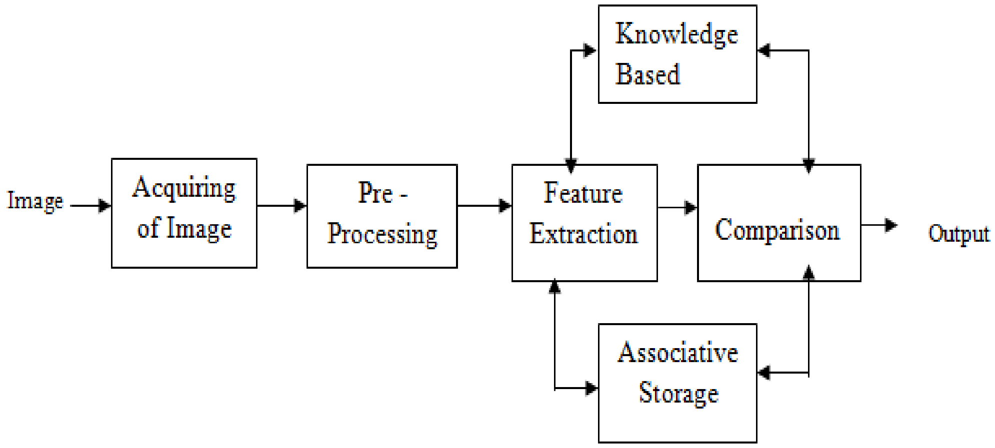


Figure 9 Image processing

3 . What you think about image?

2D function *F (x,y)* representation such as x and y are spatial coordinates, is known as an image. Amplitude of the F with a particular value of (x) and (y) is considered as quality of image on that point. For supposeif x and y, and value of amplitude is decided to a particular range then it is known as a digital image.

It consists of array of pixels organized in rows and columns. Pixels are called the elements within an image that consists of information about the colors and the intensity. Another way of representing an image is in 3D where x,y, and z are very special spatial coordinates. Pixels are organized in the shape of a matrix. This type of image is known as RGB image.

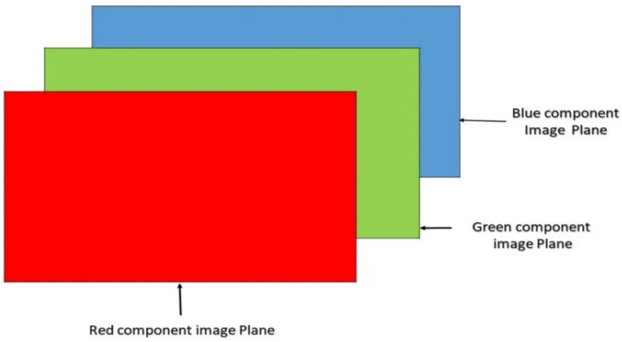


Figure 10 RBG image is a three-layered Image

Images can differentiate in various types

- RGB images: RGB images consists of three layers of 2D image, the layers are Green, Blue and Red.
- Grayscale images: gray scale images consistof black and white and has only single channel.

D. Convolutional Neural Networks and Transfer Learning

There are two categories of CNN layers: The primary layer and secondary layer, and secondary layers are optional for making the CNN. Figure 3.5 shows a detailed structure of CNN.

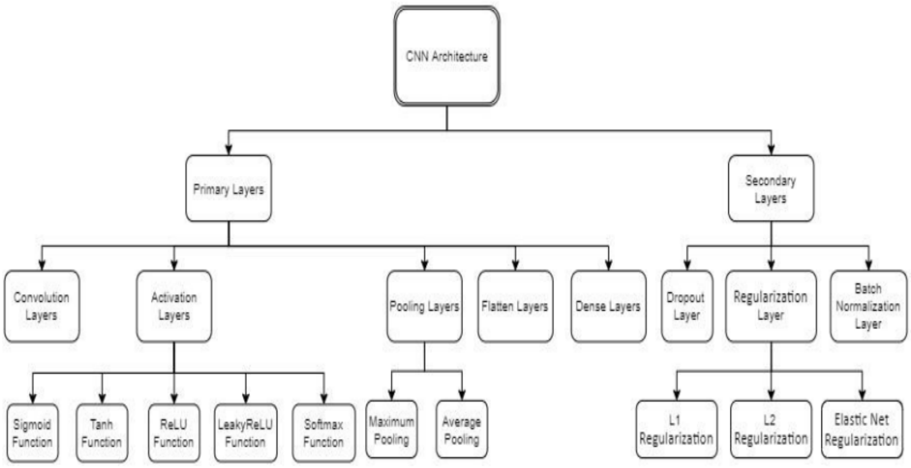


Figure 11 Convolutional neural networks Structure (CNNs)

1 . Transfer learning

Transfer learning is technique used in deep learning; it is used to train a CNN without initializing its weights from scratch. (Deng, Dong, Socher, Li, & Li, 2009). Only fine tuning of top layer is required as per recommendations of researchers (Colucci, Lia, Xiaoyang, & Fabrizio, 2020). A generic CNN model architecture can be seen in Figure 3.11.

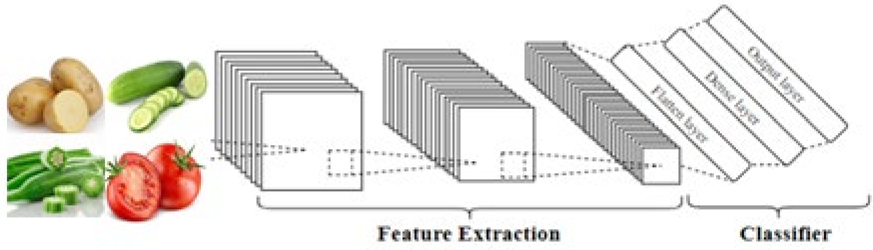


Figure 12 Convolutonal neural network architecture.

E. CNN Architectures

A convolution neural network is a special type of multilayer neural network designed to recognize visual patterns directly from pixel images with higher accuracy and minimal pre-processing. CNN architectures used in transfer learning are reviewed (Alom, et al., 2018), According to them, the rise of deep learning in image classification started in 2012 with the introduction of AlexNet, which also introduced the ReLU activation layer. Using a CNN in image classification increased its accuracy and eliminated the need to build each image with features. After AlexNet, many architectures were introduced namely VGG16, ResNet, and Xception with more features to classify images effectively (Dang, 2012).

The main problem is that network involves very large datasets selected to able all train its parameters. Figure 3.10. shows a review of the proposed networks with their number of parameters and their accuracy over the ImageNet dataset. The accuracy is calculated by dividing the properly classified studies over the total number of observations.

Achitecture	Number of Parameters	Top 5 Accuracy	Top 1 Accuracy
AlexNet	62,378,344	84.60%	63.30%
VGG16	138,357,544	91.90%	74.40%
GoogLeNet	23,000,000	92.2%	74.80%
ResNet-152	25,000,000	94.29%	78.57%
DenseNet	8,062,504	93.34%	76.39%
Xception	22,910,480	94.50%	79.00%

Figure 13 Comparison of multiple networks

IV. PROPOSED APPROACH

A. Research Design

The background of this desktop application and search is that we were searching for healthy food, so we found that in our daily life we are eating food without knowing about the quality, freshness, nutrients and minerals of food. So, we started our research about it.

We have done so much research about foods, different food and their nutrients and minerals. After that we made research that either common people are getting these nutrients or not. We have visited different farms, food courts and many places for our research and data sets. After search about the nutrients of foods we saw that there are so many people who have a lack of energy in their body as compared to their age and BMI. So, we make further research about it to find the cause.

In our research work we found that common people did not know what they are eating. The food which they are having is fresh or not, contains rich minerals not or not, getting enough energy through their food not. That's why they got sick rapidly. It is hard for a common person who is not an expert in finding food quality to judge their food. Many people cannot afford the devices which can examine their food quality. So, what should they do in this scenario?

To resolve this issue, we have decided to design such an application which can find food quality and can be affordable for common people who can’t afford expensive devices. We have designed this offline desktop application “Food inspection with Image Processing”. This application is founded in Artificial Intelligence and machine-learning. In this research we used TensorFlow which is an Artificial Intelligence plug-in along with image processing techniques.

Through this offline desktop application, the common people who can't afford expensive food quality detecting devices will become able to examine their food before having it and they will get fresh and quality food. We have designed the user-friendly user interface for this application so that anyone can interact with it easily. They will live a healthy life with their family. When one will live a healthy life then the disease rate will automatically decrease.

Furthermore, CNNs are generally trained by using transfer learning method, various studies shows the classification of food pictures applying deep-learning methods (BALOCH, n.d.) introduced a dataset called (Vegetables & Fruits fresh and Stale) dataset that contains 8180 furits and vegetables images where there are both type rotten and fresh image are available. The performance of the proposed model is acceptable compared to the predictions of the fresh and stale vegetables and fruits. (Scherer, Muller, & Behnke, 2010) researched the performance of four high-tech CNNs, such as Inception, ResNetV2, VGG16, AlexNet in the (BALOCH, n.d.) dataset.

B. Methodology

The impact of transfer learning on the accuracy of picture classification. To do this, we evaluate the classification accuracy of CNN architectures trained on ImageNet. Then, without using the ImageNet weights, we train the same datasets on the same networks. The best classifier on images of fruits and vegetables that produces the best results. We demonstrate the outcomes of training the analysed CNNs using the given dataset. the impact of transfer learning on the accuracy of picture categorization. To do this, we evaluate the classification accuracy of CNN architectures trained on ImageNet. Then, without using the ImageNet weights, we train the same datasets on the same networks. The most effective classifier for classifying fruit and vegetable images. we present findings achieved by training dataset (BALOCH, n.d.) on the studied CNNs.

1 . Evaluation Metrics

Accuracy and Kappa are the two metrics used to measure the performance of networks. Each metric is summarized as below:

The Kruskal–Wallis hypothesis test is a non-parametric testing technique (Kruskal & Wallis, 1952). No hypothesis is made from this analysis on the routine of data, and it compares the medians of various models. The null and alternative hypothesis tested are the following:

- H_0: $Thepopulationsmediansareallequal$
- H_1: $Thepopulationsmediansarenotallequal$

We first examined the ordinariness theory by using the ShapiroWilk. Considering that test makes not allow to reject the alternative hypothesis (for example our data are not normally distributed). In this manner, each method's constancy can be evaluated also modifying fortunate seeds' affect (Schmidt F, 2020).

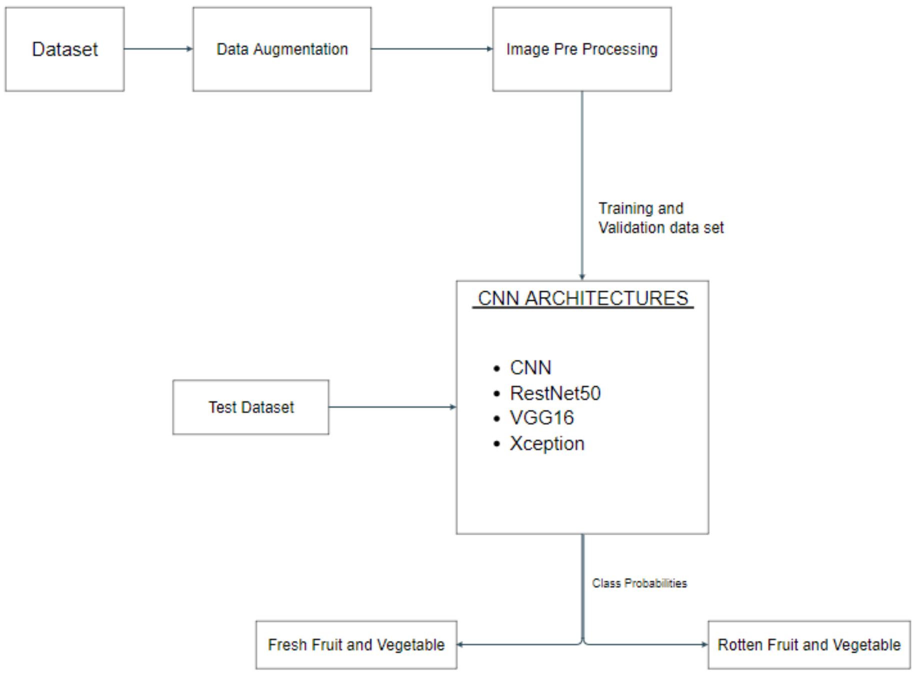


Figure 14 A schematic diagram

2 . Dataset

The dataset used in this thesis is the widely available Vegetables & Fruits fresh and Stale Dataset (BALOCH, n.d.). The dataset consists of 6 differenttypes of both fresh and stale (rotten) fruits and vegetablesApple, Banna, Cucumber, Okra, Potato and tamato for each classification has text and label, showing if the image shows a rotten and fresh. The dataset includes total number of images**12,220**. Where the maker of dataset split it hooked on training and test collections. The train set included **4162** images fresh fruits and vegtaibles (54.83% of the dataset) and **4018** images rotten fruits and vegtaibles (39.18% of the dataset), and the test set contained **2081** images fresh fruits and vegtaibles (4.16% of the dataset) and **1959** images with testing rotten fruits and vegtaibles (4.88% of the dataset). In general, the dataset used for training and testing the results is 52% and 48% respectively. Table 4.1 shows the summary of these datasets.

Table 2 Vegetables & Fruits fresh and StaleDataset Summary

No	Label	Number of Training Images	Number of Testing Images
1	Fresh Apple	731	396
2	Rotten Apple	906	387
3	Fresh Banana	887	511
4	Rotten Banana	708	370
5	Fresh Cucumber	496	279
6	Rotten Cucumber	421	255
7	Fresh Okra	635	370
8	Rotten Okra	338	224
9	Fresh Patato	536	270
10	Rotten Patato	802	370
11	Fresh Tamto	877	255
12	Rotten Tamto	843	353
	Total	**8180**	**4040**

*Total Training Dataset Images:8180 -Fresh Food: 4162-Rotten Food: 4018
*Total Testing Dataset Images:4040 - Fresh Food: 2081 -Rotten Food: 1959

C. Results

Through this research, all the metrics were set up. Every network was either fully finetuned or trained from scrap (Kingma & Ba, 2014) where Adam optimizer was used in all the researches. As stated by research paper (Kingma & Ba, 2014), the learning rate is set to be as low as 0.0001 in order to avoid dynamic variations to the original weights. All the images were resized to 224 × 224 px. The loss function used in this paper is categorical_crossentropy because the images are categorized categorically. An early-on stopping principle of 50 epochs wouldbe utilized to prevent algorithms if no updates occurred to the validation score. The batch size was chosen to be 16, and the training dataset is divided into 52% for training in addition to 48% for validating the findings in training. For increasing the dataset, four techniques for enlargement of images are used which results in making the networks more robust against overfitting. The horizontal, vertical flips and zooming augmentation techniques are used. Additionally, image augmentation is performed to balance the number of images in the two target classes, thus achieving 50% of images without fresh and 50% of images with fresh in the training set. After the training, each network's performance was tested using the dataset that was supplied by the owner and creator of the dataset. The test dataset was not used during the training phase but only in the final testing phase. The hyperparameters used are presented in Fig4.2. In the following parts, Kappa is the metric considered for comparing the performance of the different architectures.

Framework	Keras With Python
Optimizer	Adam
Learning Rate	0.0001
Loss Function	categorical_crossentropy
Early Stopping	50 epochs
Batch Size	16
Validation Split	20%
Image Augmentation	Horizontal flips
	Vertical flips
	Zooming

Figure 15 The hyperparameters were used for all the experiments

V. Experimental Result

Through this offline desktop application, the common people who can't afford expensive food quality detecting devices will become able to examine their food before having it and they will get fresh and quality food. We have designed the user-friendly user interface for this application so that anyone can interact with it easily. They will live a healthy life with their family. When one will live a healthy life then the disease rate will automatically decrease.

We have done so much research about foods, different food and their nutrients and minerals. After that we made research that either common people are getting these nutrients or not. We have visited different farms, food courts and many places for our research and data sets. After search about the nutrients of foods we saw that there are so many people who have a lack of energy in their body as compared to their age and BMI. So, we make further research about it to find the cause.

The 98% accuracy rate collected from images in our research so that we can neglect the conflict chances. We have entered a large dataset in our software database in sort to reach high accuracy. When user opens application, it will show this welcome window. User can pick the image of food from the gallery, or he can click a real time image of their food to analyze it.

A. NEED

The leftover food is wasted very much nowadays and is hard to measure for a common person because of hybirdness of food. Peoples are unable to find the quality of food. People needs to know what they have bought and what they are eating, how much nutrition, hygiene and minerals it contains. Also, people should be aware of the food contains essential amount of nutrients according to their age group and physical needs, identifying these elements is very hard nowadays without a proper hardware device and knowledge, most of the people can't afford to buy a device. In the results they are targeted by low and unhealthy food supply. The poor-quality food and nutritions can contribute to stress, tiredness and low capacity to work and over time, it is also risk of developing some illnesses and other health problems

B. The Results of a Proposed CNN model

In this section we will discuss the output came by running the deployment code of CNN**.**

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 222, 222, 32)      896

 max_pooling2d_3 (MaxPooling  (None, 111, 111, 32)     0
 2D)

 conv2d_4 (Conv2D)           (None, 111, 111, 64)      18496

 max_pooling2d_4 (MaxPooling  (None, 55, 55, 64)       0
 2D)

 conv2d_5 (Conv2D)           (None, 55, 55, 128)       73856

 max_pooling2d_5 (MaxPooling  (None, 27, 27, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 27, 27, 128)       0

 flatten_1 (Flatten)         (None, 93312)             0

 dense_2 (Dense)             (None, 512)               47776256

 dropout_3 (Dropout)         (None, 512)               0

 dense_3 (Dense)             (None, 10)                5130

=================================================================
Total params: 47,874,634
Trainable params: 47,874,634
Non-trainable params: 0
_________________________________________________________________
```

Figure 16 Training Parameters Output

C. Training Model Parameters

The above screenshot shows the result of number of parameters trained using this approach.

D. CNN Learning Model Epoch

```
Epoch 1/50
349/349 [==============================] - 314s 897ms/step - loss: 1.1023 - accuracy: 0.5895 - val_loss: 0.6391 - val_accuracy: 0.7303 - lr: 0.0010
Epoch 2/50
349/349 [==============================] - 328s 938ms/step - loss: 0.7020 - accuracy: 0.7195 - val_loss: 0.4621 - val_accuracy: 0.8063 - lr: 0.0010
Epoch 3/50
349/349 [==============================] - 329s 942ms/step - loss: 0.6114 - accuracy: 0.7512 - val_loss: 0.6150 - val_accuracy: 0.7343 - lr: 0.0010
Epoch 4/50
349/349 [==============================] - ETA: 0s - loss: 0.5715 - accuracy: 0.7568
Epoch 4: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
349/349 [==============================] - 318s 910ms/step - loss: 0.5715 - accuracy: 0.7568 - val_loss: 0.4231 - val_accuracy: 0.8046 - lr: 0.0010
Epoch 5/50
349/349 [==============================] - 307s 880ms/step - loss: 0.4282 - accuracy: 0.8141 - val_loss: 0.3475 - val_accuracy: 0.8588 - lr: 2.0000e-04
Epoch 6/50
349/349 [==============================] - 306s 878ms/step - loss: 0.3952 - accuracy: 0.8226 - val_loss: 0.3466 - val_accuracy: 0.8548 - lr: 2.0000e-04
Epoch 7/50
349/349 [==============================] - ETA: 0s - loss: 0.3791 - accuracy: 0.8323
Epoch 7: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
349/349 [==============================] - 309s 886ms/step - loss: 0.3791 - accuracy: 0.8323 - val_loss: 0.3276 - val_accuracy: 0.8559 - lr: 2.0000e-04
Epoch 8/50
349/349 [==============================] - 311s 889ms/step - loss: 0.3573 - accuracy: 0.8384 - val_loss: 0.2883 - val_accuracy: 0.8847 - lr: 4.0000e-05
Epoch 9/50
349/349 [==============================] - 306s 878ms/step - loss: 0.3453 - accuracy: 0.8505 - val_loss: 0.2986 - val_accuracy: 0.8767 - lr: 4.0000e-05
Epoch 10/50
349/349 [==============================] - ETA: 0s - loss: 0.3446 - accuracy: 0.8551
Epoch 10: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.
349/349 [==============================] - 312s 895ms/step - loss: 0.3446 - accuracy: 0.8551 - val_loss: 0.2943 - val_accuracy: 0.8784 - lr: 4.0000e-05
Epoch 11/50
349/349 [==============================] - 310s 888ms/step - loss: 0.3306 - accuracy: 0.8612 - val_loss: 0.2902 - val_accuracy: 0.8790 - lr: 8.0000e-06
Epoch 12/50
349/349 [==============================] - ETA: 0s - loss: 0.3296 - accuracy: 0.8618
Epoch 12: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-06.
349/349 [==============================] - 340s 974ms/step - loss: 0.3296 - accuracy: 0.8618 - val_loss: 0.2896 - val_accuracy: 0.8790 - lr: 8.0000e-06
Epoch 13/50
349/349 [==============================] - ETA: 0s - loss: 0.3363 - accuracy: 0.8543Restoring model weights from the end of the best epoch: 8.
349/349 [==============================] - 324s 927ms/step - loss: 0.3363 - accuracy: 0.8543 - val_loss: 0.2893 - val_accuracy: 0.8801 - lr: 1.6000e-06
Epoch 13: early stopping
```

Figure 17 CNN Epoch Output

The above screen shows the results of model learning epoch after training to check accuracy values and loss values of dataset of images.

1 . Accuracy Graph

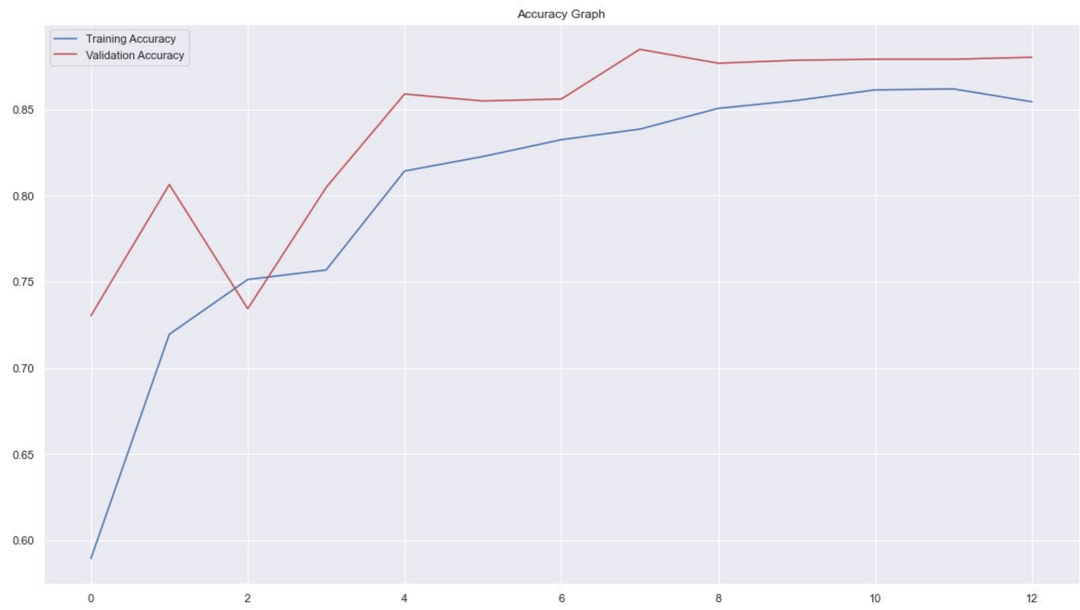


Figure 18 Model Trained Accuracy Graph

The above screenshot shows the model training accuracy and validation accuracy after learning image dataset

2 . Loss Graph

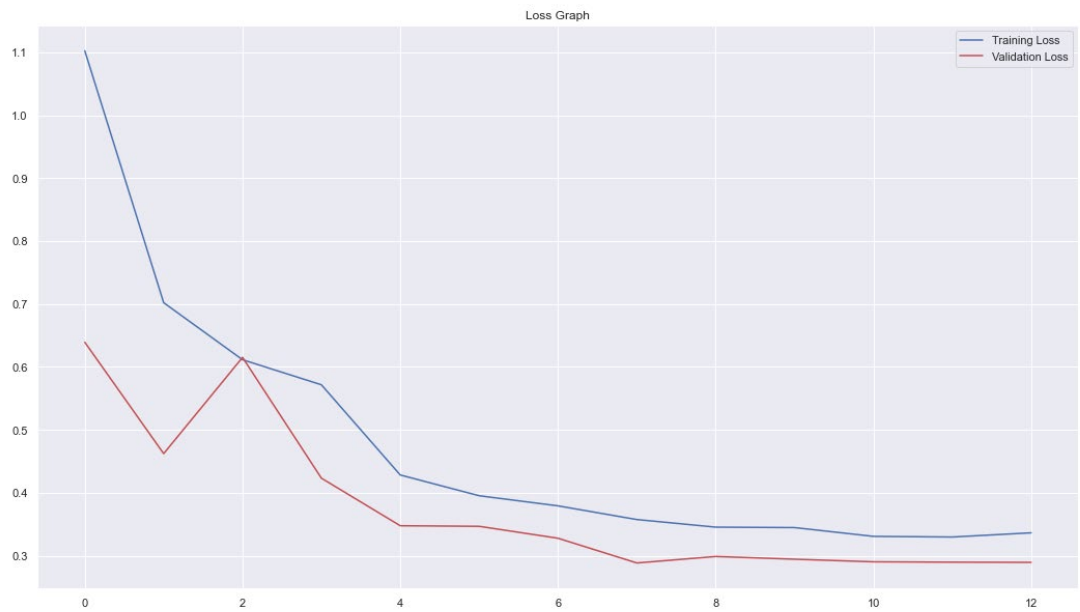


Figure 19 Model Trained Loss Graph

The above screenshot shows the model training loss and validation after learning image dataset.

3 . Confusion Matrix

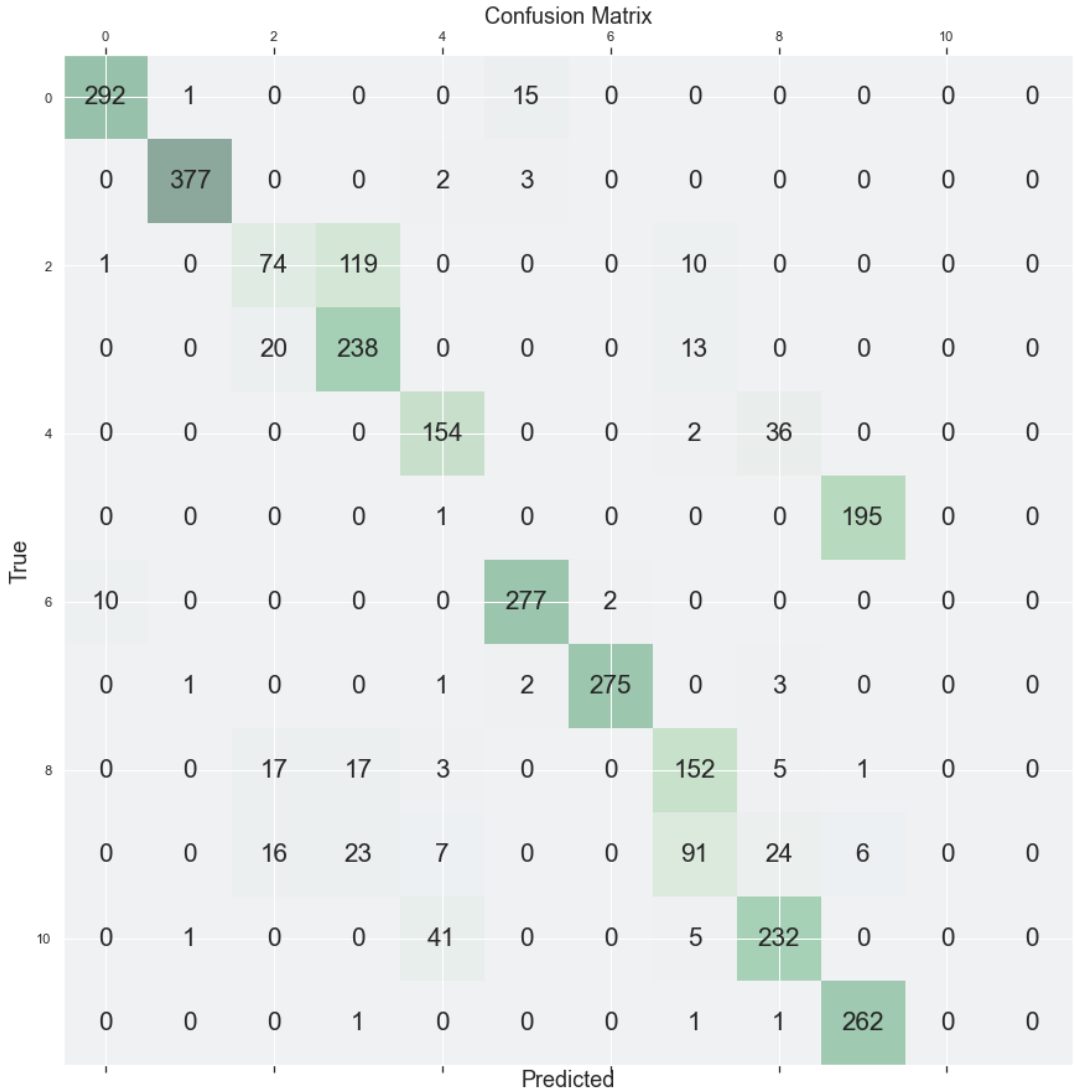


Figure 20 Confusion Matrix Predicted

The Confusion Matrix is used to define performance of algorithm by visualizing the values in a table where begin tissue is called healthy and malignant is considered cancerous.

4 . The Results of a Proposed ResNet50 model

This section will discuss output came by running the deployment code of ResNet50.Training Model Parameters

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	['input_1[0][0]']
conv1_conv (Conv2D)	(None, 112, 112, 64)	9472	['conv1_pad[0][0]']
conv1_bn (BatchNormalization)	(None, 112, 112, 64)	256	['conv1_conv[0][0]']
conv1_relu (Activation)	(None, 112, 112, 64)	0	['conv1_bn[0][0]']
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64)	0	['conv1_relu[0][0]']
pool1_pool (MaxPooling2D)	(None, 56, 56, 64)	0	['pool1_pad[0][0]']
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64)	4160	['pool1_pool[0][0]']
conv2_block1_1_bn (BatchNormalization)	(None, 56, 56, 64)	256	['conv2_block1_1_conv[0][0]']
conv2_block1_1_relu (Activation)	(None, 56, 56, 64)	0	['conv2_block1_1_bn[0][0]']
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64)	36928	['conv2_block1_1_relu[0][0]']
conv2_block1_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	['conv2_block1_2_conv[0][0]']
conv2_block1_2_relu (Activation)	(None, 56, 56, 64)	0	['conv2_block1_2_bn[0][0]']
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256)	16640	['pool1_pool[0][0]']
conv2_block1_3_conv (Conv2D)	(None, 56, 56, 256)	16640	['conv2_block1_2_relu[0][0]']
conv2_block1_0_bn (BatchNormalization)	(None, 56, 56, 256)	1024	['conv2_block1_0_conv[0][0]']
conv2_block1_3_bn (BatchNormalization)	(None, 56, 56, 256)	1024	['conv2_block1_3_conv[0][0]']
conv2_block1_add (Add)	(None, 56, 56, 256)	0	['conv2_block1_0_bn[0][0]', 'conv2_block1_3_bn[0][0]']
conv2_block1_out (Activation)	(None, 56, 56, 256)	0	['conv2_block1_add[0][0]']
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 64)	16448	['conv2_block1_out[0][0]']
conv2_block2_1_bn (BatchNormalization)	(None, 56, 56, 64)	256	['conv2_block2_1_conv[0][0]']
conv2_block2_1_relu (Activation)	(None, 56, 56, 64)	0	['conv2_block2_1_bn[0][0]']
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 64)	36928	['conv2_block2_1_relu[0][0]']
conv2_block2_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	['conv2_block2_2_conv[0][0]']

```
 conv5_block1_out (Activation)  (None, 7, 7, 2048)   0           ['conv5_block1_add[0][0]']

 conv5_block2_1_conv (Conv2D)   (None, 7, 7, 512)    1049088     ['conv5_block1_out[0][0]']

 conv5_block2_1_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block2_1_conv[0][0]']
 ization)

 conv5_block2_1_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block2_1_bn[0][0]']
 n)

 conv5_block2_2_conv (Conv2D)   (None, 7, 7, 512)    2359808     ['conv5_block2_1_relu[0][0]']

 conv5_block2_2_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block2_2_conv[0][0]']
 ization)

 conv5_block2_2_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block2_2_bn[0][0]']
 n)

 conv5_block2_3_conv (Conv2D)   (None, 7, 7, 2048)   1050624     ['conv5_block2_2_relu[0][0]']

 conv5_block2_3_bn (BatchNormal  (None, 7, 7, 2048)  8192        ['conv5_block2_3_conv[0][0]']
 ization)

 conv5_block2_add (Add)         (None, 7, 7, 2048)   0           ['conv5_block1_out[0][0]',
                                                                  'conv5_block2_3_bn[0][0]']

 conv5_block2_out (Activation)  (None, 7, 7, 2048)   0           ['conv5_block2_add[0][0]']

 conv5_block3_1_conv (Conv2D)   (None, 7, 7, 512)    1049088     ['conv5_block2_out[0][0]']

 conv5_block3_1_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block3_1_conv[0][0]']
 ization)

 conv5_block3_1_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block3_1_bn[0][0]']
 n)

 conv5_block3_2_conv (Conv2D)   (None, 7, 7, 512)    2359808     ['conv5_block3_1_relu[0][0]']

 conv5_block3_2_bn (BatchNormal  (None, 7, 7, 512)   2048        ['conv5_block3_2_conv[0][0]']
 ization)

 conv5_block3_2_relu (Activatio  (None, 7, 7, 512)   0           ['conv5_block3_2_bn[0][0]']
 n)

 conv5_block3_3_conv (Conv2D)   (None, 7, 7, 2048)   1050624     ['conv5_block3_2_relu[0][0]']

 conv5_block3_3_bn (BatchNormal  (None, 7, 7, 2048)  8192        ['conv5_block3_3_conv[0][0]']
 ization)

 conv5_block3_add (Add)         (None, 7, 7, 2048)   0           ['conv5_block2_out[0][0]',
                                                                  'conv5_block3_3_bn[0][0]']

 conv5_block3_out (Activation)  (None, 7, 7, 2048)   0           ['conv5_block3_add[0][0]']

 flatten_1 (Flatten)            (None, 100352)       0           ['conv5_block3_out[0][0]']

 dense_1 (Dense)                (None, 10)           1003530     ['flatten_1[0][0]']

==================================================================================================
Total params: 24,591,242
Trainable params: 1,003,530
Non-trainable params: 23,587,712
__________________________________________________________________________________________________
```

Figure 21 Training Data Summary

E. ResNetLearning Model Epoch

```
Epoch 1/50
349/349 [==============================] - 447s 1s/step - loss: 3.8184 - accuracy: 0.3130 - val_loss: 2.2001 - val_accuracy: 0.3948 - lr: 0.0010
Epoch 2/50
349/349 [==============================] - 474s 1s/step - loss: 2.4949 - accuracy: 0.4294 - val_loss: 1.8488 - val_accuracy: 0.4899 - lr: 0.0010
Epoch 3/50
349/349 [==============================] - 473s 1s/step - loss: 2.7191 - accuracy: 0.4602 - val_loss: 4.4447 - val_accuracy: 0.4432 - lr: 0.0010
Epoch 4/50
349/349 [==============================] - ETA: 0s - loss: 2.3841 - accuracy: 0.4979
Epoch 4: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
349/349 [==============================] - 447s 1s/step - loss: 2.3841 - accuracy: 0.4979 - val_loss: 2.8971 - val_accuracy: 0.4692 - lr: 0.0010
Epoch 5/50
349/349 [==============================] - 454s 1s/step - loss: 1.2028 - accuracy: 0.6199 - val_loss: 1.6873 - val_accuracy: 0.5095 - lr: 2.0000e-04
Epoch 6/50
349/349 [==============================] - 473s 1s/step - loss: 1.0724 - accuracy: 0.6285 - val_loss: 1.6565 - val_accuracy: 0.5164 - lr: 2.0000e-04
Epoch 7/50
349/349 [==============================] - 482s 1s/step - loss: 1.0750 - accuracy: 0.6254 - val_loss: 1.2190 - val_accuracy: 0.6017 - lr: 2.0000e-04
Epoch 8/50
349/349 [==============================] - 464s 1s/step - loss: 1.0396 - accuracy: 0.6319 - val_loss: 1.1021 - val_accuracy: 0.6069 - lr: 2.0000e-04
Epoch 9/50
349/349 [==============================] - 437s 1s/step - loss: 1.0681 - accuracy: 0.6293 - val_loss: 1.2997 - val_accuracy: 0.5706 - lr: 2.0000e-04
Epoch 10/50
349/349 [==============================] - 435s 1s/step - loss: 1.0304 - accuracy: 0.6386 - val_loss: 0.8935 - val_accuracy: 0.6692 - lr: 2.0000e-04
Epoch 11/50
349/349 [==============================] - 445s 1s/step - loss: 1.0028 - accuracy: 0.6447 - val_loss: 1.1653 - val_accuracy: 0.5908 - lr: 2.0000e-04
Epoch 12/50
349/349 [==============================] - ETA: 0s - loss: 1.0782 - accuracy: 0.6314
Epoch 12: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
349/349 [==============================] - 448s 1s/step - loss: 1.0782 - accuracy: 0.6314 - val_loss: 1.6013 - val_accuracy: 0.5326 - lr: 2.0000e-04
Epoch 13/50
349/349 [==============================] - 448s 1s/step - loss: 0.8013 - accuracy: 0.6956 - val_loss: 0.9182 - val_accuracy: 0.6646 - lr: 4.0000e-05
Epoch 14/50
349/349 [==============================] - ETA: 0s - loss: 0.7811 - accuracy: 0.7023
Epoch 14: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.
349/349 [==============================] - 444s 1s/step - loss: 0.7811 - accuracy: 0.7023 - val_loss: 0.9445 - val_accuracy: 0.6507 - lr: 4.0000e-05
Epoch 15/50
349/349 [==============================] - ETA: 0s - loss: 0.7430 - accuracy: 0.7182Restoring model weights from the end of the best epoch: 10.
349/349 [==============================] - 459s 1s/step - loss: 0.7430 - accuracy: 0.7182 - val_loss: 0.8942 - val_accuracy: 0.6749 - lr: 8.0000e-06
Epoch 15: early stopping
```

Figure 22 ResNet Epoch Output

1 . Training Data Distribution Graph

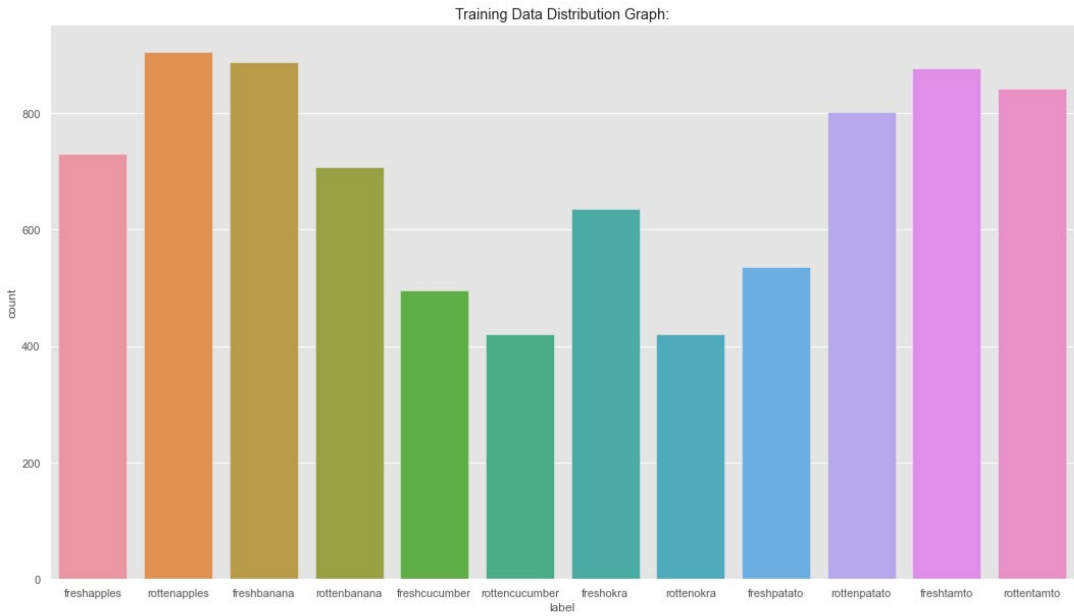


Figure 23 Training Data Distribution Graph

The Histogram screenshot shows the number of images trained of 12 classes including Fresh Apple, Rotten Apple, Fresh Banana, Rotten Banana, Fresh Cucumber, Rotten Cucumber, Fresh Potato, Rotten Potato, Fresh Tomato, Rotten Tomato, Fresh Okra, Rotten Okra.

2 . Accuracy Graph

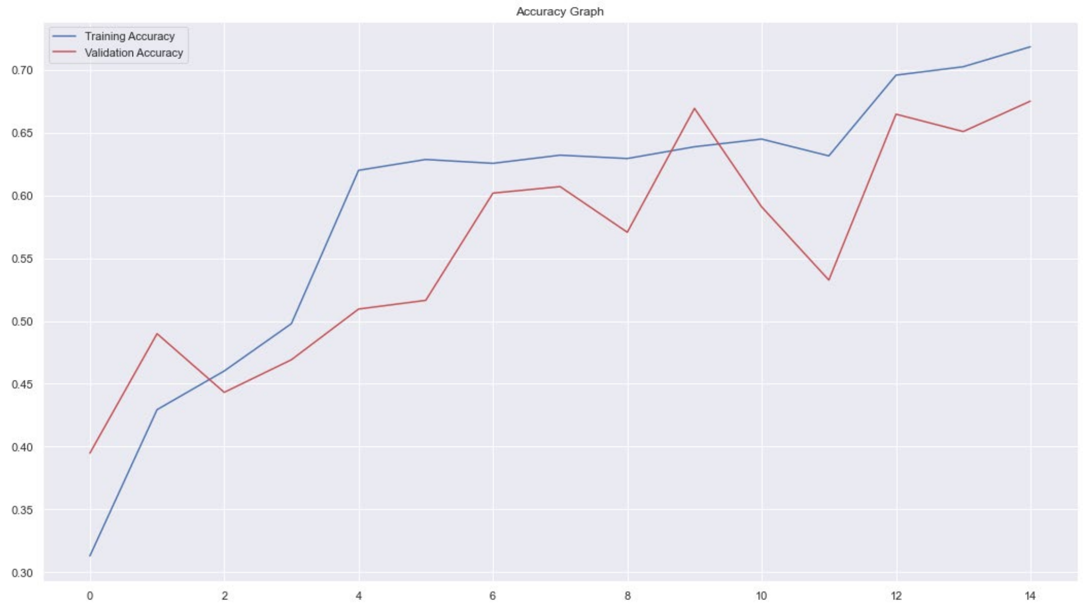


Figure 24 Model Trained Accuracy Graph

The above screenshot shows the model training accuracy and validation accuracy after learning image dataset

3 . Loss Graph

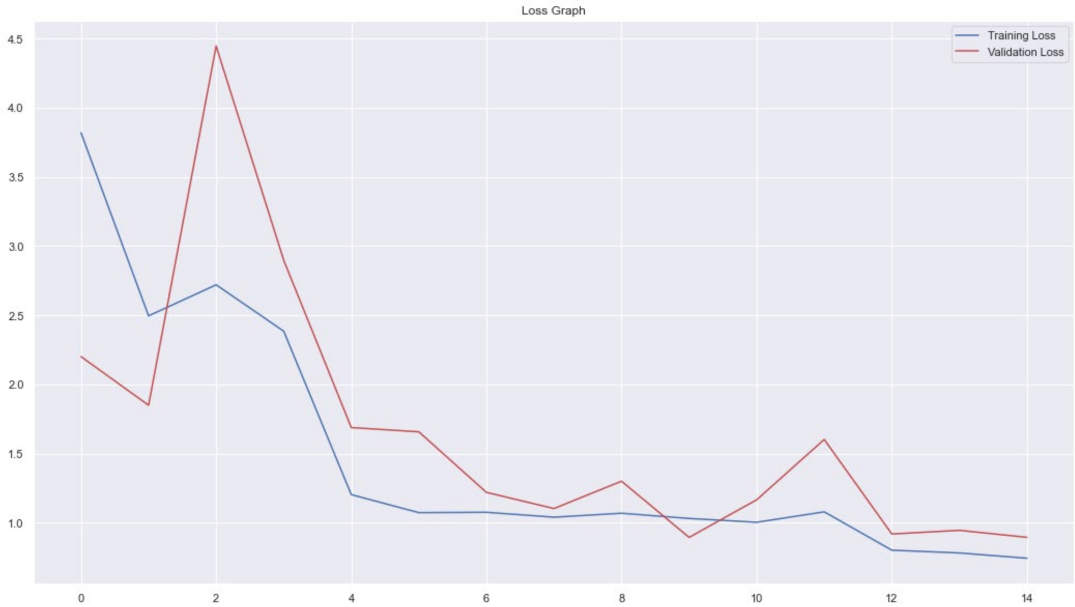


Figure 25 Model Trained Loss Graph

The above screenshot shows the model training loss and validation after learning image dataset

Confusion Matrix

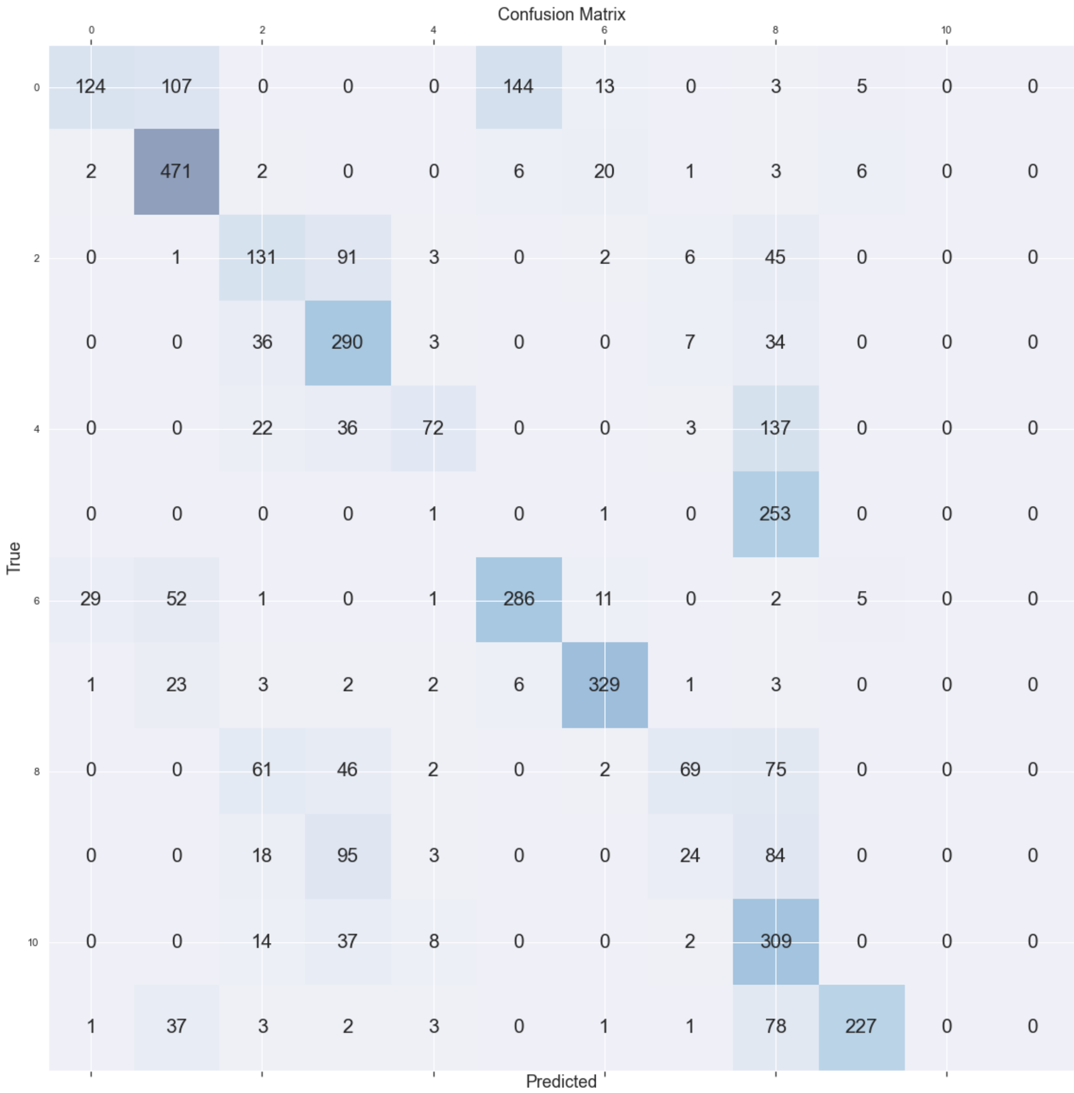


Figure 26 Confusion Matrix Predicted

The Confusion Matrix is used to define performance of algorithm by visualizing the values in a table where begin tissue is called healthy and malignant is considered cancerous.

F. The Results of a Proposed VGG16 model

This section will discuss output came by running the deployment code of ResNet50**.**

1 . Training Model Parameters

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0         
                                                                 
 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792      
                                                                 
 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928     
                                                                 
 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0         
                                                                 
 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856     
                                                                 
 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584    
                                                                 
 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0         
                                                                 
 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168    
                                                                 
 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080    
                                                                 
 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080    
                                                                 
 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0         
                                                                 
 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160   
                                                                 
 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808   
                                                                 
 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808   
                                                                 
 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0         
                                                                 
 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808   
                                                                 
 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808   
                                                                 
 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808   
                                                                 
 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0         
                                                                 
 flatten (Flatten)           (None, 25088)             0         
                                                                 
 dense (Dense)               (None, 10)                250890    
                                                                 
=================================================================
Total params: 14,965,578
Trainable params: 250,890
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 27 Training Data Summary

2 . VGGLearning Model Epoch

```
Epoch 1/50
349/349 [==============================] - 926s 3s/step - loss: 0.7254 - accuracy: 0.7646 - val_loss: 0.6970 - val_accuracy: 0.8202 - lr: 0.0010
Epoch 2/50
349/349 [==============================] - 947s 3s/step - loss: 0.4974 - accuracy: 0.8537 - val_loss: 0.2574 - val_accuracy: 0.9251 - lr: 0.0010
Epoch 3/50
349/349 [==============================] - 950s 3s/step - loss: 0.3657 - accuracy: 0.8956 - val_loss: 0.3114 - val_accuracy: 0.9170 - lr: 0.0010
Epoch 4/50
349/349 [==============================] - 955s 3s/step - loss: 0.4041 - accuracy: 0.8931 - val_loss: 0.2677 - val_accuracy: 0.9280 - lr: 0.0010
Epoch 5/50
349/349 [==============================] - 955s 3s/step - loss: 0.3608 - accuracy: 0.9073 - val_loss: 0.2701 - val_accuracy: 0.9280 - lr: 0.0010
Epoch 6/50
349/349 [==============================] - ETA: 0s - loss: 0.4004 - accuracy: 0.8969
Epoch 6: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
349/349 [==============================] - 951s 3s/step - loss: 0.4004 - accuracy: 0.8969 - val_loss: 0.4749 - val_accuracy: 0.8951 - lr: 0.0010
Epoch 7/50
349/349 [==============================] - 945s 3s/step - loss: 0.2000 - accuracy: 0.9413 - val_loss: 0.1709 - val_accuracy: 0.9625 - lr: 2.0000e-04
Epoch 8/50
349/349 [==============================] - 945s 3s/step - loss: 0.1591 - accuracy: 0.9511 - val_loss: 0.1322 - val_accuracy: 0.9643 - lr: 2.0000e-04
Epoch 9/50
349/349 [==============================] - 946s 3s/step - loss: 0.1579 - accuracy: 0.9501 - val_loss: 0.1272 - val_accuracy: 0.9671 - lr: 2.0000e-04
Epoch 10/50
349/349 [==============================] - 954s 3s/step - loss: 0.1399 - accuracy: 0.9551 - val_loss: 0.1457 - val_accuracy: 0.9602 - lr: 2.0000e-04
Epoch 11/50
349/349 [==============================] - ETA: 0s - loss: 0.1175 - accuracy: 0.9591
Epoch 11: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
349/349 [==============================] - 950s 3s/step - loss: 0.1175 - accuracy: 0.9591 - val_loss: 0.1821 - val_accuracy: 0.9568 - lr: 2.0000e-04
Epoch 12/50
349/349 [==============================] - 948s 3s/step - loss: 0.0994 - accuracy: 0.9667 - val_loss: 0.1261 - val_accuracy: 0.9654 - lr: 4.0000e-05
Epoch 13/50
349/349 [==============================] - ETA: 0s - loss: 0.1042 - accuracy: 0.9627
Epoch 13: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.
349/349 [==============================] - 950s 3s/step - loss: 0.1042 - accuracy: 0.9627 - val_loss: 0.1300 - val_accuracy: 0.9671 - lr: 4.0000e-05
Epoch 14/50
349/349 [==============================] - 942s 3s/step - loss: 0.0945 - accuracy: 0.9635 - val_loss: 0.1208 - val_accuracy: 0.9683 - lr: 8.0000e-06
Epoch 15/50
349/349 [==============================] - 942s 3s/step - loss: 0.1035 - accuracy: 0.9665 - val_loss: 0.1193 - val_accuracy: 0.9695 - lr: 8.0000e-06
Epoch 16/50
349/349 [==============================] - 940s 3s/step - loss: 0.0931 - accuracy: 0.9662 - val_loss: 0.1160 - val_accuracy: 0.9706 - lr: 8.0000e-06
Epoch 17/50
349/349 [==============================] - 944s 3s/step - loss: 0.0971 - accuracy: 0.9671 - val_loss: 0.1194 - val_accuracy: 0.9689 - lr: 8.0000e-06
Epoch 18/50
349/349 [==============================] - ETA: 0s - loss: 0.0983 - accuracy: 0.9685
Epoch 18: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-06.
349/349 [==============================] - 938s 3s/step - loss: 0.0983 - accuracy: 0.9685 - val_loss: 0.1166 - val_accuracy: 0.9689 - lr: 8.0000e-06
Epoch 19/50
349/349 [==============================] - 935s 3s/step - loss: 0.1007 - accuracy: 0.9671 - val_loss: 0.1172 - val_accuracy: 0.9683 - lr: 1.6000e-06
Epoch 20/50
349/349 [==============================] - ETA: 0s - loss: 0.0991 - accuracy: 0.9644
Epoch 20: ReduceLROnPlateau reducing learning rate to 3.200000264769187e-07.
349/349 [==============================] - 936s 3s/step - loss: 0.0991 - accuracy: 0.9644 - val_loss: 0.1184 - val_accuracy: 0.9689 - lr: 1.6000e-06
Epoch 21/50
349/349 [==============================] - ETA: 0s - loss: 0.0965 - accuracy: 0.9662Restoring model weights from the end of the best epoch: 16.
349/349 [==============================] - 940s 3s/step - loss: 0.0965 - accuracy: 0.9662 - val_loss: 0.1183 - val_accuracy: 0.9689 - lr: 3.2000e-07
Epoch 21: early stopping
```

Figure 28 ResNet Epoch Output

3 . Accuracy Graph

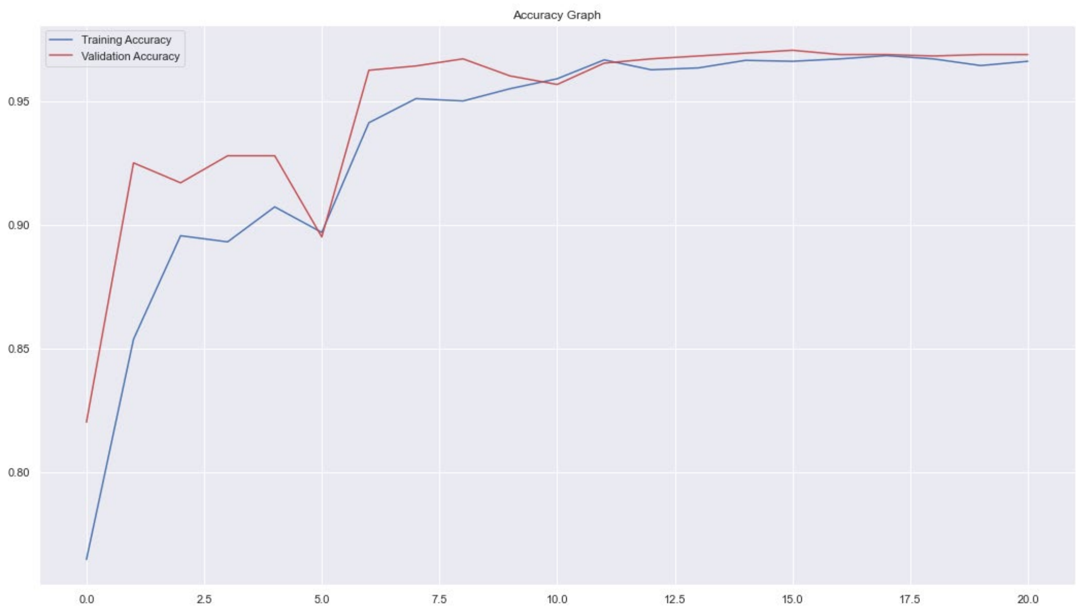


Figure 29 Model Trained Accuracy Graph

The above screenshot shows the model training accuracy and validation accuracy after learning image dataset.

4 . Loss Graph

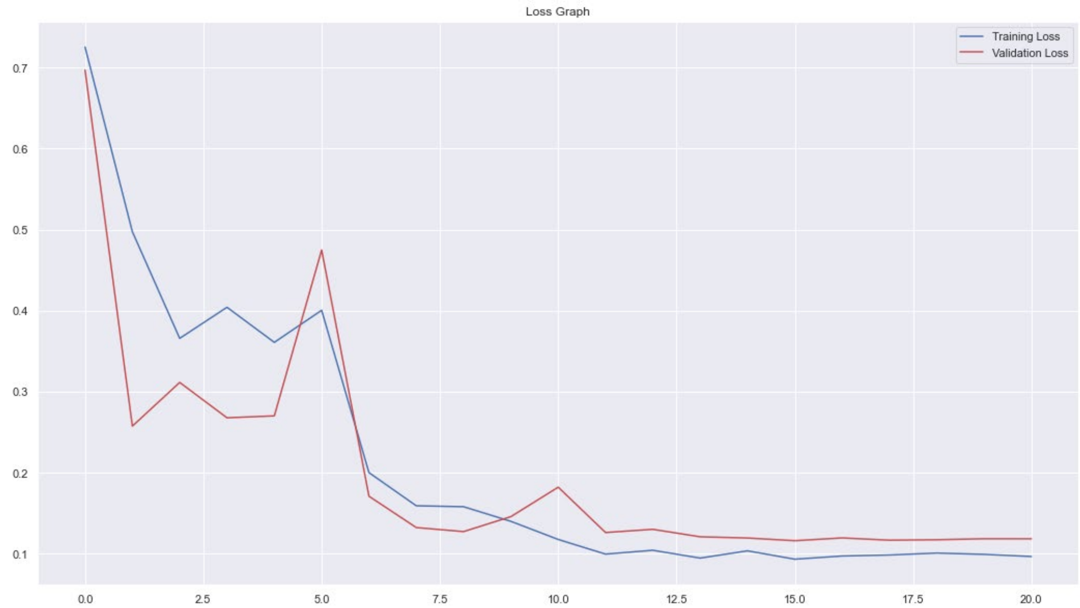


Figure 30 Model Trained Loss Graph

The above screenshot shows the model training loss and validation after learning image dataset.

5 . Confusion Matrix

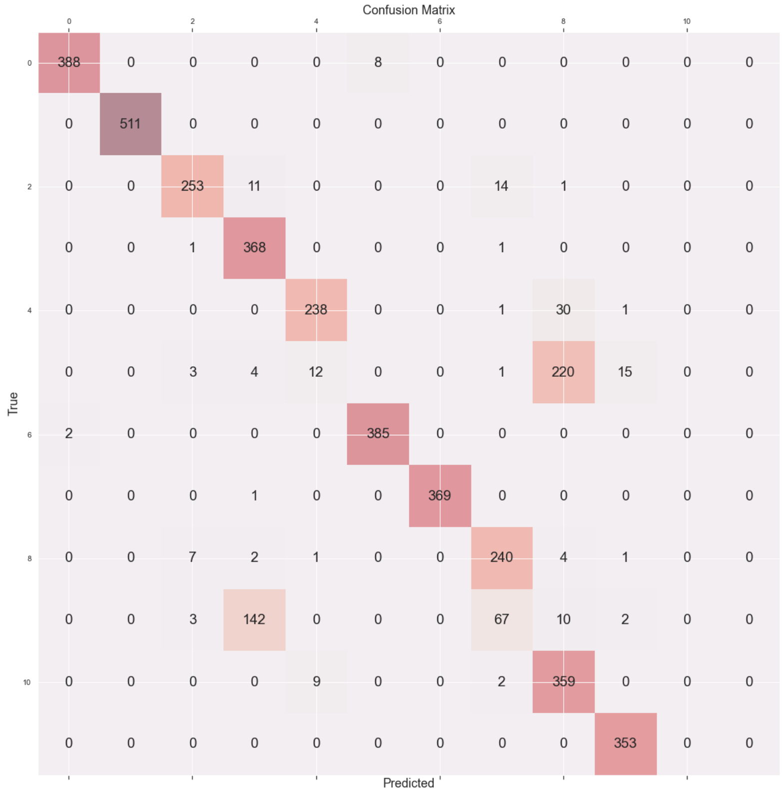


Figure 31 Confusion Matrix Predicted

The Confusion Matrix is used to define performance of algorithm by visualizing the values in a table where begin tissue is called healthy and malignant is considered cancerous.

G. The Results of a Proposed Xception model

This section will discuss output came by running the deployment code of Xception**.**

1 . Training Model Parameters

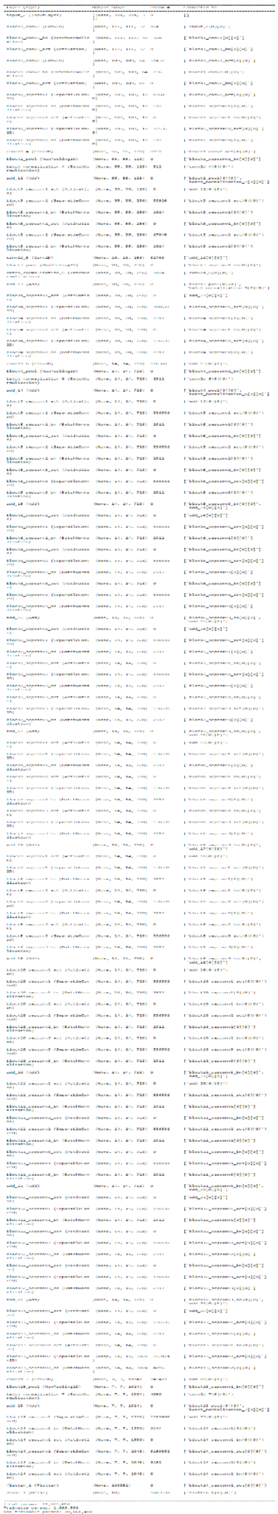


Figure 32 Training Data Summary

2 . XceptiobLearning Model Epoch

```
Epoch 1/50
349/349 [==============================] - 464s 1s/step - loss: 2.2427 - accuracy: 0.8174 - val_loss: 1.3851 - val_accuracy: 0.9066 - lr: 0.0010
Epoch 2/50
349/349 [==============================] - 483s 1s/step - loss: 1.6303 - accuracy: 0.8989 - val_loss: 1.2944 - val_accuracy: 0.9285 - lr: 0.0010
Epoch 3/50
349/349 [==============================] - 444s 1s/step - loss: 1.6414 - accuracy: 0.9119 - val_loss: 1.8764 - val_accuracy: 0.9199 - lr: 0.0010
Epoch 4/50
349/349 [==============================] - 442s 1s/step - loss: 1.3481 - accuracy: 0.9289 - val_loss: 1.4074 - val_accuracy: 0.9527 - lr: 0.0010
Epoch 5/50
349/349 [==============================] - 443s 1s/step - loss: 1.4398 - accuracy: 0.9275 - val_loss: 2.4906 - val_accuracy: 0.9112 - lr: 0.0010
Epoch 6/50
349/349 [==============================] - ETA: 0s - loss: 1.0558 - accuracy: 0.9426
Epoch 6: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
349/349 [==============================] - 443s 1s/step - loss: 1.0558 - accuracy: 0.9426 - val_loss: 1.6947 - val_accuracy: 0.9395 - lr: 0.0010
Epoch 7/50
349/349 [==============================] - ETA: 0s - loss: 0.7256 - accuracy: 0.9602Restoring model weights from the end of the best epoch: 2.
349/349 [==============================] - 448s 1s/step - loss: 0.7256 - accuracy: 0.9602 - val_loss: 1.2991 - val_accuracy: 0.9493 - lr: 2.0000e-04
Epoch 7: early stopping
```

Figure 33 ResNet Epoch Output

3 . Accuracy Graph

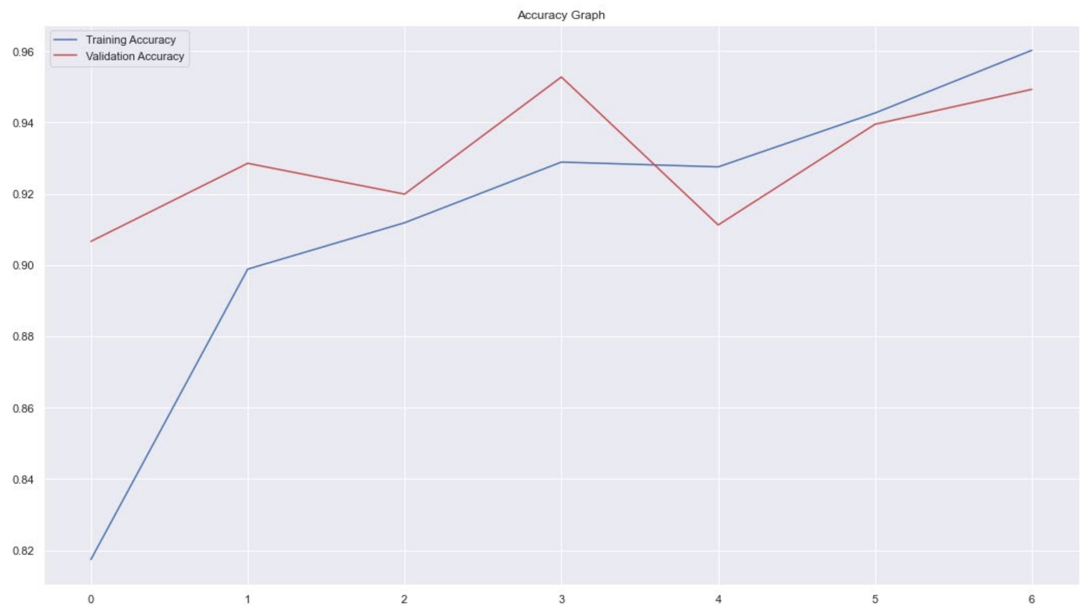


Figure 34 Model Trained Accuracy Graph

The above screenshot shows the model training accuracy and validation accuracy after learning image dataset.

4 . Loss Graph

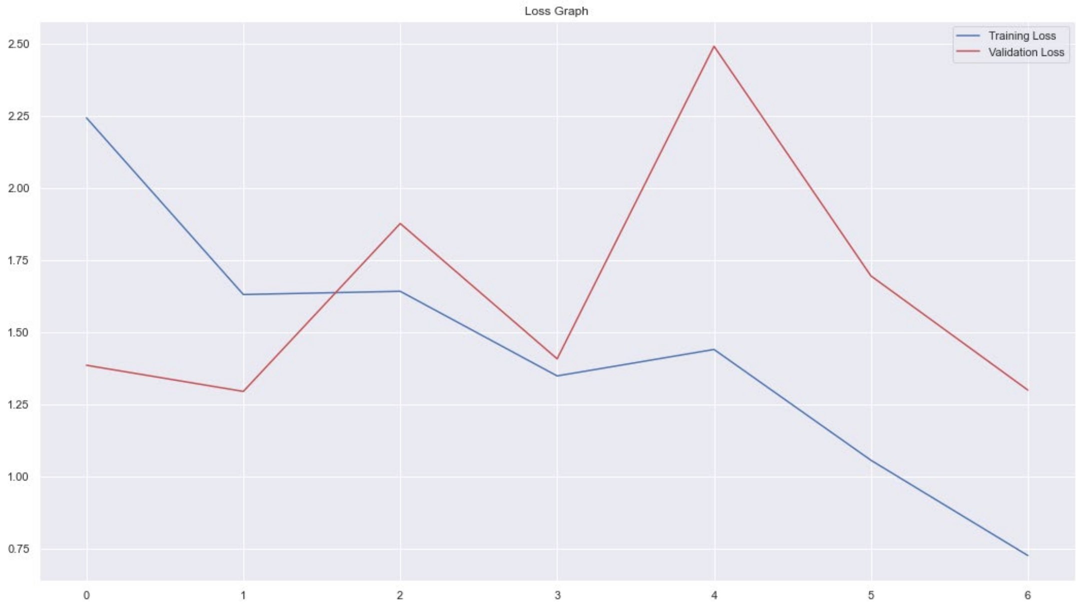


Figure 35 Model Trained Loss Graph

The above screenshot shows the model training loss and validation after learning image dataset.

5 . Confusion Matrix

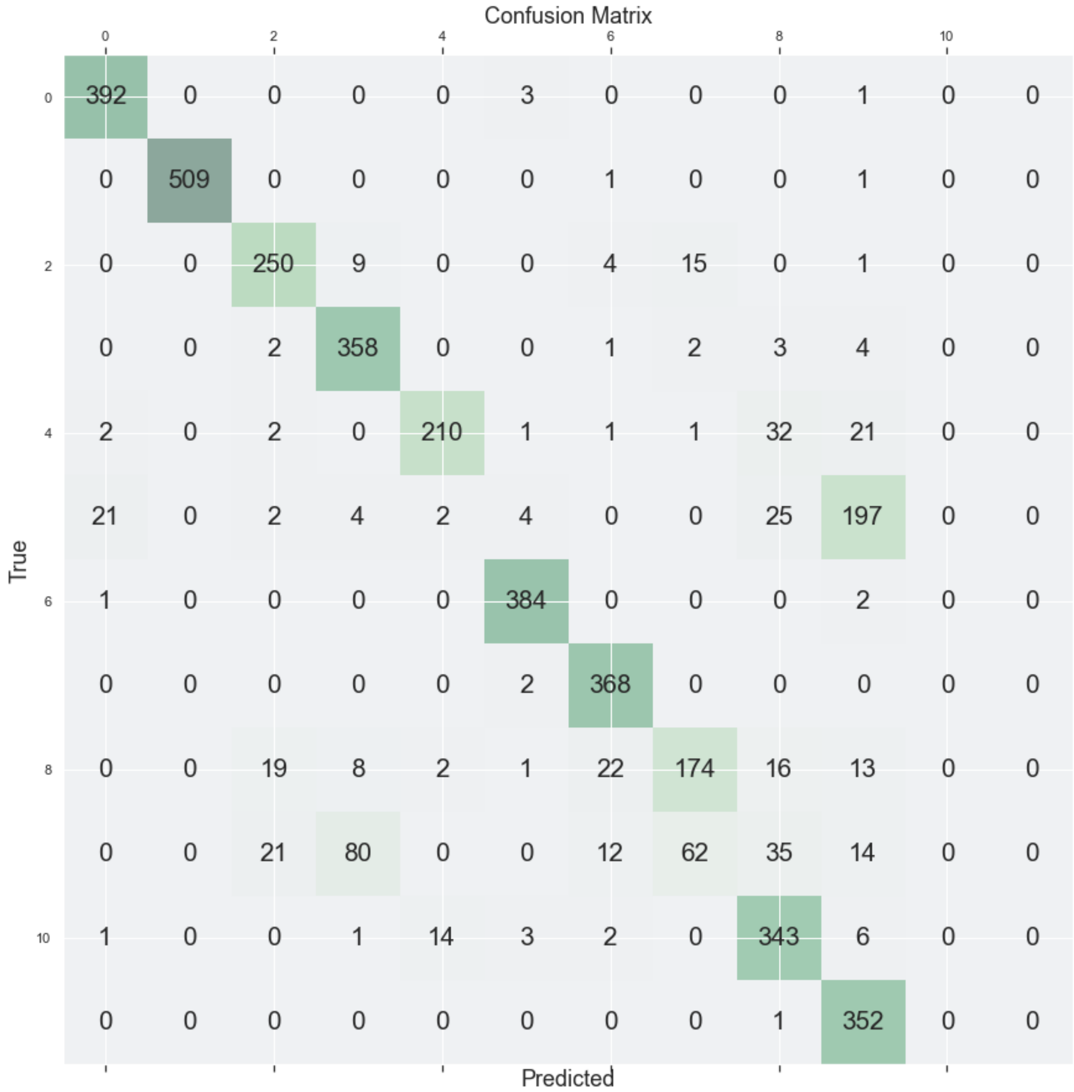


Figure 36 Confusion Matrix Predicted

6 . Results: Comparison of Models

Table 3 Results of comparison of models

Architecture	No. of Parameters	Top 5 Accuracy	Top 1 Accuracy
CNN	47,874,634	0.85%	0.88%
VGG16	14,965,578	0.96%	0.96%
ResNet50	24,591,242	0.71%	0.67%
Xception	21,865,010	0.94%	0.94%

*This is comparison of all models used in this project.
*Best accuracy is of vgg16 model

VI. Conclusion & Future Work

A. Conclusion

Food Inspection through image by using application software to measure the quality of food is research-oriented work, common person not normally identify or judge the food quality by naked eye is so difficult sometime, and many diseases spread around the globe. So, this system become provides benefit the researchers as well as industry of agricultures also.

So, we have designed this offline desktop application which helps the common people to examine their food before having it. People can find the quality and level of nutrients of their food on their own. This offline desktop application is designed by using Artificial Intelligence, Machine learning, tensorflow and image processing.

Through this offline desktop application, the common people who can't afford expensive food quality detecting devices will become able to examine their food before having it and they will get fresh and quality food. We have designed the user-friendly user interface for this application so that anyone can interact with it easily. They will live a healthy life with their family. When one will live a healthy life then the disease rate will automatically decrease.

We have done so much research about foods, different food and their nutrients and minerals. After that we made research that either common people are getting these nutrients or not. We have visited different farms, food courts and many places for our research and data sets. After search about the nutrients of foods we saw that there are so many people who have a lack of energy in their body as compared to their age and BMI. So, we make further research about it to find the cause.

To achieve better accuracy ratio, we collected a huge number of foods images through our research so as we can avoid conflict risks.

We have entered a large dataset in our software database in line to achieve high accuracy. When user opens application, it will show this welcome window. User can pick the image of food from the gallery, or he can click a real time image of their food to analyze it.

So, we have designed this offline desktop application which helps the common people to examine their food before having it. People can find the quality and level of nutrients of their food on their own. This offline desktop application is designed by using Artificial Intelligence, Machine learning, tensorflow and image processing.

B. Future Direction and Suggestion

Future work will aim on improving performance the model classifier by labelling the defective areas more accurately and tune the model more to achieve perfection in all results. A significant number of source data is necessary to construct a more robust and accurate fruit freshness measurement deep learning model, which is popular deep learning method. Noises should be included in the data.

Even though the developed model functioned effectively and was capable of achieving the required results, but there are no photos of fruits and vegetables taken in the natural environment, as contrast to what is utilised in this project. Secondly the dataset utilised lacked fruit variance within a category, such as add more fruits and vegetables images in the dataset. The future suggestion of this project is that we intend to develop a smartphone application that detects the quality of fruits, vegetables and labels them appropriately, also detect its freshness and rottenness. Another goal is to extend the data collection to cover more fruits and vegetables. This is a longer procedure since we want to include items that were not included in most other category

VI. REFERENCES

BOOKS

Scherer, D., Muller, A., & Behnke, S. (2010). **Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition**. International Conference on Artificial Neural Networks (ICANN).

Schmidt F, W. Y. (2020). **Measuring SARS-CoV-2 neutralizing antibody activity using pseudotyped and chimeric viruses** . Journal of Experimental Medicine, 217.

Treadaway, C. (2007). **Digital Crafting and Crafting the Digital**. An International Journal for All Aspects of Design.

kta Sonwani, U. B. (2022). **An Artificial Intelligence Approach Toward Food Spoilage Detection and Analysis**. Digital Public Health,.

Kingma, D., & Ba, J. (2014). **A Method for Stochastic Optimization. Machine Learning**.

ARTICLES

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., . . . Asari, V. K. (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches**. Computer Vision and Pattern Recognition**, 23-26.

Colucci, D., Lia, M., Xiaoyang, Z., & Fabrizio, L. (2020). An automatic computer vision pipeline for the in-line monitoring of freeze-drying processes. **Computers in Industry**, 0166-3615.

Deng, J., Dong, W., Socher, R., Li, L.-J., & Li, K. (2009). ImageNet: a Large-Scale Hierarchical Image Database. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 20-25.

Kruskal, W. H., & Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. **Journal of the American Statistical Association**, 583-621.

M. K. Prem Kumar, A. P. (2020). Quality Grading of the Fruits and Vegetables Using Image Processing Techniques and Machine Learning: **A Review. Advances in Communication Systems and Networks**, 477–486.

ELECTRONIC SOURCES

URL-1 Dang, A. T. (2012). Top 10 CNN Architectures Every Machine Learning Engineer Should Know. Retrieved from https://towardsdatascience.com/: https://towardsdatascience.com/top-10-cnn-architectures-every-machine-learning-engineer-should-know-68e2b0e07201

URL-2 Gilani, R. (2020, 6 13). Main Challenges in Image Classification. Retrieved from towardsdatascience: https://towardsdatascience.com/main-challenges-in-image-classification-ba24dc78b558

URL-3 BALOCH, A. (n.d.). Fresh and Stale Images of Fruits and Vegetables for classification. Retrieved from kaggle: https://www.kaggle.com/datasets/alibaloch/vegetables-fruits-fresh-and-stale

RESUME

Introduction:

Name Surname: Abdul Khalique Baloch

Education:

2015-2019 - BCS (Computer Science) University of Sindh
2020-2022 - AIDS (Artificial Intelligence and Data Science) İstanbul Aydin University

Work Expression:

Software Developer and Instructor **-** Feb 2019 – Sept 2019

Skills, Programming	HTML5, CSS, PHP MySQL, JavaScript, Bootstrap Framework, SharePoint, SharePoint Webpart, WordPress Theme Development, Python, MATLAB , Facebook API, Twitter API, GitHub API
Software's	Adobe Photoshop, Visual Studio Code, Spider, Jupiter, Anaconda
DBMS Used	MySQL, SQL

Languages:

-Sindh: Native Language

-Urdu: Advanced

-Turkish: Intermediate

-English: Advanced