ADINA INSTITUTE OF SCIENCE AND TECHNOLOGY SAGAR (M.P.)

Department Of Computer Science And Engineering

SESSION: 2024-2025

A MINOR PROJECT REPORT

ON

"FRUIT DETECTION SYSTEM"

Submitted in partial fulfillment for the required of Bachelor's Degree in

Computer Science And Engineering

RAJIV GANDHI PRODYOGIKI VISHWAVIDHYALAYA , BHOPAL

(University Of Technology in Madhya Pradesh)

Submitted by:-

Pavan Kushwaha

(0612CS221049)

Mr. Vaibhav Jain	Mr. Rajneesh Pachouri	Dr. Abhishek Jain
(Project Guide)	(H.O.D)	(Principal)
CSE Dept.	CSE Dept.	A.I.S.T.

ADINA INSTITUTE OF SCIENCE AND TECHNOLOGY SAGAR (M.P.)

Department Of Computer Science And Engineering

SESSION : 2024-25

Certificate

This is hereby certified that PAVAN KUSHWAHA (0612CS221049) of Computer Science & Engineering. Department of this college have submitted the project "FRUIT DETECTION SYSTEM" for the partial fulfillof his/her Bachelor Degree under the guidance of Asst. Prof. Mr. Vaibhav Jain We wish all the best for their future.

(Internal Examiner) **(External Examiner)**

CSE Deptt..

ACKNOWLEDGEMENT

I express my deep sense of gratitude to **Asst. Prof Rajneesh Pachouri** Head, Department of Computer Science and Engineering, Adina Institute of Science and Technology Sagar (M.P), Whose kindness valuable guidance and timely help encouraged me to complete this work.

A special thank to **Asst. Prof. Vaibhav Jain** who helped me in completing the minor project by exchanging her interesting ideas, thoughts which made this work easy and accurate. I am indebted to him for the valuable time he has spared for me during this work.

I express my thanks to **Dr. Abhishek Jain (Principal)** for extending his support. I am also grateful to **Dr. Sunil Jain (Director**), who provided all the official facilities and guidance to me. I also thank the authors whose works have been consulted by me during the research.

I would also thank my institution and my faculty members without whom this project would have been a distant reality.

I take this opportunity to express my regards and obligation to my father and other family members whose support and encouragement I can never forget in my life. At last but not the least I want to thank my best friend who appreciated me for my work and motivated me and finally to god who made all the things possible.

Pavan Kushwaha **0612CS221049**

ADINA INSTITUTE OF SCIENCE & TECHNOLOGY SAGAR (M.P.)

Department Of Computer Science And Engineering

SESSION : 2024-25

DECLARATION

I/We, hereby declare that the work, which is being presented in the Project Synopsis Report entitled **"Fruit Detection System"** is my/our own work, carried out under the noble guidance of my/our guide **Mr. Vaibhav Jain**, Department of Computer Science & Engineering, Adina Institute of Science and Technology, Sagar (M.P.).

The matter embodied in this Project Report work has not been submitted in any other university for the award of any degree.

Pavan Kushwaha (0612CS221049)

Approved by

Dr. Abhishek Jain	**Mr. Rajneesh Pachouri**	**Mr. Monali Sahoo**
Principal	**H.O.D.**	**Project Guide**
A.I.S.T. Sagar	**CSE deptt.**	**CSE deptt.**

ABSTRACT

Fruits are very important in day-to-day life. To be healthy we need to take some fruits based on the vitamins. Nowadays people are now giving less importance to Fruits. So, we have created a model to detect fruits and vitamins which will be useful for this generation. The ultimate aim of this paper is to build a fast and accurate fruit detection system. The proposed project can detect some rare and unknown fruits. In the moment's world where everyone is health conscious, the capability to identify fruits by name is veritably important in the food industry. The request includes various different sorts of fruits. still, relating the stylish quality fruit is a daunting task. thus, we developed a system to fete fruits under natural light conditions. The styles used are texture recognition styles, color recognition styles, and shape recognition. This methodology uses image segmentation to identify specific fruits.

This helps us to gain some extra Knowledge. The project is mainly useful for fruit Industries to allocate the fruit in a particular box. It is an offline application. Fruit detection is used to reduce the effort of humans.

To identify the fruit types used on artificial neural networks. It is the best model for the deep learning technique to easily identify the fruit types. Deep Learning models, or to be more specific, TensorFlow uses Convolutional Neural Networks (CNN) models are proposed in this project. A comparison is done between proposed algorithms and current algorithms reveals that the accuracy of fruits detection using CNN is greater than other algorithms. It is predicted that the success obtained results will increase if the CNN is added with extra feature extraction methods to detect fruits type successfully.

Thus higher accuracy is achieved and able to detect fruits and vitamins present on that fruits and deploying the model in a web application using Django Framework.

The Fruit Detection System aims to identify and classify fruits in real-time using machine learning techniques. This system leverages computer vision algorithms and deep learning models to detect different types of fruits from images or video feeds, making it suitable for various agricultural applications such as sorting, quality control, and automated harvesting.

By training the model with a large dataset of fruit images, the system can achieve high accuracy in recognizing various fruits, even under different lighting conditions or partial occlusions. This system will contribute significantly to the agricultural industry, improving productivity, reducing human labor, and ensuring quality control in fruit production. Keywords Fruit detection, web development, HTML, CSS, JavaScript, TensorFlow.js, convolutional neural networks, machine learning,image classification.

INDEX

INTRODUCTION

1.1 SIGNIFICANCE

Object detection and recognition is a demanding work belonging to the field of computer vision. Objects in the images are detected and recognized using machine learning models when trained on a sufficient number of available images. When we have a large number of training images, the accuracy of object recognition is improved. This concept motivates us in developing such a model which can recognize a fruit or a vegetable and predicts its name. There may be a variety of applications of fruit recognition in agricultural work when we are to recognize thousands of fruit images in a less amount of time. It can also be applied in automating the billing process at a fruit shop where the model can recognize the fruit and calculate its price by multiplying with weight. In this project, we will recognize the fruit where the Convolutional Neural Network will predict the name of the fruit given its image. We will train the network in a supervised manner where images of the fruits will be the input to the network and labels of the fruits will be the output of the network. After successful training, the CNN model will be able to correctly predict the label of the fruit.

1.2 BACKGROUND

Although many researchers have tackled the problem of fruit detection, the problem of creating a fast and reliable fruit detection system persists. This is due to high variation in the appearance of the fruits in field settings, including colour, shape, size, texture and reflectance properties. Furthermore, in the majority of these settings, the fruits are partially abstracted and subject to continually-changing illumination and shadow conditions.Various works presented in the literature address the problem of fruit detection as an image segmentation problem (i.e., fruit vs. background). A fruits recognition system supported a contemporay deep learning technique (2019) Dang Thi Phuong Chung andDinh Van Tai during this project they have made to identify the image of the fruit and for that the have designed a dataset named- Fruit 360. With the assistance of that dataset they have completed their project.

This paper briefly discusses the employment of deep learning (DL) for recognizing fruits and its other applications. The paper will also provide a concise explanation of convolution neural networks (CNNs) and the Efficient Net architecture to recognize fruit using the Fruit 360 dataset. The results show that the proposed model is 95% more accurate. This paper explores a fruits recognition classifier based on Efficient Net algorithm. The recognition rate has dramatically improved throughout the experiment.

Using deep learning to recognise fruits from photos (2018) Mihai Oltean (13 Mures) Mihai Oltean (13 Mures) Mihai Oltean (13 Mures They finished their project by employing the dataset Fruit-360 and CNN in their project. They present a fresh, high-quality collection of photos containing fruits in this work. They also show the results of a numerical experiment that was used to train a neural network to detect fruits.

They highlighted why we chose fruits for this research and suggested a few applications that could benefit from such a classifier. Fruit and vegetable quality assessment using computer vision (2021) Anuja Bhargava, Atul Bansal The application of image processing and computer vision technology in the food and agriculture industries is the subject of the project paper. Size, colour, form, texture, and flaw are the most essential quality features of agricultural products.

1.3 SCOPE

Time consumption is the most important factor in any field thus, thus system allows to save time by using and automatic and effective method of evaluation for fruits and vegetables using machine learning technique. This system will allow various application in future and otherwise as the existing classifier can be used in various applications such as user friendly, a self checkout system for the visually impaired customers at the markets, sorting of products before releasing to the markets.

We have trained a Convolutional Neural Network Model which is capable of identifying fruits and vitamins. This model will be useful in sorting the fruits in current companies working in the Augmented Reality field. Google Lens is online we need network connectivity to search but in our project, we don't need internet and it's an offline application that will be user-friendly.

Advantages:

The proposed project is able to detect the fruits type and vitamins.

This helps us to gain some extra knowledge.

Fruit detection will be able to reduce the current ongoing problems.

It reduces confusion among the particular fruit . Its an offline application.

LITERATURE REVIEW

2.1 HISTORY:

All the projects which we have referred they were only able to detect the various types of fruits and vegetables. The projects were only limited till the detection part only. Dataset Fruit-360 and CNN was used in all the below mentioned projects. Image processing and computer technology are also used in these projects.

2.2 COMPARISON WITH EXISTING IMPLEMENTATIONS

2.1.1. A fruits recognition system based on a modern deep learning technique (2019) Dang Thi Phuong Chung and Dinh Van Tai

In this project they have made to identify the image of the fruit and for that have designed a dataset named- Fruit 360. With the help of that dataset they have completed their project. This paper briefly discusses the use of deep learning (DL) for recognizing fruits and its other applications. The paper will also provide a concise explanation of convolution neural networks (CNNs) and the Efficient Net architecture to recognize fruit using the Fruit 360 dataset. The results show that the proposed model is 95% more accurate. This paper explores a fruits recognition classifier based on Efficient Net algorithm.

The recognition rate has dramatically improved throughout the experiment. Among all the cases, the model achieved the best test accuracy of 98% in case 4 from 11 to 15 epochs and best training accuracy of 96.79% at epoch 13. This type of higher accuracy will cooperate to stimulate the overall performance of the machine more adequately in fruits recognition. In the future, our plan is to improve recognition system by extending its functions to process and recognize more variety of different fruit images.

2.2.2 Fruit recognition from images using deep learning (2018) Mihai Oltean, Horea 13 Mures

In this project they have used the dataset Fruit-360 using CNN and completed their project. In this paper they have introduce a new, high-quality, dataset of images containing fruits. They also present the results of some numerical experiment for training a neural network to detect fruits. They have discussed the reason why we chose to use fruits in this project by proposing a few applications that could use such classifier.

They have also made some numerical experiments by using TensorFlow library in order to classify the images according to their content. The main objectives for the future is to improve the accuracy of the neural network.

2.2.3 Fruits and vegetables quality evaluation using computer vision (2021) Anuja Bhargava, Atul Bansal

The project paper highlights the use of image processing and computer vision technology in the field of food industry and agriculture. The most important quality characteristics of agricultural products are size, color, shape, texture and defect. To replace manual inspection of food, computer vision system is used which provide authentic, equitable and non-destructive rating. The computer vision based quality inspection comprises of four main steps, namely, acquisition, segmentation, feature extraction and classification. The paper have made an attempt has been made to explore and compare the various methods/algorithms proposed by researchers in each step. It can be concluded from the extensive survey carried out. In the literature the images of fruits and vegetables are captured mainly from one direction. However, the system performance may improve by considering the images of fruits and vegetables captured from different directions. Authors have utilized different color spaces for the color based feature extraction, still one may explore combination/ other color space to improve the performance

2.2.4 Fruits and Vegetables Classification using Progressive Resizing and Transfer Learning (2021) Kishore M, S. B. Kulkarni, K. Senthil Babu

They have proposed automatic and an effective method of evaluation for fruits and vegetables using Machine Learning techniques. The algorithm used here does not require human intervention, and the system has higher accuracy compared to the human-involved systems because it uses an automated computer algorithm. The fruits and vegetables classifier is efficient, non-destructive, 14 and accurate which reduces man pow- er. The existing classifier can be used in various applications such as user friendly, a self-checkout system for the visually impaired customers in supermarkets, sorting of products before releasing it to the markets. Through their analyses, they were able to develop several models that classify fruits based on an image. Given the nature of our dataset, it was crucial that we first started by pre-processing our data using progressive-resizing. By implementing this feature extraction method, which reduced the dimensionality of our dataset on scaling them up and fine- tuning performance, we were able to achieve high accuracies, even with our plain dataset models. From there, we developed more complex neural networks, which gave us our best accuracy scores.

2.3 PROBLEM DEFINITION

In this day and age, Data is very crucial in every sector and the sorting and grading can be done by human but it is inconsistent, time consuming, variable, subjective, onerous, expensive and easily influenced by surrounding. The human intervention also causes errors and inaccuracy. Time consumption is the most important factor hence, this issue was the main concern of our project. Therefore, this system allows to save time by using an automatic and an effective method of 11 evaluation for fruits and vegetables using Deep Learning techniques such as Neural networks and transfer learning.

Develop the fruit recognition systems using Deep literacy neural networks and make a stylish model that will identify fruit from recognition system models. make the recognition of the fruits using cooperative filtering with stochastic grade descent. In the moment's world where everyone is health conscious, the capability to identify fruits by name is veritably important in the food industry.

SYSTEM REQUIREMENT AND ANALYSIS

3.1 SOFTWARE REQUIREMENTS

The experiment setup is carried out on a computer system which has the different software specifications as given in Table 3.3 The software used to build a fruits and vegetables classifier is Keras and Tensorflow. It is an open-source library that is written in the programming language Python. Data sets that we would be using includes fruits 360 because of its wide variety.

Specifications

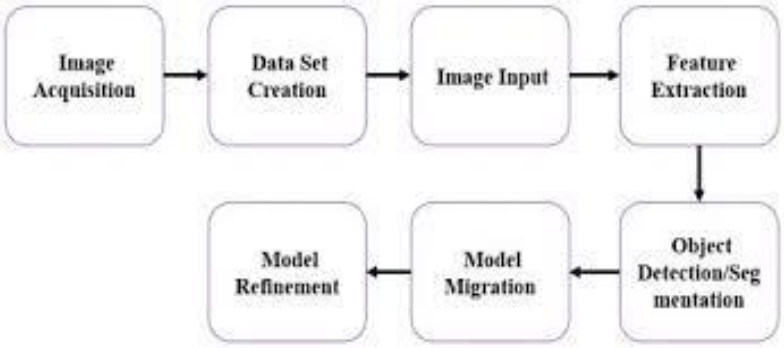
Processor	**Intel**
HDD	**215 GB**
RAM	**Minimum 4 GB**

Software details:

Operating System	Windows 10
Programming Language	Python 3.8
Libraries	TensorFlow & Machine Learning Libraries
Software/ Package Manager	Anaconda
Datasets	Fruits 360 Kaggle

METHODOLOGY

4.1 BLOCK DIAGRAM



4.2 Proposed Architecture:

The aim of the proposed algorithm is to implement fruits and vegetables classification by using deep learning neural network framework – CNN and highlight the importance of deep learning technology in classification problems. The proposed algorithm is consisting of six main steps.

1) Image collection : Receiving the input image

2) Data set splitting :training and test models

3) Feature extraction: Convolution 18

4) Normalization : Reducing pixel values between 0 and 1

5) Train CNN algorithm : A convolution neural network has multiple hidden layers that help in extracting information from an image.

6) Classify using CNN algorithm : Classifying the Species in an image The system comprises of two stages, training and testing. In the training stage, a set of images are provided as visual examples. In the testing stage, image i.e. the test image is given as input to the classifier. With the help of the knowledge gained from training, the test image is accordingly classified into the most favorable class. A. Receiving the input image In the proposed system, the test image is fed as input which is then converted into a binary pattern.

A set of previously labelled images are present in the dataset whose features are matched with those of the test image, in order to determine the species of the fruit or vegetable present. B. Feature Extraction The test image which is received as input can be transformed into a reduced set of features.

The selected features may contain the significant information from the input data, due to which the desired task can be performed by using this reduced amount of data instead of the initial unaltered data.

Deep neural networks recognize features from images, and determine multiple levels of representation, with higher-level features depicting more abstract characteristics of the data. C. Classifying the Species in an image For the task of

classification of species, the corresponding output layer generates the probabilities of the fruit or vegetable detected in the image belonging to one of the possible classes. Even though providing such a result would save a huge amount of human effort that will be needed in recognizing the correct species, the testing of this hypothesis will require human knowledge. 3.2 Implementation Details To detect an object, we need to have some idea where the object might be and how the image is segmented.

To recognize the shape of an object, we need to know its location, and to recognize the location of an object, we need to know its shape. The location and size is typically defined using a bounding box, which is stored in the form of corner coordinates. Using a rectangle is simpler than using an arbitrarily shaped polygon, and many operations, such as convolution, are performed on rectangles in any case. The sub-image contained in the bounding box is then classified by an algorithm that has been trained using deep learning. So, the proposed work is implemented with libraries keras, tensorflow, matplotlib and other mandatory libraries. Deep learning algorithm is applied is CNN.

Transfer Learning:

Transfer learning is used to reduce the computation as the time to retrain the whole model can be reduced. The machine uses the previous information to increase the accuracy of predictions. In transfer learning, the first layers are kept the same and only task-specific layers are retrained. Transfer learning also reduces memory use. The proposed algorithm consists of six main steps.

1) Image collection: Receiving the input image

2) Data set splitting: training and test models

3) Feature extraction: Convolution 18

4) Normalization : Reducing pixel values between 0 and 1

5) Train CNN algorithm: A convolution neural network has multiple hidden layers that help in extracting informationfrom an image.

6) Classify using CNN algorithm: Classifying the Species in an image The dataset comprises two sets, training, and testing. In the training stage, a set of images are provided as visual examples and the machine is trained to recognize those images. In the testing stage, the image is given as input to the classifier to check whether it is able to detect the images according to

A. Receiving the input image: In the proposed system, the test image is used as input. The RGB image matrix is flattened and then divided by 255 so that the values are in between 0 to 1 to increase the accuracy.

A set of trained images are present in the dataset whose features are matched with those of the test image, in order to determine the species of the fruit or vegetable present. Greater the accuracy more is the success of system

B. Feature Extraction The test image which is received as input is segmented to extract the features. The selected features contain the task specific characteristics due to which the desired task can be performed. Deep neural networks recognize these features from images, and determine the number of levels of representation. The distinct characteristics are more easily detected.

C. Classifying the Species in an image For the task of classification of species, the machine selects all the probabilities of the fruit or vegetable matched with the image and gives the answer with highest accuracy.

4.3 Samples used

Fruits 360 dataset, Version: 2020.05.18.0 was used. A total of 67692 images were taken for training and 22688 images for testing. There are total of 131 classes of fruits and vegetables which includes common ones such as lemon, orange, strawberry, and rare varieties like huckleberry, raspberry, kaki, clementine and more

.

Kaggle 360 randomly placed fruit dataset

Each fruit type is its unique folder which is accessible to the code.Then the training takes place after placing and downloading the dataset properly.

All fruits have an unique folder.

There are 3 channels (red, green, blue) , 100*100 image size. Each value represents the colour intensity with respect to the channel colour.

Training and Validation Dataset:

Next, we will split the data randomly to achieve 3 sets of data:

. 1.Training Set : Train the model

2.Validation Set: Evaluate the model

3. Test Set: Report the final accuracy of the model

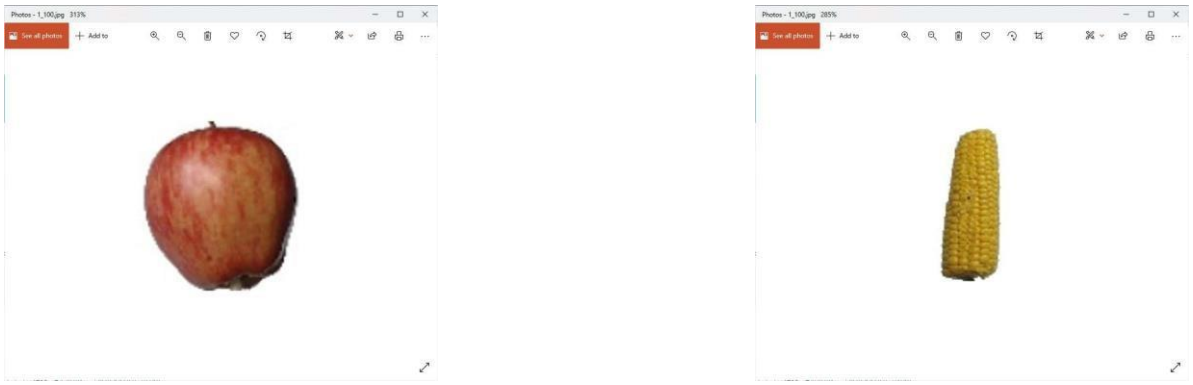
Size of training dataset: 67,692

Size of validation dataset: 22,688

As we have a total of 57, 692 training images, we should split our images into smaller batches before training our model using Model fit Generator. Working with a smaller set of data reduces memory space and in turn, increases the speed of training.

For our dataset, we will use a batch size of 16.

Trained Apple Red 1 Dataset



Apple red and corn trained dataset

Convolutional Neural Networks:

CNN:- Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

- **Convolutional layer**
- **Pooling layer**
- **Fully-connected (FC) layer**

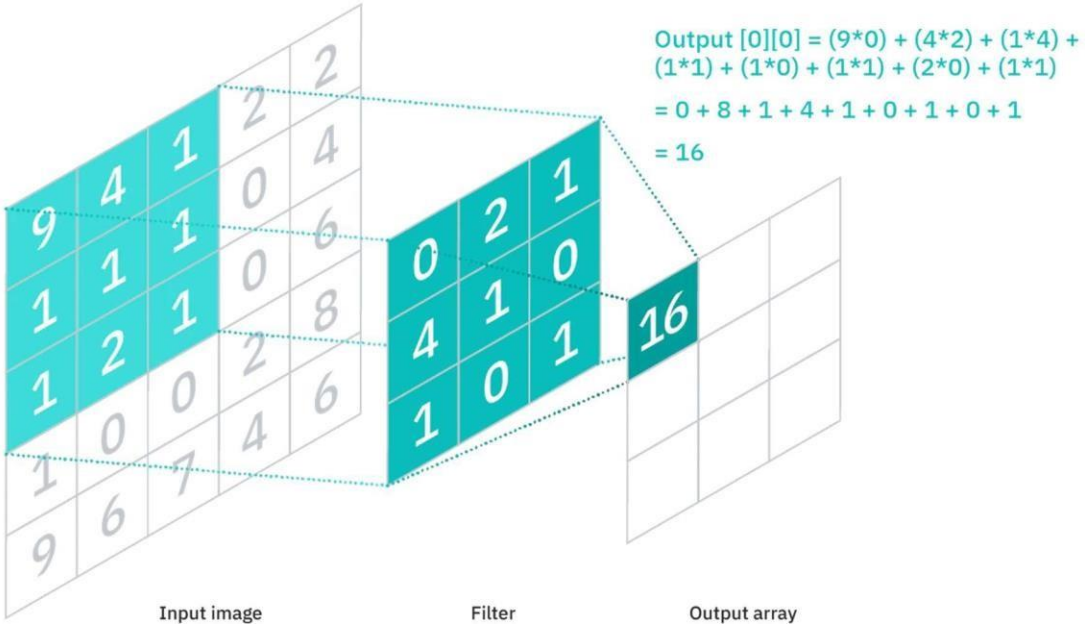
The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully- connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image.

Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

Convolutional Layer

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution. The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image.

While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.



Convolution

As you can see in the image above, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as "partially connected" layers. However, this characteristic can also be described as local connectivity.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent.

However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.

2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.

3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

● **Valid padding**: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.

● **Same padding**: This padding ensures that the output layer has the same size as the input layer

● **Full padding**: This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

As we mentioned earlier, another convolution layer can follow the initial convolution layer. When this happens, the structure of the CNN can become hierarchical as the later layers can see the pixels within the receptive fields of prior layers. As an example, let's assume that we're trying to determine if an image contains a bicycle.

You can think of the bicycle as a sum of parts. It is comprised of a frame, handlebars, wheels, pedals, et cetera. Each individual part of the bicycle makes up a lower-level pattern in the neural net, and the combination of its parts represents a higher-level pattern, creating a feature hierarchy within the CNN.

Pooling Layer

Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

● Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

● Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

Fully-Connected Layer

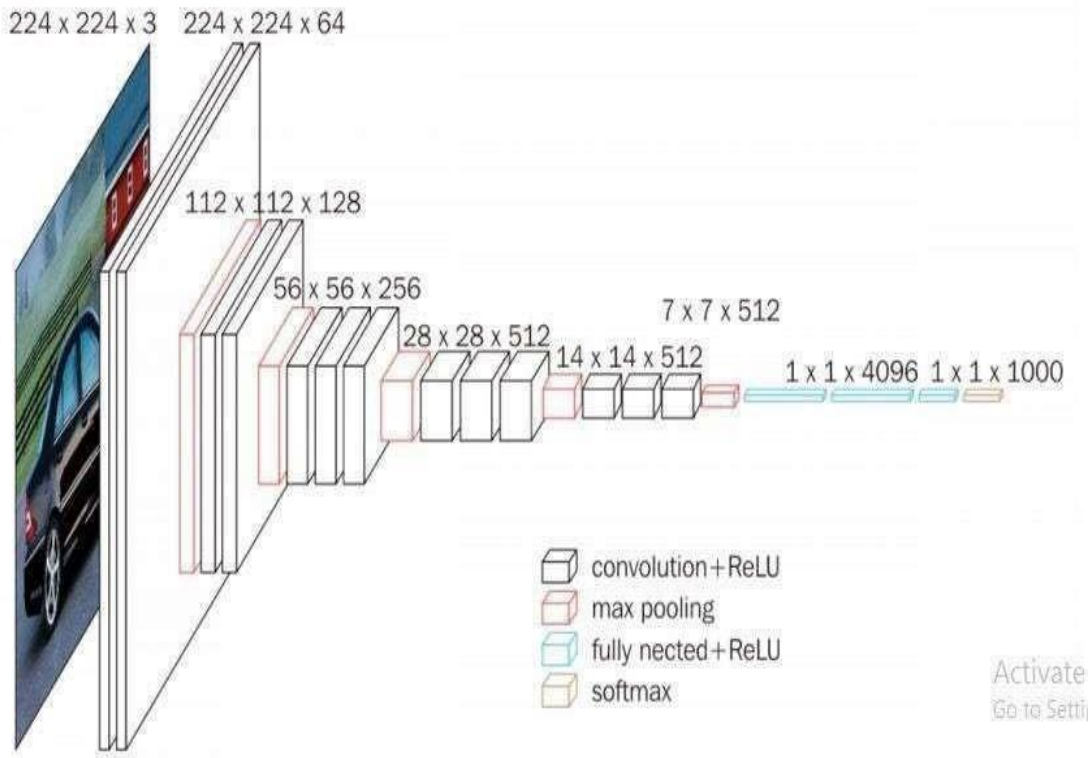
The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

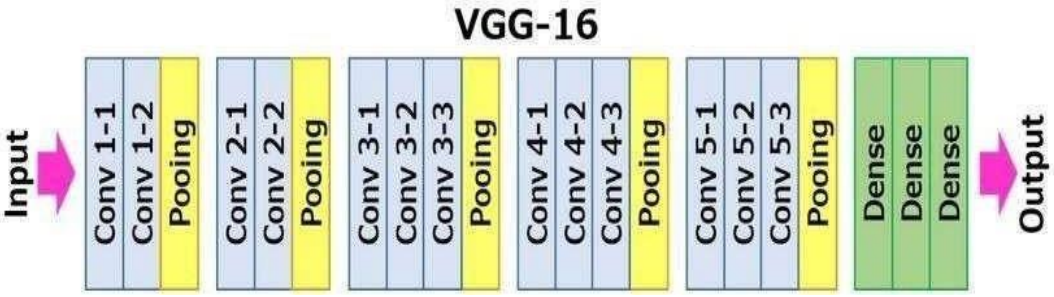
This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1

VGG16:- A convolutional neural network is also known as a ConvNet, which is a kind of artificial neural network. A convolutional neural network has an input layer, an output layer, and various hidden layers. VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date.

The creators of this model evaluated the networks and increased the depth using an architecture with very small (3 × 3) convolution filters, which showed a significant improvement on the prior-art configurations. They pushed the depth to 16–19 weight layers making it approx — 138 trainable parameters.

VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.


224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution+ReLU
max pooling
fully nected+ReLU
softmax
Activate
Go to Setti


VGG-16
Input
Conv 1-1
Conv 1-2
Pooing
Conv 2-1
Conv 2-2
Pooing
Conv 3-1
Conv 3-2
Conv 3-3
Pooing
Conv 4-1
Conv 4-2
Conv 4-3
Pooing
Conv 5-1
Conv 5-2
Conv 5-3
Pooing
Dense
Dense
Dense
Output

● The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.

● VGG16 takes input tensor size as 224, 244 with 3 RGB channel

● Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2.

● The convolution and max pool layers are consistently arranged throughout the whole architecture

● Conv-1 Layer has 64 number of filters, Conv-2 has 128 filters, Conv-3 has 256 filters, Conv 4 and Conv 5 has 512 filters.

● Three Fully-Connected (FC) layers follow a stack of convolutional layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer.

SOFTWARE IMPLEMENTATION

SOFTWARE IMPLEMENTATION

The code of the project is written in the “Python”.

Basics of Python:

Compiling

Since Python is an interpreted language, compilation of Python code can mean many things, including compilation to byte code or transformation to another language. When one learns how to take Python code and compile it into an executable for Windows platforms, one can create a Python program and have Windows users seamlessly run it. This is done by first writing a error free code then saving it as “mysetupfile.py”.

Run the mysetupfile.py file through the py2exe program: c:\Python> Python mysetupfile.py py2exe.Wait until the py2exe compiler finishes producing its output. If all goes well, the compiler may inform you of the DLLs that your Python program is dependent on once it finishes compiling.

Look in the directory to see an exe of the project. Test the compile by running the program: c:\Python> mycode.exe. This is how the compilation takes place.

Applications

There are various applications of python such as in web development, Machine Learning, Data science, Game development etc. Python is just the perfect option for software development. Popular applications like Google, Netflix, and Reddit all use Python.

This language offers amazing features like:

- Platform independence
- Inbuilt libraries and frameworks to provide ease of development.
- Enhanced code reusability and readability
- High compatibility

Apart from these Python offers enhanced features to work with rapidly growing technologies like Machine learning and Artificial intelligence. All these embedded features make it a popular choice for software development.

Overview

This is truly a stand-alone programmable system. The main reason behind this is its extensive libraries and frameworks that fuel up the process. Further, it provides ease in building a plethora of applications, web development processes, and a lot more.

Class:- Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its class) for modifying its state.

Compared with other programming languages, Python's class mechanism adds classes with a minimum of new syntax and semantics. It is a mixture of the class mechanisms found in C++ and Modula-3. Python classes provide all the standard features of Object Oriented Programming: the class inheritance mechanism allows multiple base classes, a derived class can override any methods of its base class or classes, and a method can call the method of a base class with the same name. Objects can contain arbitrary amounts and kinds of data. As is true for modules, classes partake of the dynamic nature of Python: they are created at runtime, and can be modified further after creation.

In C++ terminology, normally class members (including the data members) are public ,and all member functions are virtual. As in Modula-3, there are no short hands for referencing the object's members from its methods: the method function is declared with an explicit first argument representing the object, which is provided implicitly by the call.

As in Smalltalk, classes themselves are objects. This provides semantics for importing and renaming. Unlike C++ and Modula-3, built-in types can be used as base classes for extension by the user. Also, like in C++, most built-in operators with special syntax (arithmetic operators, subscripting etc.) can be redefined for class instances.

(Lacking universally accepted terminology to talk about classes, I will make occasional use of Smalltalk and C++ terms. I would use Modula-3 terms, since its object-oriented semantics are closer to those of Python than C++, but I expect that few readers have heard of it.)

Google Colab:- Google Colab was developed by Google to provide free access to GPU'sand TPU's to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.

What is Jupyter Notebook?

Jupyter Notebook is an application that allows editing and running Notebook documents through a web browser or an Integrated Development Environment (IDE). Instead of files, you will work with Notebooks.

Jupyter Notebook is a web-based interactive computing environment that allows users to create, share, and collaborate on documents that contain live code, equations, visualizations, and narrative text.

 # Key Features

1. **Interactive coding**: Jupyter Notebook allows users to write and execute code in a variety of programming languages, including Python, R, and Julia.

2. **Rich media support**: Jupyter Notebook supports a wide range of media formats, including images, videos, and audio files.

3. **Collaboration tools**: Jupyter Notebook provides tools for collaboration, including real-time commenting and version control.

4. **Sharing and publishing**: Jupyter Notebook allows users to share and publish their notebooks, making it easy to collaborate with others and showcase work.

 # Use Cases

1. **Data science and analytics**: Jupyter Notebook is widely used in data science and analytics for data exploration, visualization, and modeling.

2. **Machine learning and AI**: Jupyter Notebook is used in machine learning and AI for building and training models, as well as for data preprocessing and visualization.

3. **Education and research**: Jupyter Notebook is used in education and research for teaching and learning programming concepts, as well as for conducting research and publishing results.

4. **Business and industry**: Jupyter Notebook is used in business and industry for data analysis, visualization, and reporting, as well as for building and deploying machine learning models.

 # Benefits

1. **Improved productivity**: Jupyter Notebook provides an interactive and flexible environment for coding and data analysis, improving productivity and efficiency.

2. **Enhanced collaboration**: Jupyter Notebook enables real-time collaboration and commenting, making it easier to work with others.

3. **Increased transparency**: Jupyter Notebook provides a transparent and reproducible environment for data analysis and modeling, making it easier to share and publish results.

4. **Flexibility and customization**: Jupyter Notebook provides a flexible and customizable environment for coding and data analysis, allowing users to tailor their workflow to their specific needs.

.

Google Colab provides tons of exciting features that any modern IDE offers, and much more. Some of the most exciting features are listed below.

- Interactive tutorials to learn machine learning and neural networks.
- Write and execute Python 3 code without having a local setup.
- Execute terminal commands from the Notebook.
- Import datasets from external sources such as Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud service, GPUs and TPUs.
- Integrate with PyTorch, Tensor Flow, Open CV.
- Import or publish directly from/to GitHub.

Tensorflow:- TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources.

Web development using Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.It has no database abstraction layer, form validation, or any other components where pre- existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on. Web development:

Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

The word Web Development is made up of two words, that is:

• **Web:** It refers to websites, web pages or anything that works over the internet.

• **Development:** Building the application from scratch

.

DISCUSSION AND RESULTS

The complete development of the project was discussed and this system was divided into the following stages:

Problem definition stage;

Designing block diagram;

Testing and Troubleshooting;

Developing algorithm for software;

Writing individual part of software; Compiling the code;

Testing and Running.

Problem definition stage

Nowadays, identification systems are very much in use for various purposes but the accuracy of which varies depending on the models in use. Thus, we have proposed a project in which we are using Convolutional neural networks and Transfer learning to make an efficient fruit and vegetable identification system that identifies each fruit or vegetable and provides its calorie count.

Designing block diagram

At this stage we have categorized the whole system into different individual layers. These convolutional layers are play a major role in every step from start to end.The Block diagram shows the different layers and how the image is extracted and filtered. So the result was the block diagram of the project.

Developing an Efficient Model and Dataset

To get the logical flow of the desired output, the development of Model is having a prominent role. So that we have analyzed the complete requirement and chose the VGG16 model which helped the identification to be as accurate as we aspired. We also went 26 through with Kaggle fruit360 dataset and also tried to make our own dataset by taking images of different varieties of fruits.

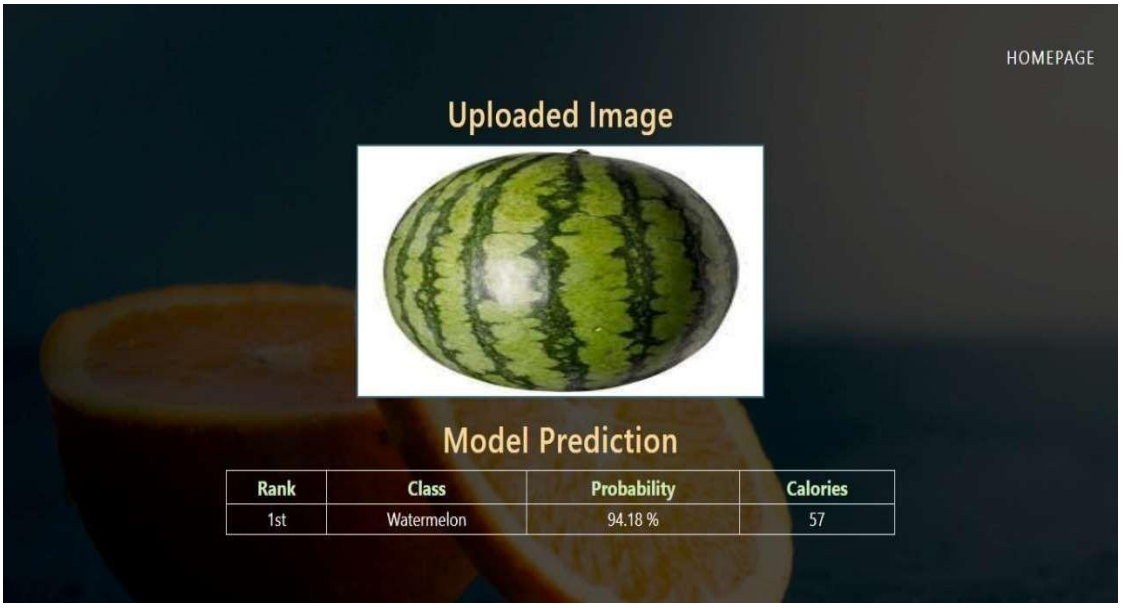
Compiling the code

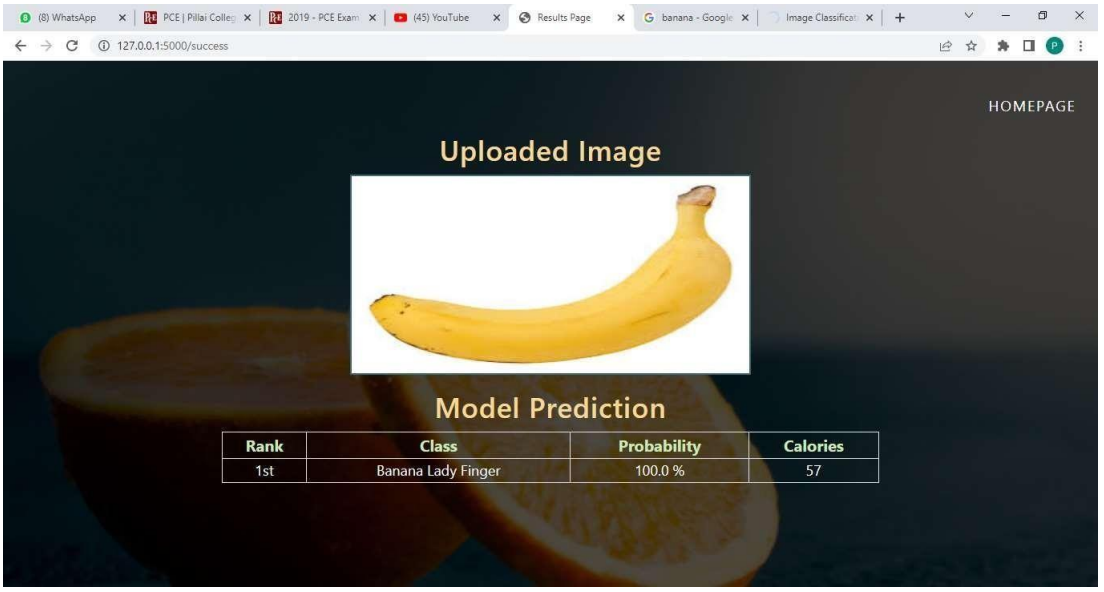
After choosing the Model as VGG16 we used google colab and jupytr as our work base in which using python we trained and tested our dataset. Using Flask we also made a Web Application in order to identify the fruits and vegetables by taking input as images.

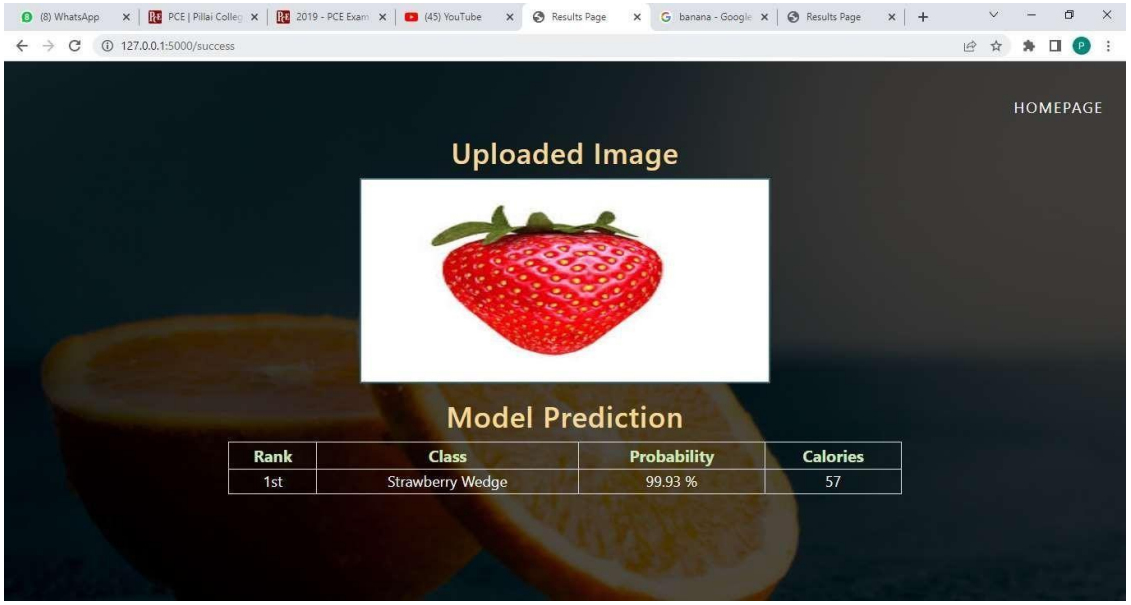
Testing and Running

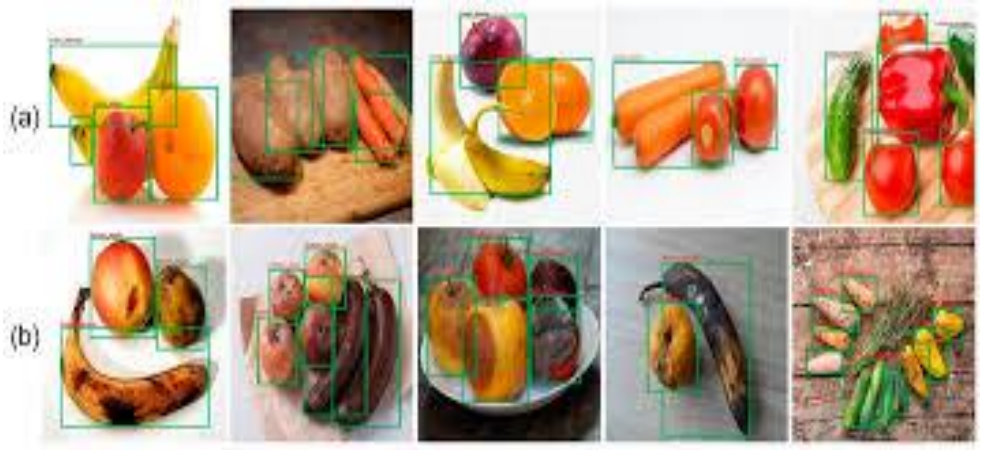
In the Web App we tested many inputs as images to find out the accuracy of the model. Any errors found were removed successfully. This is the last and final stage of development of our project.

Output:


HOMEPAGE
Uploaded Image
Model Prediction
Rank
Class
Probability
Calories
1st
Watermelon
94.18 %
57


127.0.0.1:5000/success
HOMEPAGE
Uploaded Image
Model Prediction
Rank
Class
Probability
Calories
1st
Banana Lady Finger
100.0 %
57


(8) WhatsApp
PCE | Pillai Colleg
2019 - PCE Exam
(45) YouTube
Results Page
banana - Google
Results Page
127.0.0.1:5000/success
HOMEPAGE
Uploaded Image
Model Prediction
Rank
Class
Probability
Calories
1st
Strawberry Wedge
99.93 %
57


(a)
(b)

CONCLUSION

In this project, several computer vision and image-processing approaches adopted for the classification of fruits and vegetables is explored. The survey of literature for image- processing based solutions that use different features for recognition and classification of fruits and vegetables is presented.

The CNN techniques is explained with examples. The approach of making the app using deep learning techniques for classification of fruits and vegetables is proposed. The performance measures like precision and recall are described in this report. The App will also provide the information regarding the Nutrient content of the identified fruit or vegetable. The different standard datasets or variable inputs are defined that may be used in experiment for this domain systems.

The applications of this domain is identified and presented. This will improve public awareness of some uncommon and unfamiliar fruits. The project's primary goal is to reduce human effort and make people's lives simpler. Fruit identification has the potential to reduce the present chronic issue. Reduce the amount of overlap between each fruit.

Developing a web app is a future task that could be added to this project. Users may access this application from anywhere and at any time. Finally, we conclude that we train the dataset model and test the model on multiple fruits. After researching and analyzing various research papers, we found that in the past, various algorithms such as CNN, RCNN, and DCNN achieved 90% accuracy,

The proposed project is able to detect the fruits type and vitamins. This helps us to gain some extra knowledge.Fruit detection will be able to reduce the current ongoing problems. It reduces confusion among the particular fruit. Its an offline application.To identify the fruits type based on artificial neural network.It is the best deep learning model to identify the fruits types and it shows how much of vitamins present.

Also we made some numerical experiments by using TensorFlow library in order to classify the images according to their content. From our point of view one of the main objectives for the future is to improve the accuracy of the neural network.

Tasks to be done in future work:

- In the near future we plan to create a mobile application which will take pictures of fruits and labels them accordingly.To deploy this model as mobile application.
- To implement this model to an AI using voice command feedback.

REFERENCES

[1] Wikipedia F1 Score. 2016. [(accessed on 31 July 2016)]. Available online: https://en.wikipedia.org/wiki/F1_score.

[2] Dang Thi Phuong Chung, Dinh Van Tai A fruits recognition system based on a modern deep learning technique https://www.researchgate.net/publication/336378274

[3] Mihai Oltean, Horea Mures Fruit recognition from images using deep learning https://www.researchgate.net/publication/321475443

[4] Anuja Bhargava, Atul Bansal Fruits and vegetables quality evaluation using computer vision https://www.researchgate.net/publication/325584667

[5] Kishore M, S., B. Kulkarni, K. Senthil Babu Fruits and Vegetables Classification using Progressive Resizing and Transfer Learning https://jusst.org/wp-contents/uploads/2021/02

[6] Stanford University. CS231n: Convolutional Neural Networks for Visual Recognition (2016). Available online: http://cs231n.github.io/transfer-learning/ (accessed on 31 July 2016).

[7] L Deng, G Hinton, and B Kingsbury 2013 New types of deep neural network learning for speech recognition and related applications: An overview IEEE International Conference on Acoustics, Speech and Signal Processing 8599–8603

[8] Y LeCun and Y Bengio Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks 3361(10) 1995

[9] J. L. Joseph, V. A. Kumar, and S. P. Mathew, Fruit Classif cation Using Deep Learning, vol. 756, Springer, Singapore.

[10] J. M. Ponce,A.Aquino,andJ.M.And´ ujar,"Olive-fruit variety classifcation by means of image processing and convolutional neural networks," IEEE Access, vol. 7, pp. 147629–147641, 2019.

[11] R. Yamparala, R. Challa, V. Kantharao, and P. S. R. Krishna, "Computerized classifcation of fruits using convolution neural network," in Proceedings of the International Confer ence on Smart Structures and Systems (ICSSS), Chennai, India, July 2020.

[12] S. Lu, Z. Lu, S. Aok, and L. Graham, "Fruit classifcation based on six layer convolutional neural network," in Proceedings of the International Conference on Digital Signal Processing (DSP), Shanghai, China, November 2018.

[13] W. C. Seng and S. H. Mirisaee, "A new method for fruits recognition system," in Proceedings of the International Conference on Electrical Engineering and Informatics, pp. 130–134, Selangor, Malaysia, August 2009.

[14] O. O. Abayomi-Alli, R. Damaˇseviˇ cius, S. Misra, and R. Maskeliunas, "Cassava disease recognition from low quality images using enhanced data augmentation model and deep learning," Expert Systems, vol. 38, 2021.

APPENDIX

CODE OF THE PROJECT:-

```
fromkeras.applicationsimportvgg16
img_rows = 100
img_cols=100
model=vgg16.VGG16(weights='imagenet',
                  include_top = False,
                  input_shape=(img_rows,img_cols,3))
for (i,layer) in enumerate(model.layers):
   print(str(i)+""+layer.class.name,layer.trainable) for layer
in model.layers:
   layer.trainable=False
for(i,layer)in enumerate(model.layers):
   print(str(i)+""+layer.class.name,layer.trainable) def
addTopModel(bottom_model, num_classes, D=256):
   """createsthetoporheadofthemodelthatwillbe placed
   ontop of the bottom layers"""
   top_model=bottom_model.output
   top_model = Flatten(name = "flatten")(top_model)
   top_model=Dense(D,activation="relu")(top_model)
   top_model = Dropout(0.3)(top_model)
   top_model=Dense(num_classes,activation="softmax")(top_model)
   return top_model
fromkeras.modelsimportSequential
fromkeras.layersimportDense,Dropout,Activation,Flatten
fromkeras.layersimportConv2D,MaxPooling2D,ZeroPadding2D
#from keras.layers.normalization import BatchNormalization
fromkeras.modelsimportModel
num_classes = 131
FC_Head=addTopModel(model,num_classes)
modelnew=Model(inputs=model.input,outputs=FC_Head)
print(modelnew.summary())
fromkeras.preprocessing.imageimportImageDataGenerator
train_data_dir = 'fruits-360_dataset/fruits-360/Training'
validation_data_dir = 'fruits-360_dataset/fruits-360/Test'
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range


    =0.2, horizontal_flip=True,
    fill_mode='nearest')
validation_datagen=ImageDataGenerator(rescale=1./255)

train_batchsize=16
```

```
val_batchsize=10
train_generator=train_datagen.flow_from_directory(
        train_data_dir,
        target_size=(img_rows,img_cols),
        batch_size=train_batchsize,
        class_mode='categorical')
validation_generator=validation_datagen.flow_from_directory(
        validation_data_dir,
        target_size=(img_rows,img_cols),
        batch_size=val_batchsize,
        class_mode='categorical',
        shuffle=False)
modelnew.compile(loss='categorical_crossentropy',
            optimizer = "adam" ,
            metrics=['accuracy'])
nb_train_samples = 67692
nb_validation_samples=22688
epochs= 20
batch_size=16
history=modelnew.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples//batch_size,
    epochs = epochs,
    #callbacks= callbacks,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples//
    batch_size)
modelnew.save("tlfinal.h5")
```

App.py

```
import os
import uuid
import flask
importurllib
fromPILimport Image
fromtensorflow.keras.modelsimportload_model
fromflaskimportFlask,render_template,request,send_file
fromtensorflow.keras.preprocessing.imageimportload_img,img_to_array
app = Flask( name )
BASE_DIR = os.path.dirname(os.path.abspath( file ))
model=load_model(os.path.join(BASE_DIR,'tl2.h5'))
ALLOWED_EXT = set(['jpg' , 'jpeg' , 'png' , 'jfif'])
def allowed_file(filename):
    return'.'infilenameand\
          filename.rsplit('.',1)[1]inALLOWED_EXT
```

```
classes=os.listdir('fruits-360_dataset/fruits-
360/Training') def predict(filename , model):
    img=load_img(filename,target_size=(100,100)) img
    = img_to_array(img)
    img=img.reshape(1,100,100,3)
    img = img.astype('float32')
    img = img/255.0
    result=model.predict(img)
    dict_result = {}
    for i in range(131):
        dict_result[result[0][i]]=classes[i]
    res=result[0]
    res.sort()
    res = res[::-1]
    prob = res[:3]
    cal_result=[]
    prob_result = []
    class_result = []
    for i in range(1):
        prob_result.append((prob[i]*100).round(2))
        class_result.append(dict_result[prob[i]])
    ifclass_result[0]=="AppleBraeburn":
        cal=87
    elifclass_result[0]=="Tomato1":
        cal=33
    elifclass_result[0]=="PotatoWhite":
        cal=77
    elifclass_result[0]=="Kiwi":
        cal=61
    elifclass_result[0]=="Eggplant":
        cal=25
    elifclass_result[0]=="Pear"or"Pear2":
        cal=57
    elifclass_result[0]=="GingerRoot":
        cal=80
    elifclass_result[0]=="GrapefruitWhite":
        cal=42
    elifclass_result[0]=="Pinapple":
        cal=50
    elifclass_result[0]=="Lemon":
        cal=29
    elifclass_result[0]=="Orange":
        cal=47
    elifclass_result[0]=="Strawberry":
```

```
   cal=33
elifclass_result[0]=="Mango":
   cal=135
elifclass_result[0]=="Avacado":
   cal=160
elifclass_result[0]=="AppleGolden3"or"AppleGolden1"or"AppleGolden2": cal=73
elifclass_result[0]=="Blueberry":
   cal=84
elifclass_result[0]=="AppleGrannySmith":
   cal=83
elifclass_result[0]=="AppleRed1"or"AppleRed2"or"AppleRed3": cal=72
elifclass_result[0]=="AppleRedYellow1"or"AppleRedYellow2": cal=60
elifclass_result[0]=="Banana"
   : cal=89
elifclass_result[0]=="CornHusk":
   cal=106
elifclass_result[0]=="Plum"or"Plum2"or"Plum3":
   cal=46
elifclass_result[0]=="Peach"or"Peach2":
   cal=68
elifclass_result[0]=="CucumberRipe"or"CucumberRipe2":
   cal=30
elifclass_result[0]=="PearStone":
   cal=101
elifclass_result[0]=="PeachFlat":
   cal=50
elifclass_result[0]=="Fig":
   cal=37
elifclass_result[0]=="Apricot"
   : cal=48
elifclass_result[0]=="Watermelon":
   cal=30
elifclass_result[0]=="Nectarine":
   cal=44
elifclass_result[0]=="Cauliflower":
   cal=25
elifclass_result[0]=="Pomegranate":
   cal=64
elifclass_result[0]=="PearForelle":
   cal=100
elifclass_result[0]=="Papaya":
   cal=39
elifclass_result[0]=="PepperOrange"
   : cal=31
```

```
elifclass_result[0]=="Cherry1":
    cal=50
elifclass_result[0]=="TomatoCherryRed":
    cal=41
elifclass_result[0]=="OnionRed":
    cal=32
elifclass_result[0]=="Carambula":
    cal=41
elifclass_result[0]=="Beetroot":
    cal=60
elifclass_result[0]=="CactusFruit":
    cal=42
elifclass_result[0]=="NectarineFlat":
    cal=45
elifclass_result[0]=="Granadillla":
    cal=97
elifclass_result[0]=="NutForest":
    cal=514
elifclass_result[0]=="Kaki":
    cal=127
elifclass_result[0]=="OnionRedPeeled":
    cal=40
elifclass_result[0]=="OnionWhite":
    cal=40
elifclass_result[0]=="Kumquats":
    cal=71
elifclass_result[0]=="PotatoRed":
    cal=123
elifclass_result[0]=="PassionFruit":
    cal=97
elifclass_result[0]=="PotatoRedWashed":
    cal=151
elifclass_result[0]=="Avacadoripe":
    cal=250
elifclass_result[0]=="Quince":
    cal=57
elifclass_result[0]=="PotatoWhite":
    cal=67
elifclass_result[0]=="PotatoSweet":
    cal=86
elifclass_result[0]=="Clementine":
    cal=47
elifclass_result[0]=="AppleCrimsonSnow":
    cal=100
```

```
    elifclass_result[0]=="ApplePinkLady":
        cal=80
    elifclass_result[0]=="Cocos":


        cal=150
    elifclass_result[0]=="MangoRed":
        cal=70
    elifclass_result[0]=="Limes":
        cal=30
    elifclass_result[0]=="PepperRed":
        cal=70
    elifclass_result[0]=="PepperGreen"or"PepperYellow"or"PepperOrange":
        cal=20
    elifclass_result[0]=="AppleRedDelicious":
        cal=72
    elifclass_result[0]=="PomeloSweetie":
        cal=231
    elifclass_result[0]=="BananaLadyFinger":
        cal=140
    elifclass_result[0]=="Chestnut":
        cal=131
    elifclass_result[0]=="PearMonster":
        cal=121
    elifclass_result[0]=="Mangostan":
        cal=143
    elifclass_result[0]=="NutPecan":
        cal=690
    elifclass_result[0]=="GrapeWhite"or"GrapeWhite2"or"GrapeWhite3"or"Grape
White 4" :
        cal=69
    elifclass_result[0]=="PearKaiser":
        cal=100
    elifclass_result[0]=="TomatoYellow":
        cal=32
    elifclass_result[0]=="AppleRedYellow1"or"AppleRedYellow2":
        cal=252
    elifclass_result[0]=="GrapefruitPink":
        cal=52
    elifclass_result[0]=="Tangelo":
        cal=47
    elifclass_result[0]=="GrapeBlue":
        cal=30
    elifclass_result[0]=="Huckleberry":
        cal=37
```

```
    elifclass_result[0]=="Raspberry":
        cal=53
    elifclass_result[0]=="Plum3"or"Plum2":
        cal=90
    elifclass_result[0]=="Dates":
        cal=282


    elifclass_result[0]=="Maracuja":
        cal=97
    elifclass_result[0]=="TomatoMaroon":
        cal=95.1
    elifclass_result[0]=="CherryWaxYellow"or"CherryWaxBlack"or"CherryWax
Red":
        cal=87
    elifclass_result[0]=="Salak":
        cal=82
    elifclass_result[0]=="Hazelnut":
        cal=628
    elifclass_result[0]=="Tamarillo":
        cal=30
    elifclass_result[0]=="Walnut":
        cal=654
    elifclass_result[0]=="Guava":
        cal=68
    elifclass_result[0]=="LemonMeyer":
        cal=20
    elifclass_result[0]=="Mulberry":
        cal=43
    elifclass_result[0]=="BananaRed":
        cal=90
    elifclass_result[0]=="Redcurrant":
        cal=68
    elifclass_result[0]=="PineappleMini":
        cal=189
    elifclass_result[0]=="Pepino":
        cal=46
    elifclass_result[0]=="Physalis":
        cal=53
    elifclass_result[0]=="PitahayaRed":
        cal=60
    elifclass_result[0]=="PhysaliswithHusk":
        cal=53
    elifclass_result[0]=="PearAbate":
        cal=40
```

```
    elifclass_result[0]=="Rambutan":
        cal=75
    elifclass_result[0]=="PearWilliams":
        cal=54
    elifclass_result[0]=="MelonPieldeSapo":
        cal=34
    elifclass_result[0]=="Cantaloupe1"or"Cantaloupe2":
        cal=60
    elifclass_result[0]=="CherryRainier":
        cal=90
    elifclass_result[0]=="Cherry2":
        cal=63
    elifclass_result[0]=="StrawberryWedge":
        cal=32
    else:
        cal=0
    for i in range(1):
         cal_result.append(cal)
    returnclass_result,prob_result,cal_result
@app.route('/')
def home():
        return render_template("index.html")
@app.route('/success',methods=['GET','POST'])
def success():
    error = ''
    target_img=os.path.join(os.getcwd(),'static/images')
    if request.method == 'POST':
        if(request.form):
            link=request.form.get('link')
            try :
                resource=urllib.request.urlopen(link)
                unique_filename = str(uuid.uuid4())
                filename = unique_filename+".jpg"
                img_path=os.path.join(target_img,filename)
                output = open(img_path , "wb")
                output.write(resource.read())
                output.close()
                img=filename
                class_result,prob_result,cal_result=predict(img_path,model)
                predictions = {
                     "class1":class_result[0],
                     "prob1":
                     prob_result[0],
                     "cal1":cal_result[0],
                }
```

```
            exceptExceptionase:
                print(str(e))
                error='Thisimagefromthissiteisnotaccesibleorinappropriateinput'
            if(len(error) == 0):
                returnrender_template('success.html',img=img,predictions=predictions)
            else:
                returnrender_template('index.html',error=error)
        elif (request.files):
            file = request.files['file']
            if file and allowed_file(file.filename):
                file.save(os.path.join(target_img,file.filename)
                )

                img_path=os.path.join(target_img,file.filename)
                img = file.filename
                class_result,prob_result,cal_result=predict(img_path,model)
                predictions = {
                     "class1":class_result[0],
                      "prob1":prob_result[0],
                      "cal1":cal_result[0],
                 }

            else:
                error="Pleaseuploadimagesofjpg,jpegandpngextensiononly"
            if(len(error) == 0):
                returnrender_template('success.html',img=img,predictions=predictions)
            else:
                returnrender_template('index.html',error=error)
    else:
        returnrender_template('index.html'
) if name___== "main ":_
    app.run(debug=True)
```

templates->index.html

```
<!DOCTYPEhtml>
<html lang="eng">
   <head>
      <metacharset="UTF-8">
      <metaname="viewport"content="width=device-width,initial-scale=1.0">
      <linkrel="stylesheet"href={{url_for('static',filename='css/normalize.css')}}>
      <linkrel="stylesheet"href={{url_for('static',filename='css/grid.css')}}>
      <linkrel='stylesheet'href={{url_for('static',filename='css/style.css')}}>
       <link
rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
       <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
      <link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,100;0,300;0,400;1,3
00&display=swap" rel="stylesheet" type='text/css'>
      <title>ImageClassificationWebApp</title>
      <metaname="keywords"content="cutomfileinput,styling,label,cross-browser,
accessible, input type file" />
         <style>
               .content{
                 max-width:500px;
                 margin: auto;

               }.bg-img{
                     width: 100%;
                     height:100vh;
                     background-image:linear-gradient(rgba(0,0,0,0.71),rgba(0,0,0,
0.7)),url('static/css/front1.jpg');
                     background-size: cover;
                     background-
                     position:center;
               }

         </style>
```

```
   </head>
   <body>
<divclass="bg-img">
<divclass="content">
      <divclass="index-main">
      <divstyle="width:100%;padding-top:5%;">
         <divclass="header-content">
            <h1style="text-align:center;color:white"><spanclass="header-content-
text">Fruit Detection Model</span></h1>
         </div>
      </div>
      <divclass="upload-section">
         <divclass="upload-file">
            <formclass="file-form"action="/success"method="post",
enctype="multipart/form-data">
               <inputclass="file-form-input"type="file",name="file"/>
               <buttonclass="btnbtn-successbtn-lg">Upload</button>
            </form>
         </div>
         <div class="uploadOR">
            <span>OR</span>
         </div>
         <divclass="upload-link">
            <formclass="link-form"action="/success"method="post",
enctype="multipart/form-data">
               <inputclass="link-form-input"type="text",maxlength="1000",name=
"link" placeholder="Paste the image URL"/>
               <buttonclass="btnbtn-successbtn-lg">Proceed</button>
            </form>
         </div>
         <divclass="header-content-sub">
               <pstyle="text-align:center;margin-top:15px;color:red"class="header-
content-info">{{error}}</p>
         </div>
      </div>
      <divclass= "footer">
         <divclass="header-content-sub">

               <pstyle="text-align:center;color:white"class="header-content-info"></p>
         </div>
         <divclass="header-content-sub">
               <pstyle="text-align:center;color:orange"class="header-content-
info"></p>
         </div>
```

```
      </div>
    </div>
</div>
</div>
   </body>
</html>
```

Templates-> success.html

```
<!DOCTYPEhtml>
<html lang="eng">
   <head>
      <metacharset="UTF-8">
      <metaname="viewport"content="width=device-width,initial-scale=1.0">
      <linkrel="stylesheet"href={{url_for('static',filename='css/normalize.css')}}>
      <linkrel="stylesheet"href={{url_for('static',filename='css/grid.css')}}>
      <linkrel='stylesheet'href={{url_for('static',filename='css/styleSucc.css')}}>
      <link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,100;0,300;0,400;1,3
00&display=swap" rel="stylesheet" type='text/css'>
      <link
rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.
min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6
JX m" crossorigin="anonymous">
      <title>ResultsPage</title>
      <styletype="text/css">th
         {
            text-align:center;
            font-size: 20px;
         }

         td{
            text-align:center;
            font-size: 18px;
         }

      </style>
   </head>
   <body>
      <divclass="second-main"style="height:100%,width:100%;">
         <nav>

            <ulstyle="padding-right:2%;"class="main-nav">
               <li><ahref="/">HomePage</a></li>
            </ul>
```

```
        </nav>
        <divclass= "header">
          <rowstyle="width:100%;display:flex;justify-content: center;">
            <h3class="header-text">UploadedImage</h3>
          </row>
          <rowstyle="width:100%;display:flex;justify-content: center;">
            <imgclass="result-img"src={{url_for('static',filename='images/'+
img)}}>
          </row>
        </div>
        <divclass= "info">
          <rowstyle="width:100%;display:flex;justify-content: center;">
            <h3class="header-text">ModelPrediction</h3>
          </row>
          <rowstyle="width:100%;display:flex;justify-content: center;">
            <tableclass="table-borderedtext-lighttable-custom">
              <tr>
                <th>Rank</th>
                <th>Class</th>
                <th>Probability</th>
                <th>Calories</th>
              </tr>
              <tr>
                <td>1st</td>
                <td>{{predictions.class1}}</td>
                <td>{{predictions.prob1}}%</td>
                <td>{{predictions.cal1}}</td>
               </tr>
            </table>
          </row>
        </div>
      </div>
    </body>
</html>
```