CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Agriculture is an important part of the Indian economy, producing a broad variety of fruits, vegetables, and grains for both internal and export consumption. In terms of fruit output, India is ranked second in the world. When choosing a fruit to buy the smell, shape, color and ripeness are all significant elements to consider.

Traditional methods for sorting of fruits were to manually handpick each fruit and segregate them into good and rotten fruits, however this heavily time consuming, laborious and expensive method.

To solve this crisis, a transfer learning based MobilenetV1 image classification method has been developed so that it can be implemented in industries that deal with a large number of fruits on a daily basis, such that it can be implemented in conveyer belts so that it can automatically scan the fruit and segregate them as fresh and rotten by taking into account shape, color, texture, spots, fungal growth and rotting while simultaneously predicting their shelf-life.

1.2 IMAGE CLASSIFICATION

Image classification is where a computer can analyze an image and identify the ‘class’ the image falls under. (Or a probability of the image being part of a ‘class’.) A class is essentially a label, for instance, ‘car’, ‘animal’, ‘building’ and so on. Early image classification relied on raw pixel data. This meant that computers would break down images into individual pixels. The problem is that two pictures of the same thing can look very different. They can have different backgrounds, angles, poses, etcetera. This made it quite the challenge for computers to correctly ‘see’ and categorize images. There are two types of classification: supervised and unsupervised.

1.2.1 SUPERVISED CLASSIFICATION

Supervised classification is based on the idea that a user can select sample pixels in an image that are representative of specific classes and then direct the image processing software to use these training sites as references for the classification of all other pixels in the image.

Some examples are K-Nearest Neighbors Classifier and Maximum likelihood classifier.

1.2.2 UNSUPERVISED CLASSSIFICATION

Unsupervised classification is where the groupings of pixels with common characteristics are based on software analysis of an image without the user defining training fields for each land cover class. All this is done without the help of training data or prior knowledge.

The image analyst's responsibility is to determine the correspondences between the spectral classes that the algorithm defines. In unsupervised classification, there are two basic steps to follow. These include generate clusters and assigning classes. Using the remote sensing software, an analyst will first create clusters and identify the number of groups to generate. After this, they assign land cover classes to each cluster. E.g.: K-means, ISODATA.

1.3 DEEP LEARNING

Deep learning is a type of machine learning; a subset of artificial intelligence (AI) that allows machines to learn from data. Deep learning involves the use of computer systems known as neural networks. In neural networks, the input filters through hidden layers of nodes. These nodes each process the input and communicate their results to the next layer of nodes. This repeats until it reaches an output layer, and the machine provides its answer. There are different types of neural networks based on how the hidden layers work. Image classification with deep learning most often involves convolutional neural networks, or CNN's. In CNN's, the nodes in the hidden layers don't always share their output with every node in the next layer (known as convolutional layers).

Deep learning allows machines to identify and extract features from images. This means they can learn the features to look for in images by analyzing lots of pictures. So, programmers don't need to enter these filters by hand.

1.4 TRANSFER LEARNING

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

Two common approaches are as follows:

1. Develop Model Approach
2. Pre-trained Model Approach

For the Pre-trained model approach:

a. **Select Source Model**. A pre-trained source model is chosen from available models. Many research institutions release models on large and challenging datasets that may be included in the pool of candidate models from which to choose from.
b. **Reuse Model**. The model pre-trained model can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used.
c. **Tune Model**. Optionally, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

1.5 MOBILENETV1

MobileNet is a type of convolutional neural network designed for mobile and embedded vision applications. They are based on a streamlined architecture that uses depth wise separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices. MobileNets are built primarily from depth wise separable convolutions and subsequently used in Inception models to reduce the computation in the first few layers. Flattened networks build a network out of fully factorized convolutions and showed the potential of extremely factorized networks. For MobileNets the depth wise convolution applies a single filter to each input channel.

The point wise convolution then applies a 1×1 convolution to combine the outputs the depth wise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depth wise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.[2]

MobileNet uses 3 × 3 depth wise separable convolutions which uses between 8 to 9 times less computation than standard convolutions at only a small reduction in accuracy.

The mobilenet Architecture is presented below:

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig 1.1: MobilenetV1 architecture (Tabular form)

1.6 CONFUSION MATRIX

A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

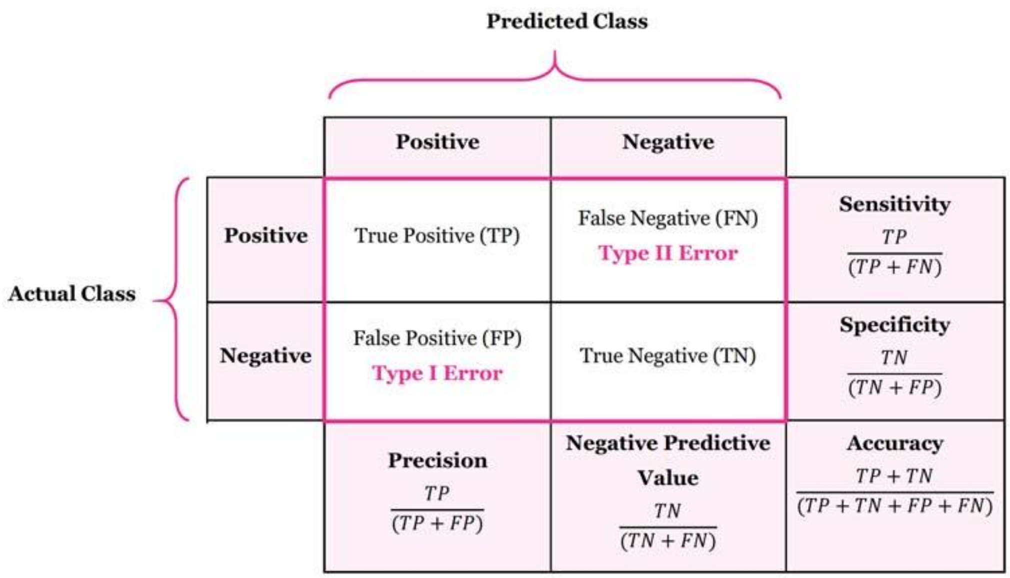


Fig 1.2: Confusion Matrix

1.7 KERAS

Keras is a deep learning API written in Python, running on top of the machine learning platform Tensor Flow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is:

- **Simple** -- but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.
- **Flexible** -- Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.

1.8 REACT JS:

React JS is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

React makes it painless to create interactive UIs. Design simple views for each state in your application and react will efficiently update and render just the right components when your data changes.

Since Web browsers understand JavaScript, we can use React to describe Web User Interfaces. I like to use the word describe here because that's what we basically do with React, we just tell it what we want and react will build the actual User Interfaces, on our behalf, in the Web browser. Without React or similar libraries, we would need to manually build User Interfaces with native Web APIs and JavaScript. Declarative views make your code more predictable and easier to debug.

1.9 MATERIALS UI:

Material-UI is simply a library that allows us to import and use different components to create a user interface in our React applications. This saves a significant amount of time since the developers do not need to write everything from scratch.

MUI offers a comprehensive suite of UI tools to help you ship new features faster. Start with Material UI, our fully loaded component library, or bring your own design system to our production-ready components.

Material-UI widgets are heavily inspired by Google's principles on building user interfaces. It is, therefore, easy for developers to build visually appealing applications. You can learn more about Google's material design principles from here.
To incorporate the Material-UI library and use its components in a React.js application.

1.10 PYTHON:

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects [magic methods]). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It uses dynamic name resolution (late binding), which binds method and variable names during program execution.

Its design offers some support for functional programming in the Lisp tradition. It has filter, mapandreduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Its core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Readability counts.

Rather than building all of its functionality into its core, Python was designed to be highly extensible via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, wrote: "To describe something as 'clever' is not considered a compliment in the Python culture."

1.11 CASCADING STYLE SHEETS:

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

1.12 MACHINE LEARNING

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products.

There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

- Supervised learning: In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.
- Unsupervised learning: This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
- Semi-supervised learning: This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.
- Reinforcement learning: Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

1.13 HYPER PARAMETER TUNING

Fine-tuning deep learning algorithms will help to improve the accuracy of a new neural network model by integrating data from an existing neural network and using it as an initialization point to make the training process time and resource efficient.

Although fine-tuning proves beneficial in training new deep learning algorithms, it can be used only when the dataset of an existing model and the new deep learning model are similar to each other.

Fine-tuning takes a model that has already been trained for a particular task and then fine-tuning or tweaking it to make it perform a second similar task. For example, a deep learning network that has been used to recognize cars can be fine-tuned to recognize trucks.

Since the input information for the new neural network is similar to a pre-existing deep learning model, it becomes a relatively easy task to program the new model.

1. The first step includes importing the data of the existing similar deep learning network.
2. The second step involves removing the output layer of the network as it was programmed for tasks specific to the previous model.
3. The third step is optional and depends upon the similarities of both the learning models. You may require adding or remove certain layers depending upon the similarities of the two models. Once you've added or removed layers depending upon the data required, you must then freeze the layers in the new model.
4. Freezing a layer means the layer doesn't need any modification to the data contained in them, henceforth. The weights for these layers don't update when we train the new model on the new data for the new task.
5. The final step involves training the model on the new data.

1.14 PROBLEM STATEMENT

1. Customers are stringent with the fruits they pick in supermarkets. Without proper management a large number of fruits can go to waste for various reasons such as shape, marks, rotten spots, bacterial growth, over-ripe.

2. Shelf-life also plays a huge role when it comes to the exports of fruits.

    a. Fruits with longer shelf-life can be transported over long distances to farther states in India or abroad. These fruits with longer shelf-life can also be put into cold storage where they can be stored for some period of time before being transported/ exported without the fear of it rotting.

b. Whereas fruits with shorter shelf-life can be transported to local marts and vendors where they can be sold to the customers when they are at their prime ripeness.

3. By creating a system that can take care of these problems to reduce the overall food wastage while simultaneously catering to the customers and vendors need will be beneficial to all.

1.15 EXSISTING SYSTEM

One of the most prevalent methods being used in the frit and vegetable industry for the sorting is manual inspection. While this has been an effective method in the past, with the boom in population over the last 10 years and increase in food demand, manual inspection has not been able to keep up with the demand. Manual inspection is not only a heavily time-consuming and laborious method but also gives a huge room for human error.

The other image-classification involves converting the image pixels into a greyscale which may not be able to take into the factor the color of the fruit for the ripeness factor.

They also are based on heavy models of Convolutional Neural Network (CNN). Although CNN are great for image classification, they require a huge amount of data to train, and a computer with high storage and processing powers and hence cannot be deployed on devices that do not support/meet these requirements.

1.16 PROPOSED SYSTEM

To solve the above problems in the existing systems, a classification model is built using Transfer learning model MobilnetV1 to which hyper parameter tuning is performed in order to improve the efficiency and accuracy of the system.

Shelf-life feature has also been added to the model which predicts the shelf-life based on the color of the fruits for fruits that show color change as they ripen.

1.17 OBJECTIVES

- To build an image classification model that can classify fresh and rotten fruits with an excellent accuracy based on the image.
- To also integrate a shelf-life model that can predict the shelf life of the fruit based on the color of the fruit for fruits that show a color difference as they ripen.
- The prime objective is to build a model that can classify fruits as fresh or rotten with an excellent accuracy and can be deployed on systems to be used in warehouses and industries that deal with a large number of fruits on a daily basis.

1.18 SCOPE OF PROJECT

Make use of modern Machine Learning and Image classification models, to predict the shelf-life and classify fruits as fresh and rotten accurately. This accuracy will help prevent food wastage and maximize profit for the farmers and vendors.

1.19 ORGANIZATION OF REPORT

The remaining of the report is organized as below: -

Chapter 2: Literature survey on: Real-time visual inspection system for grading fruits, feature based fruit quality grading system, Maturity status classification of papaya fruits, Ripeness classification of bananas, Hep-2 cell image classification, automatic fruit classification, Fruit classification by extracting color chromaticity, shape and texture features and Real-time grading method of apples based on features extracted from defects.

Chapter 3: System Requirement Specification (Hardware Requirements, Software

Requirements,Functional Requirements and Non-Functional Requirements).

Chapter 4: Gives the complete system analysis and design. It explains the high-level design of the project, the block diagram as well as the low-level design.

Chapter 5: Gives description of all the technologies used for this project: REACT JS, CSS and Materials UI, Python.

Chapter 6: Various tests performed on the project.

Chapter 7: Conclusion and future enhancements that could be applied to the project.

CHAPTER 2

LITERATURE REVIEW

Deep learning-based low-cost machine vision system for grading the fruits based on their outer appearance or freshness. Various state-of-the-art deep learning models and stacking ensemble deep learning methods were applied to two data sets of fruits.

2.1. REAL-TIME VISUAL INSPECTION SYSTEM FOR GRADING FRUITS USING COMPUTER VISION AND DEEP LEARNING TECHNIQUES:

A framework for learning and classifying bananas is developed first. It uses neural network technology to detect the fruit's ripening stage. Due to the complexity of the banana fruit's ripening stages, it is necessary to develop image processing tools that can identify the various fresh incoming bunches. The goal is to create an image processing system that can detect the different stages of the fruit's ripening process. This method would help determine the optimal eating quality and the price of bananas.

The model was tested, trained and compared the performance of different deep learning models including DenseNet, ResNet, NASNet MobileNetV2and Efficient Net to find out which one is the best model for the grading of fruits. The model provides a real time visual inspection using a low-cost Raspberry Pi module with a camera and a touch screen display for user interaction.

2.2. FEATURE BASED FRUIT QUALITY GRADING SYSTEM USING SUPPORT VECTOR MACHINE:

Computer vision is a widely used technique for processing images. In this paper, it describes about the study the various aspects of machine learning for the classification of fruits and vegetables. Through a variety of data sources, we found that SVM achieves better accuracy than other machine learning techniques. We perform the Recognition and classification of fruits and vegetables and detection of disease in fruits and vegetables among the horticulture products under the agriculture field using computer vision.

The model is divided into three modules image pre-processing, image segmentation and image classifier. First phase describes about extracting the image features such as image quality, area, perimeter, mean, variance, color, intensity. Second phase describes about k-means clustering is used for image segmentation. In last phase the vectors of color, intensity, segments, and image quality feature were utilized for training of the support vector machine structure.

2.3. MATURITY STATUS CLASSIFICATION OF PAPAYA FRUITS BASED ON MACHINE LEARNING AND TRANSFER LEARNING APPROACH:

This paper proposed a classification model for maturity status classification of papaya fruits in two approaches, machine learning and transfer learning approach. Overall, the VGG19 is better as VGG19 is based on transfer learning, there is no requirement of feature extraction and feature selection process. Although the transfer learning approach needs complex architecture, high training time and large data sets it is one time only. However, the achieved accuracy in both machine learning and transfer learning is 100% and beat the previous method 94.7% of accuracy.

The experimentation is done with 300 papaya fruit sample images with 100 of each three maturity stages. The machine learning approach includes three sets of features and three classifiers with their different kernel functions. The features and classifiers used in machine learning approaches are local binary pattern (LBP), histogram of oriented gradients (HOG), Gray Level Co-occurrence Matrix (GLCM) and k-nearest neighbor (KNN), support vector machine (SVM), Naïve Bayes respectively.

2.4 RIPENESS CLASSIFICATION OF BANANAS USING AN ARTIFICIAL NEURAL NETWORK:

A deep learning-based framework for fruit classification was proposed in this work. The proposed paper tells us they have created a fuzzy model to check whether the fruit banana is ripe, unripe or overripe. For this they have used Regression Tree Algorithm and also the classification method. This process is evaluating the banana on different ripening stages on the MUSA database.

Artificial neural network-based framework which uses color, development of brown spots, and Tamura statistical texture features is employed to classify and grade banana fruit ripening stage. Results and the performance of the proposed system are compared with various techniques such as the SVM, the naive Bayes, the KNN, the decision tree, and discriminate analysis classifiers.

2.5. HEP-2 CELL IMAGE CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS:

The proposed paper gives the information that Convolution Neural Network has feature extraction which can be used for object recognition, semantic segmentation and image super-resolution. For object recognition we can use CNN architecture such as AlexNet, VGG16 and VGG19.

For hybrid classification we will be using the CNN architecture along with support vector machine classifier. VGG19 CNN Architecture is best comparing with the others. Using deep convolutional neural network, the cell image classification is done.

2.6. AUTOMATIC FRUIT CLASSIFICATION USING DEEP LEARNING FOR INDUSTRIAL APPLICATIONS:

Deep Learning is machine learning technique which depends on supervised, semi supervised and unsupervised learning. For image processing, computer vision and pattern recognition. In working part first, it will be working on Deep Neural Network in Deep Learning area and then convolution neural network and they are LeNet, AlexNet, GoogleNet, VGG16, VGG19, Resnet50 etc.

The framework is based on two different deep learning architectures. The first is a proposed light model of six convolutional neural network layers, whereas the second is a fine-tuned visual geometry group-16 pretrained deep learning model. Two color image datasets, one of which is publicly available, are used to evaluate the proposed framework.

2.7. FRUIT CLASSIFICATION BY EXTRACTING COLOR CHROMATICITY, SHAPE AND TEXTURE FEATURES: TOWARDS AN APPLICATION FOR SUPERMARKETS:

In this paper, it has been discussed there may arise a human error while checking the quality of the fruit so using image acquisition and image classification for checking the fruit quality and three important factors used are image segmentation, image pre-processing, Classifier. In addition to this k-means clustering is used to achieve better improvement of accuracy and speed of the classifier.

The purpose to use the HSV (Hue, Saturation, Value) space because it is possible to extract and process the chromaticity data, but without the undesirable intensity effects of the RGB space. On the other hand, before the color is characterized, we complete a selection of the chromaticity's that contribute with important data about the fruits.

2.8. A REAL-TIME GRADING METHOD OF APPLES BASED ON FEATURES EXTRACTED FROM DEFECTS:

In this proposed paper, it is discussed about apple defect detection where first the apple image is captured and then background is deleted, and the algorithm used in this paper is Fuzzy C-means Algorithm and the Nonlinear Programming Genetic Algorithm (FCM-NPGA) and for multivariate image analysis. Using this algorithm, the image of apple that is apple is examined every angle so that the defect can be found easily.

The classification probabilities of the objects were summarized and on this basis the fruits were graded using quadratic discriminant analysis. The fruits were correctly graded with a rate of 73%

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Software Requirements

- Operating system: Windows 10
- Coding Language: Python, CSS, js.
- Framework: ReactJS
- Version: Python 3.6.8
- IDE: Python 3.6.8 IDLE
- ML Packages: NumPy, Pandas, Sklearn, Matplotlib, Seaborn, Keras, MobileNetV1
- ML Algorithms: MobilenetV1

3.2 Hardware Requirements

Processor : Minimum - 1.9 gigahertz (GHz) x86- or x64-bit dual core processor.

Recommended - 3.3 gigahertz (GHz) or faster 64-bit dual core processor.

GPU – 2GB NVIDIA

Memory : Minimum - 4GB RAM

Recommended - 8GB RAM or more

Graphic card : Minimum of 2GB NVIDIA Graphics card

3.3 Functional Requirements:

- A function of software system is defined in functional requirement.
- The behavior of the system is evaluated when it is presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.

3.4 Non-Functional Requirements:

- **Reliability:** Reliability is an attribute of any computer-related component (software, or hardware, or a network, for example) that consistently performs according to its specifications. It has long been considered one of three related attributes that must be considered when making, buying, or using a computer product or component. Reliability, availability, and serviceability - RAS, for short - are considered to be important aspects to design into any system. In theory, a reliable product is totally free of technical errors; in practice, however, vendors frequently express a product's reliability quotient as a percentage. Evolutionary products (those that have evolved through numerous versions over a significant period of time) are usually considered to become increasingly reliable, since it is assumed that bug s have been eliminated in earlier releases.

- **Performance:** Computer performance is the efficiency of a given computer system, or how well the computer performs, when taking all aspects into account. A computer performance evaluation is defined as the process by which a computer system's resources and outputs are assessed to determine whether the system is performing at an optimal level. It is similar to a voltmeter that a handyman may use to check the voltage across a circuit. The meter verifies that the correct voltage is passing through the circuit. Similarly, an assessment can be done on a PC using established benchmarks to see if it is performing correctly.

- **Portability:** Portability, in relation to software, is a measure of how easily an application can be transferred from one computer environment to another. A computer software application

- is considered portable to a new environment if the effort required to adapt it to the new environment is within reasonable limits. The meaning of the abstract term 'reasonable' depends upon the nature of the application and is often difficult to express in quantifiable units. The phrase "to port" means to modify software and make it adaptable to work on a different computer system.

  For example, to port an application to Linux means to modify the program so that it can be run in a Linux environment. Portability refers to the ability of an application to move across environments, not just across platforms.

- **Scalability:** Scalability is the ability for IT systems – such as applications, storage, databases and networking – to continue to function properly when changed in size or volume. It often refers to increasing or decreasing resources as needed to meet the higher or lower demands of a business. Vertical (scale-up) scalability increases the capacity of hardware or software by adding resources to a physical system, such as adding processing power to a server to make it faster. For scale-up storage, this means adding more devices, such as disk drives, to an existing system when more capacity is required. Horizontal (scale-out) scalability connects multiple items in order to work as a single logical unit. For scale-out storage, this means adding devices in connected arrays or clusters. Each cluster can have many nodes (devices), and nodes can be separated geographically. Scale-out NAS (network-attached storage) grows by adding clustered nodes. Because each node includes storage capacity, processing power and I/O (input/output) bandwidth, performance increases along with storage capacity.

- **Flexibility:** In the computer world, "flexible" may refer to hardware, software, or a combination of the two. It describes a device or program that can be used for multiple purposes, rather than a single function.

- **Security:** Computer security deals with the protection of computer systems and information from harm, theft, and unauthorized use. The main reason users get attacked frequently is that they lack adequate defenses to keep out intruders, and cybercriminals are quick to exploit such weaknesses. Computer security ensures the confidentiality, integrity, and availability of your computers and stored data.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

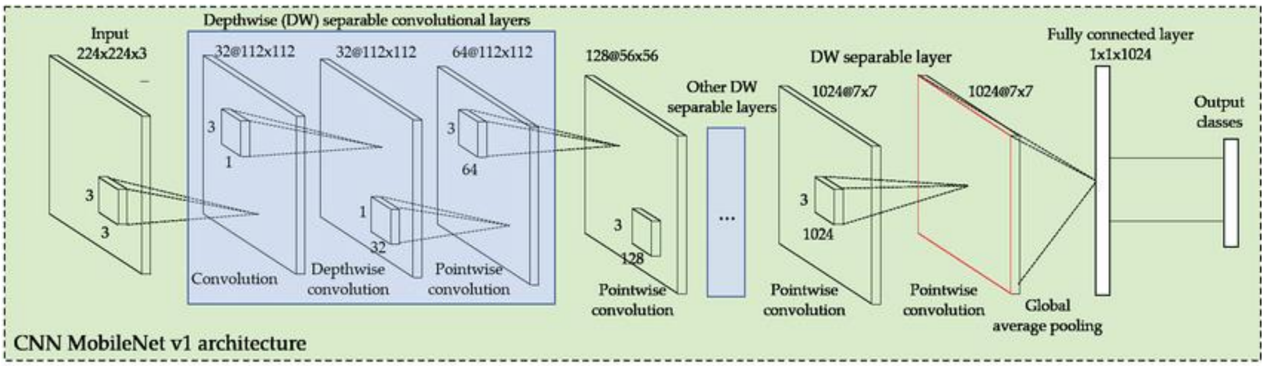


Fig 4.1: MobileNet v1 architecture

MobileNetV1 is an efficient CNN architecture that use the depth wise separable convolutions, which factorize a standard convolution into a depth wise convolution and a point wise convolution, in order to efficiently build lighter models with respect to earlier architectures. Moreover, MobileNetV1 introduces two global hyper-parameters, which allow performing a trade-off between latency and accuracy, namely the width multiplier and resolution multiplier. Therefore, the MobileNetV1 is built with multiple depth wise separable convolution layers and each depth wise separable convolution layer consists of a depth wise convolution and a point wise convolution. The MobileNetV1 has 28 layers by counting depth wise and point wise convolution as separate layers. The size of the input images is 224 X 224 X 3 pixels; thus, the images are properly resized before feeding them into the proposed CNN.

Width multiplier (denoted by α) is a global hyper parameter that is used to construct smaller and less computationally expensive models. Its value lies between 0 and 1. For a given layer and value of α, the number of input channels 'M' becomes α * M and the number of output channels 'N' becomes α * N hence reducing the cost of computation and size of the model at the cost of performance. The computation cost and number of parameters decrease roughly by a factor of α2. Some commonly used values of α are 1,0.75,0.5,0.25.

The second parameter introduced in MobileNets is called resolution multiplier and is denoted by ρ. This hyper parameter is used to decrease the resolution of the input image and this subsequently reduces the input to every layer by the same factor. For a given value of ρ the resolution of the input image becomes 224 * ρ. This reduces the computational cost by a factor of ρ2.

4.2 BLOCK DIAGRAM:

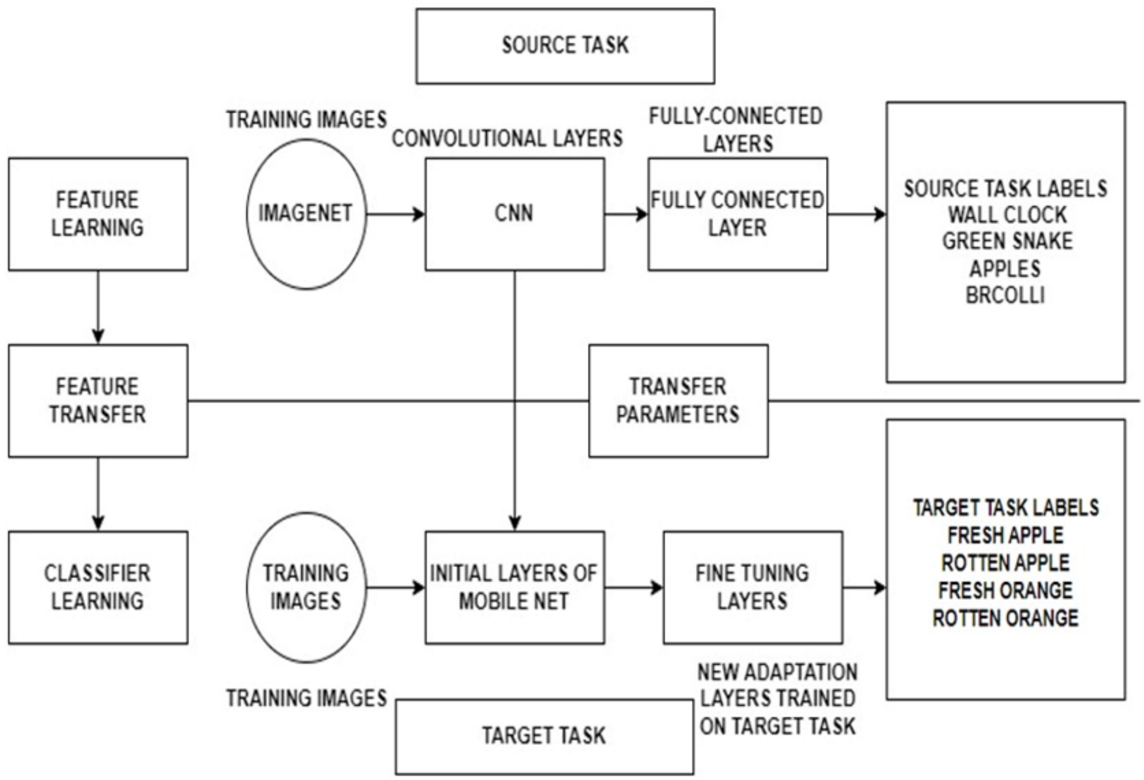


Fig 4.2: Block diagram of Transfer Learning

The TL model is initially trained on the ImageNet database, where it is taught to identify 1000 images using various layers of convolutional neural networks and a fully connected layer. These trained models can subsequently be used for our unique implementations.

The weights from the previously trained dataset are transferred to the early layers of mobile net, while the final few layers can be fine-tuned/modified, and the model re-trained on the unique target task. The dataset being used consists of 14,334 images that have been obtained from Kaggle. They have been split into train, test and validation data. The 6 classes fresh apples, fresh bananas, fresh oranges, rotten apples, rotten bananas and rotten oranges. This fine-tuned model undergoes testing with the testing dataset. The trained model is then validated and tested against the test data which then goes on to make predictions.

4.3 FLOW CHART OF TRANSFER LEARNING:

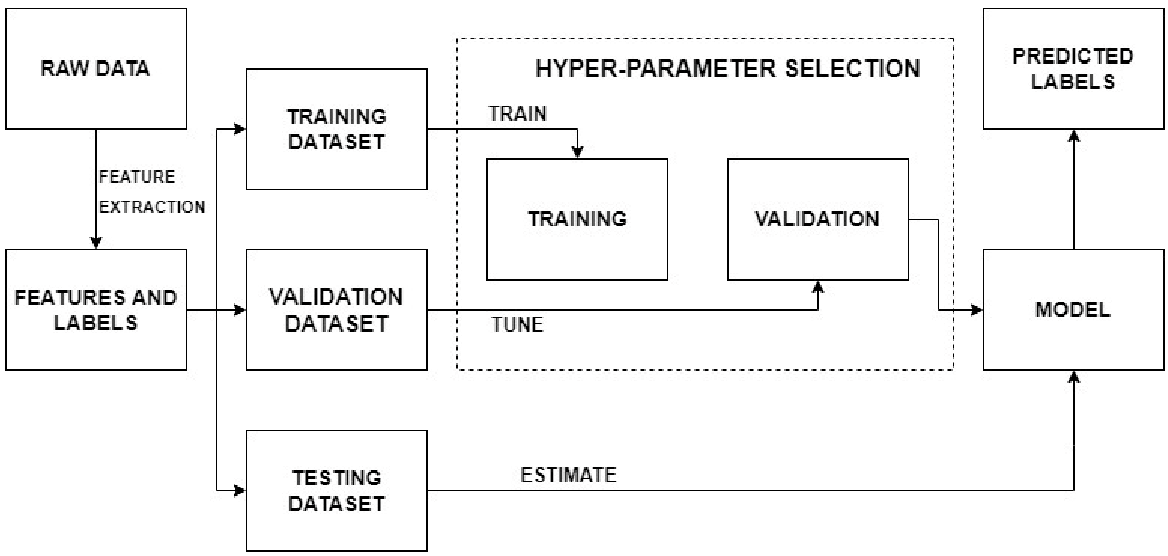


Fig 4.3: Flowchart of Transfer Learning

The training data is pre-processed with the use of a mobilenet preprocessing function for transfer learning, which includes feature extraction and resizing. Following that, we fine-tune the mobile net to assure better performance with our dataset. This is given to the model as training input.

The training is then checked against the validation set for accuracy and loss, as well as to verify if our model is over fitting or underfit. Once the prediction model is complete, we pass our test set to the model, which then produces a classification of fruit quality.

4.4 USE CASE DIAGRAM:

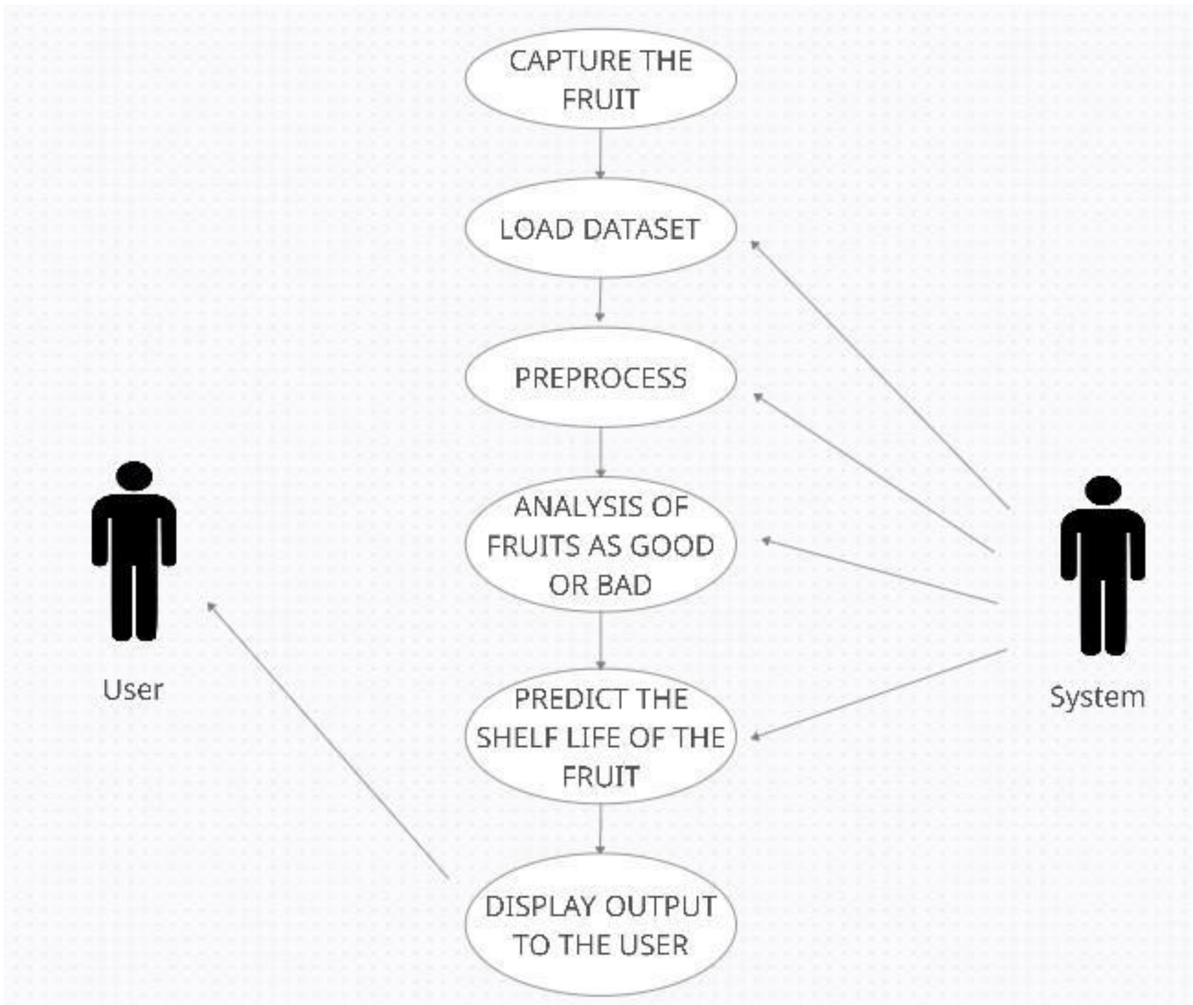


Fig 4.4: Use case diagram

Fig 4.4 shows the Use-case diagram. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. The user and the system interaction is represented in the use case diagram. The image of the fruit is captured, and the dataset is being uploaded. The system interacts with the classed where the dataset is uploaded, and the processing starts. The data is preprocessed where actions such as data cleaning, transformation, feature extraction, etc. is performed. After this the System predicts the quality of the fruit and predicts if it is fresh or rotten, along with this the system also determines the shelf life of the give fruit. This data is then later made available to the user, where they can view the quality and the shelf life of the given fruit.

4.5 ALGORITHM:

1. First, preprocessing of the dataset is done by constructing an input pipeline, in this instance Keras Image Data Generator, and running the preprocessing function, which turns the data into a format that the mobilenet understands and normalizes the RGB pixels to a range of -1 to 1. The dimensions of the input photos (224,224) are specified.
2. Further, perform modification to the fully connected layer and add a dense layer classify the 6 different classes of our dataset using the softmax activation function as it is multi-classification.
3. Next fine-tune the model by freezing all but the final 5 layers and re-training it with a low learning rate on our dataset.
4. Lastly compiling using the adam optimizer and the categorical cross entropy loss function is done.
5. Finally, training the model for 100 epochs, a training accuracy of 99.99 and validation accuracy of 95.42, with a training and validation loss of 1.375 and 0.2758 was achieved respectively.

Our platform was designed and developed using various technology stack in order to provide good accuracy in detecting the quality and shelf life of the fruits and vegetables.

CHAPTER 5

IMPLEMENTATION

5.1 FLOWCHART:

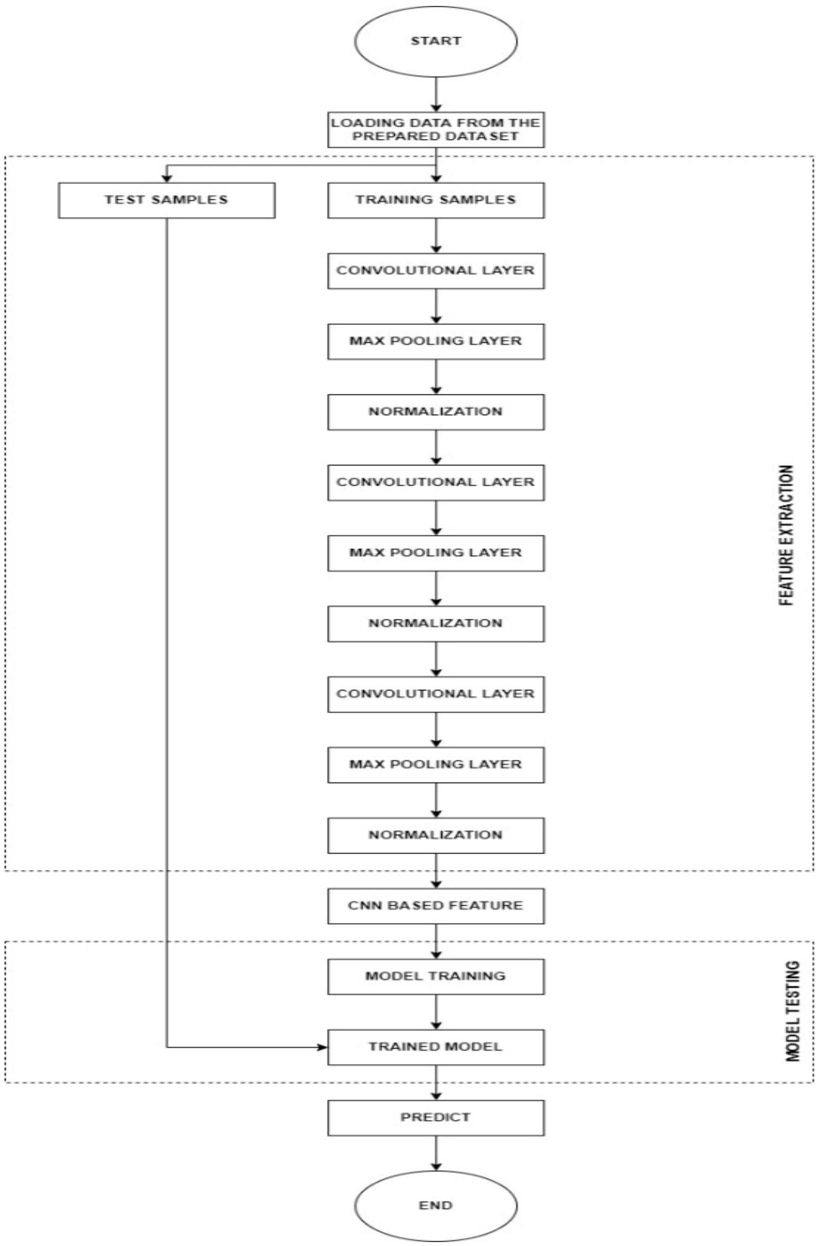


FIG 5.1 FLOWCHART

Fig 5.1. shows the flowchart of working model. First, the raw data, which are the images of the fruits are collected and fed into the pre-trained model which is trained with more datasets to get greater accuracy. After feature extraction process the dataset is divided into training dataset, validation dataset and testing dataset. The training dataset is sent for training and the validation datasets are sent for fine tuning the model.

Once this process is done the training results are tested are tested with the testing datasets. Hence at the final stage the model gives out the predictions.

5.2 BLOCK DIAGRAM:

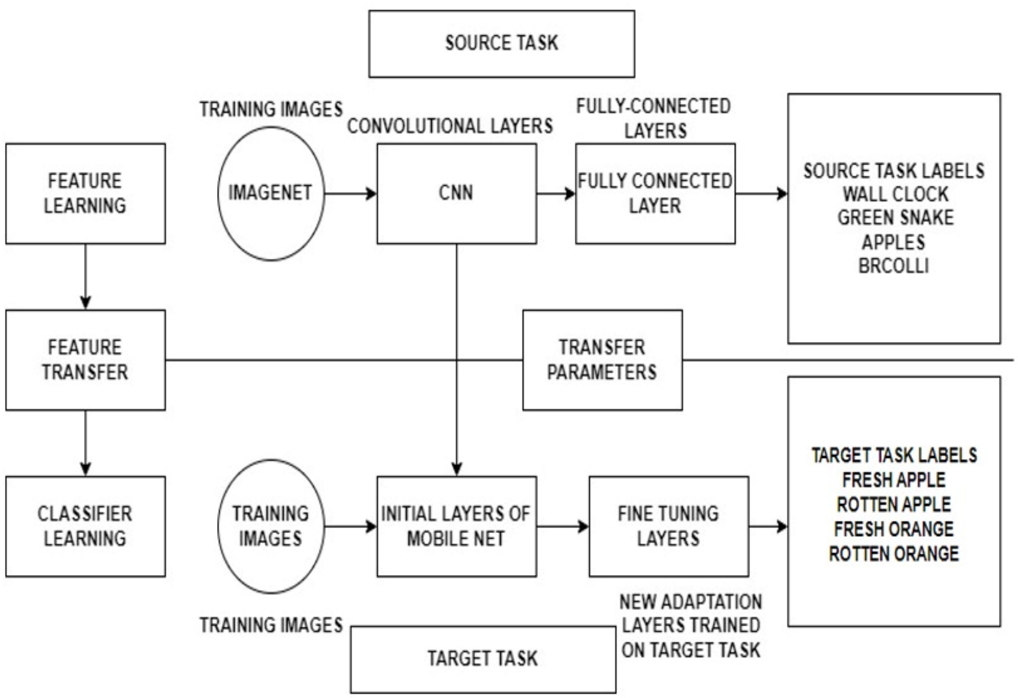


Fig 5.2. Block Diagram of Deep learning

The Figure 5.2 shows the block diagram of deep learning. The block diagrams explain the working of the model in detail, where the datasets are fed into the model and the feature extraction takes place and gives out training dataset, validation dataset and testing datasets. The training datasets are used to train the model and the validation datasets are used to fine tune the model during training. The model used here is mobilenet v1 which has several layers of neural network in it.

The model is already pre-trained. This model is trained with huge number of datasets to get the greater accuracy in prediction. The last few dese layers in the model is removed and trained with our datasets to get favorable accuracy of the model.

5.3 CATEGORICAL CROSSENTROPY:

Categorical cross entropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.

The categorical cross entropy is well suited to classification tasks, since one example can be considered to belong to a specific category with probability 1, and to other categories with probability 0.

The model uses the categorical cross entropy to learn to give a high probability to the correct digit and a low probability to the other digits.

SoftMax is the only activation function recommended to use with the categorical cross entropy loss function. The SoftMax activation rescales the model output so that it has the right properties. Use a single Categorical feature as target.

This will automatically create a one-hot vector from all the categories identified in the dataset. Each one-hot vector can be thought of as a probability distribution, which is why by learning to predict it, the model will output a probability that an example belongs to any of the categories.

CHAPTER 6

TESTING

6.1 TESTING:

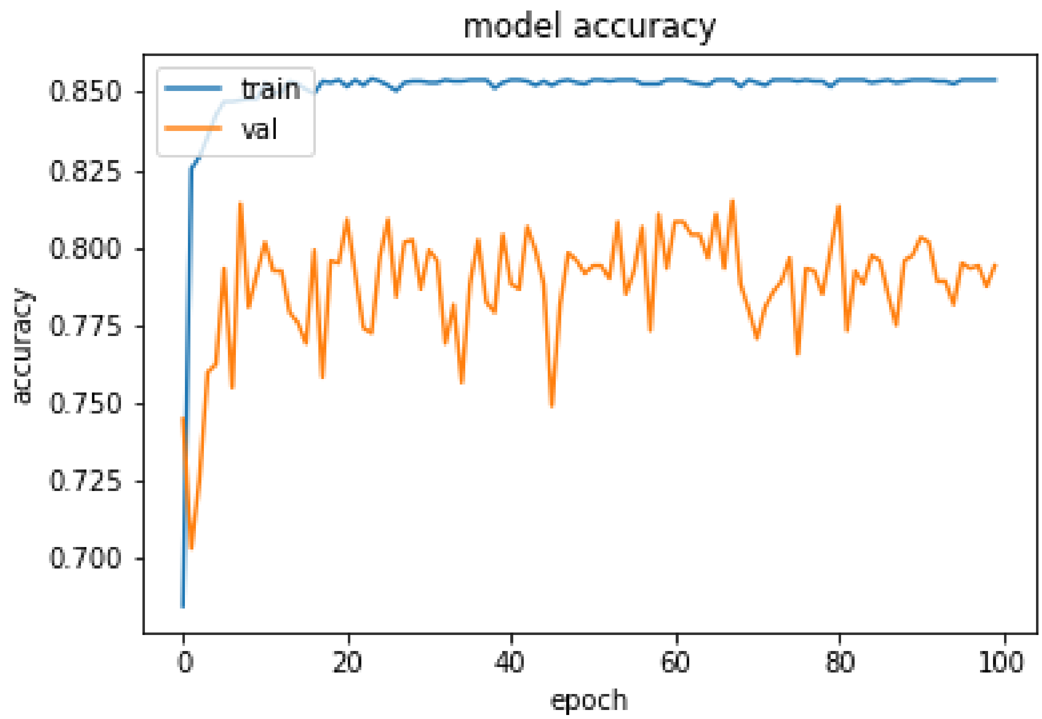


Fig 6.1: Training Accuracy graph after fine-tuning last 7-layers

1. **Fine-Tuning the last 7 layers:**

Training the last 7 layers for 100 epochs, the train dataset results in a model which has a decent accuracy of 85% , but the validation accuracy is very low with a value of close to 80% with a lot of variations as shown in the above graph.

This indicates that the model has not trained in a consistent manner and requires further training and modification to the layers.

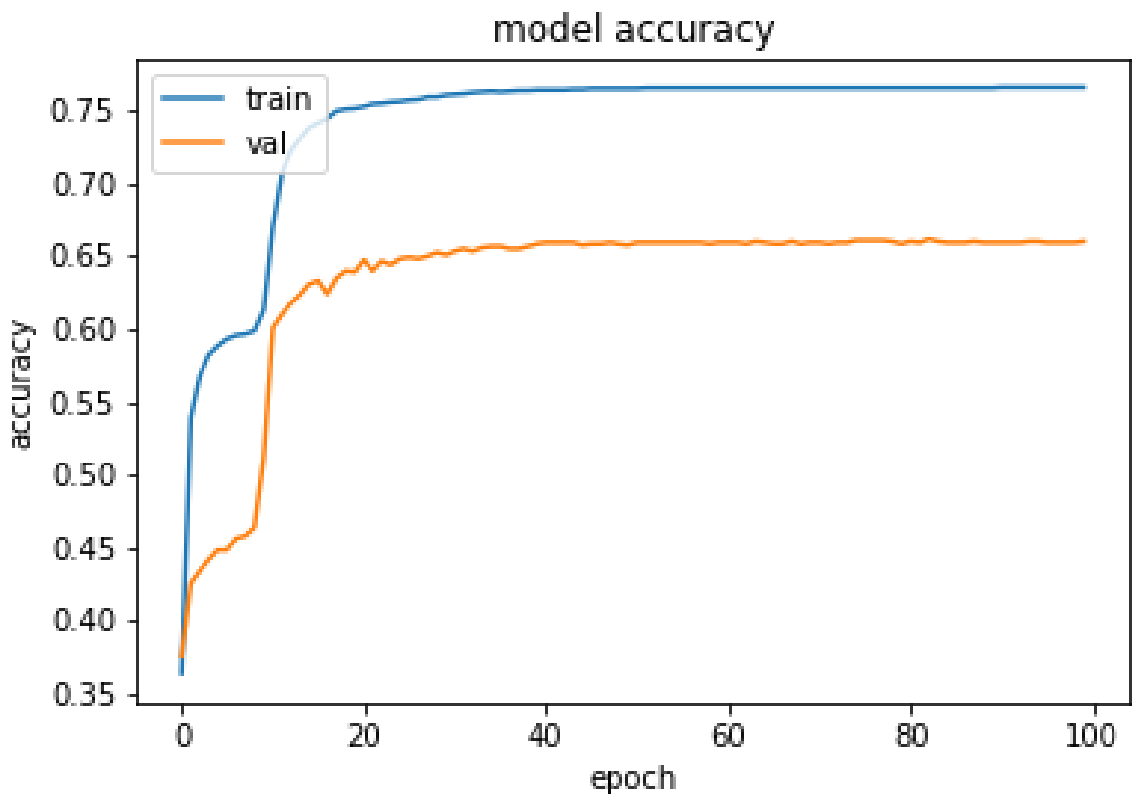


Fig 6.2: Training Accuracy graph after fine-tuning last 9-layers

2. **Fine-Tuning the last 9 layers:**

Training the last 9 layers for 100 epochs, the train dataset results in a model which has an accuracy of 75%, but the validation accuracy is very low with a value of close to 60%. This tells us that even though the model is training consistently, the model is not able to perform well on data that it has not encountered before.

This indicates that the model requires further training and modification to the layers.

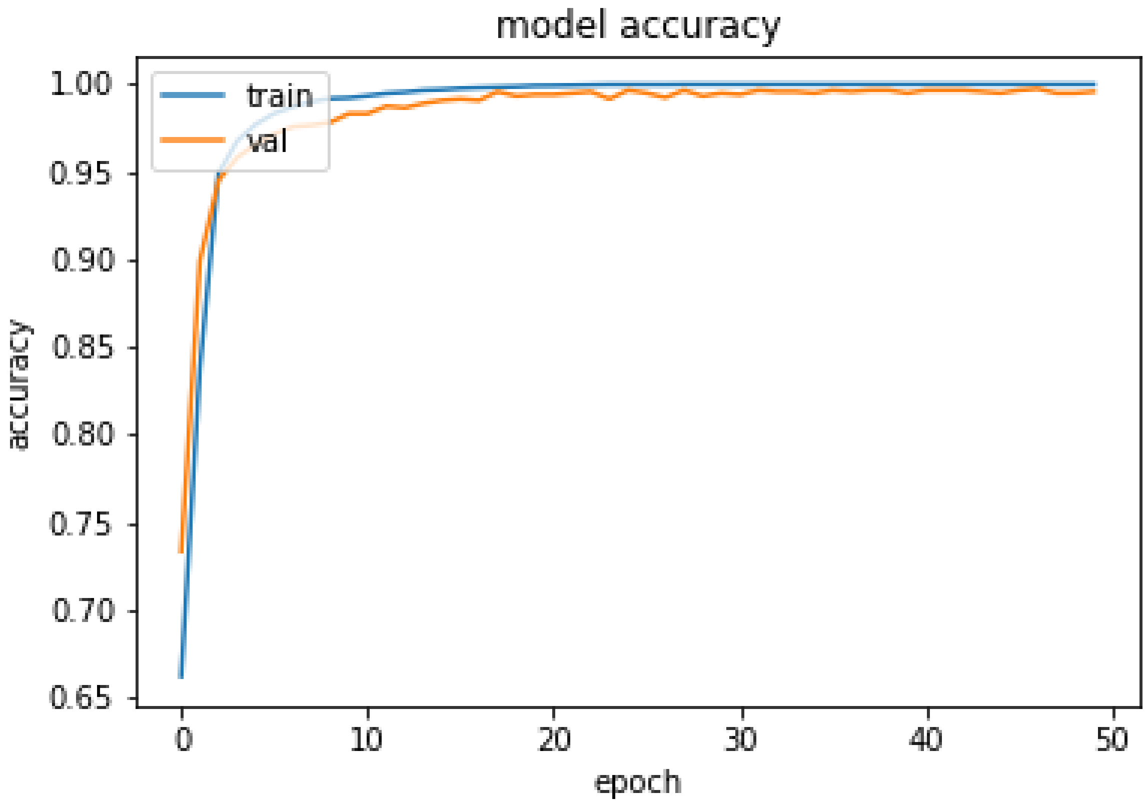


Fig 6.3: Training Accuracy graph after fine-tuning last 5-layers

3. Fine-Tuning the Last 5 layers:

Training the last 5 layers in the data set for 50 epochs leads to model with very consistent training achieving an excellent accuracy of 100% and a validation accuracy of 99.58%. This shows that the model is performing exceptionally well in training and validation showing little to no variation in the graph.

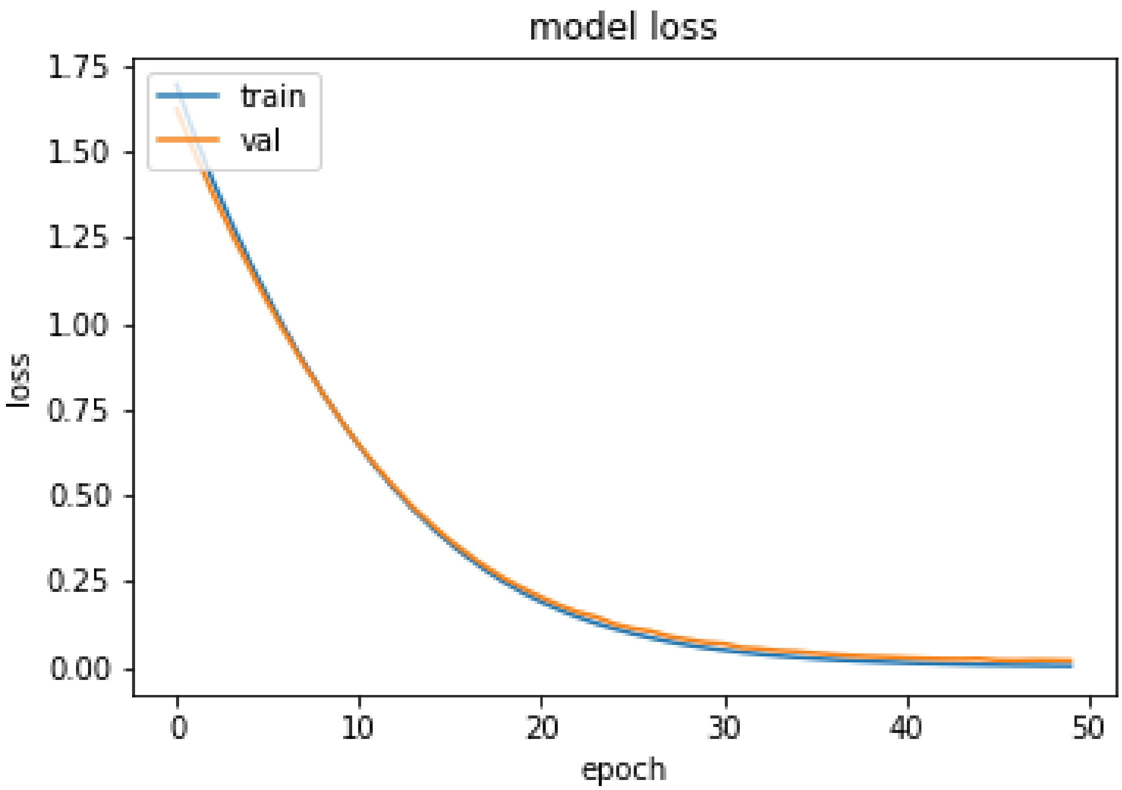


Fig 6.4: Training Loss graph of the model

The loss graph below indicates that the model has a very minimal training loss of 0.35% and a validation loss of 1.8% which indicates that the model is performing really well even with data that it hasn't seen before.

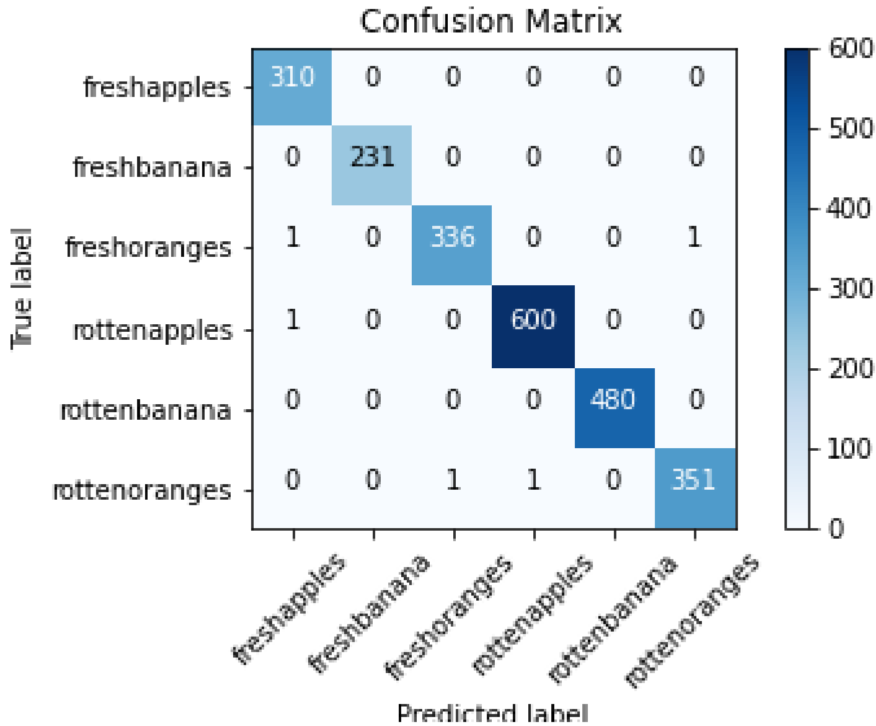


Fig 6.5: Confusion matrix of the trained model

The confusion matrix indicates the number of instances the model is misclassifying from the test dataset (a dataset that the model has seen for the first time). The confusion matrix helps us understand how the model performs, where the model is misclassifying the greatest number of instances. By providing this information, measures can be taken to make changes ot the model or the dataset to ensure the model performs well. In this case, the model does perform well by misclassifying only 5 instances out of 2313 instances.

CHAPTER 7

SNAPSHOTS

The images show the dashboard of the project which is provisioned to upload images of fruits for classification and shelf-life prediction and few snapshots of examples are shown.

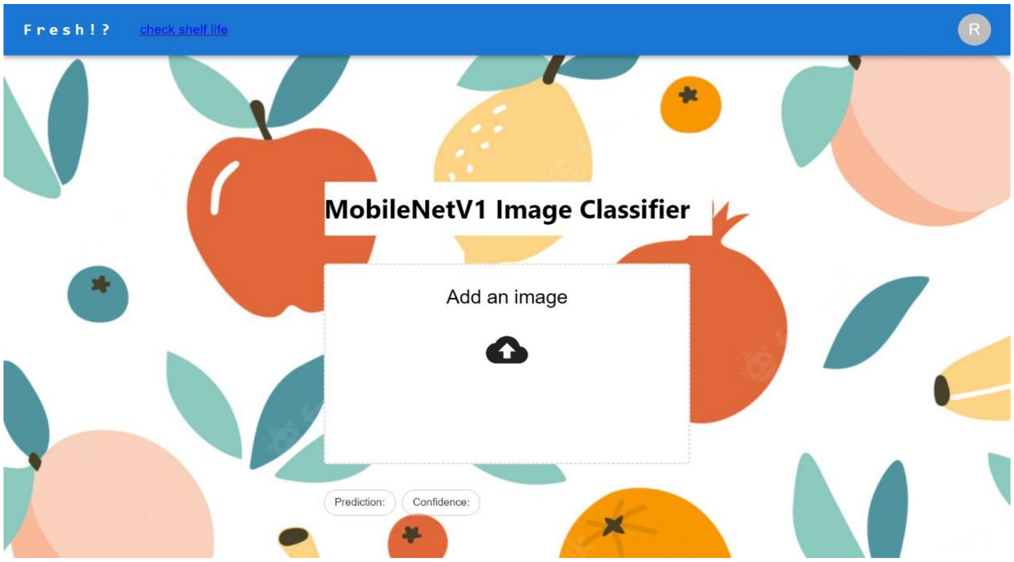


Fig 7.1: Quality Analysis of the fruit (Home Page)

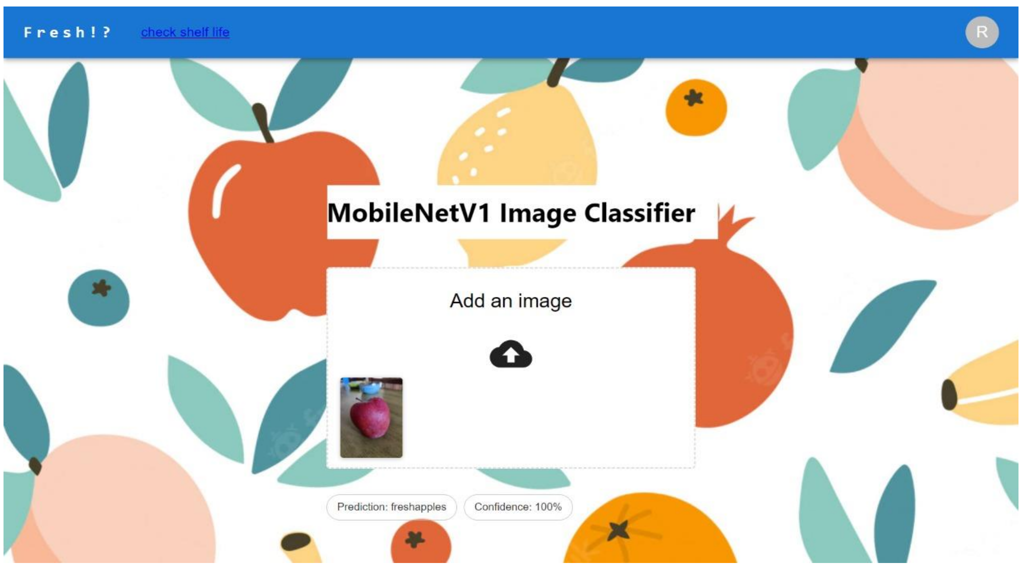


Fig 7.2: Quality of the fruit (shows fresh apple with 100% confidence)

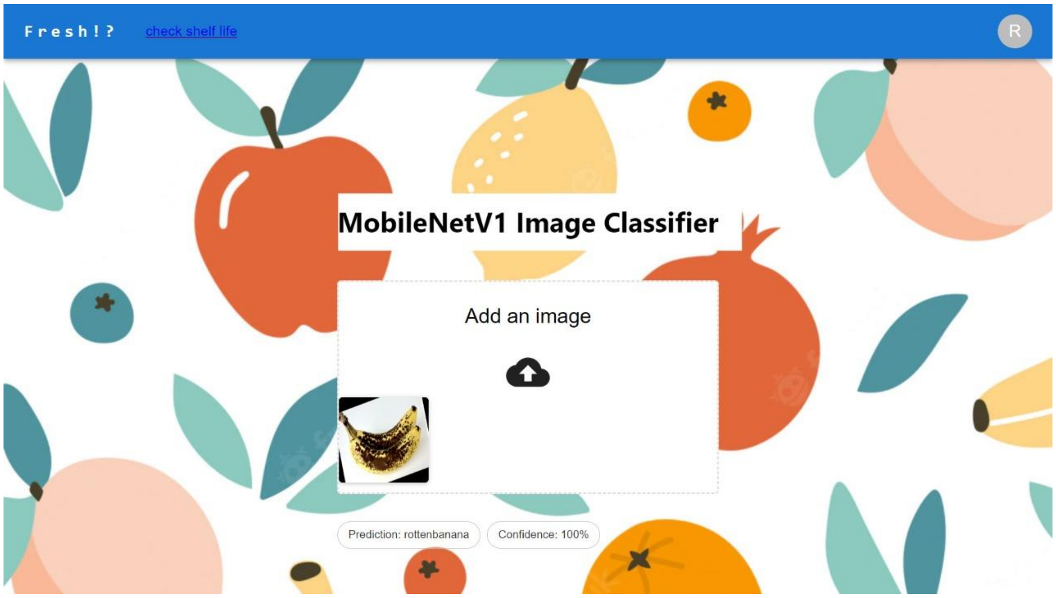


Fig 7.3: Quality of the fruit (shows rotten banana with 100% confidence)

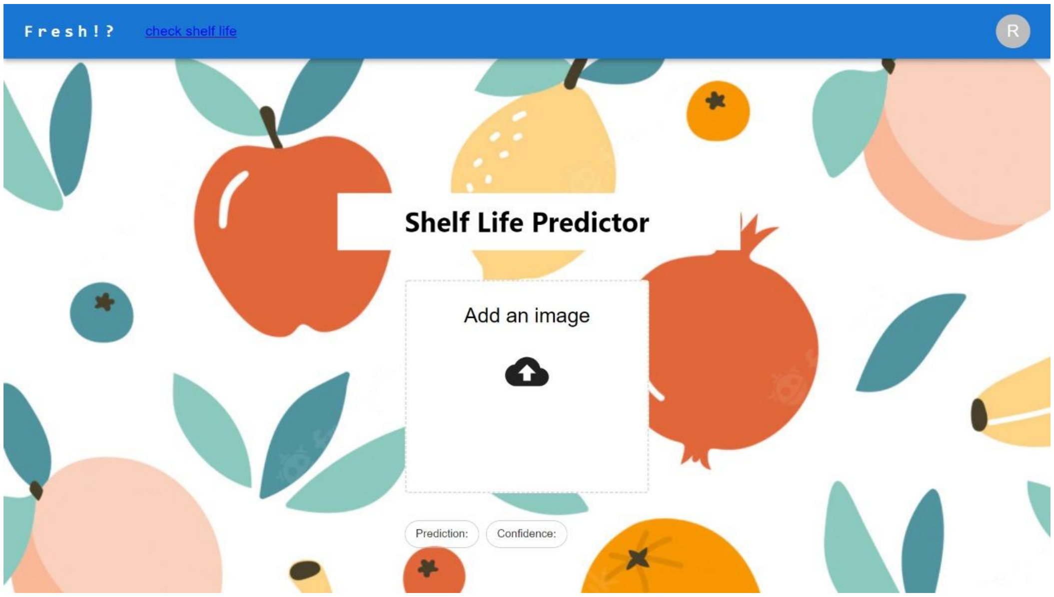


Fig 7.4: Shelf-life predictor

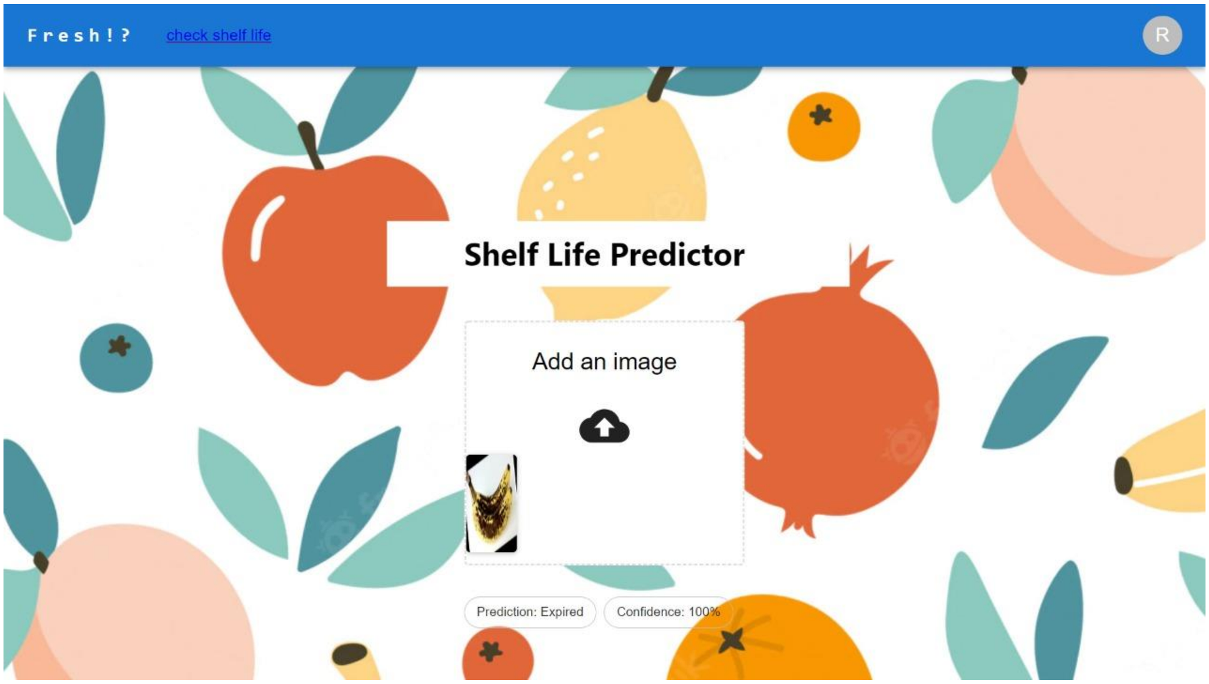


Fig 7.5: Shelf life of rotten banana fruit (shows expired with 100% confidence)

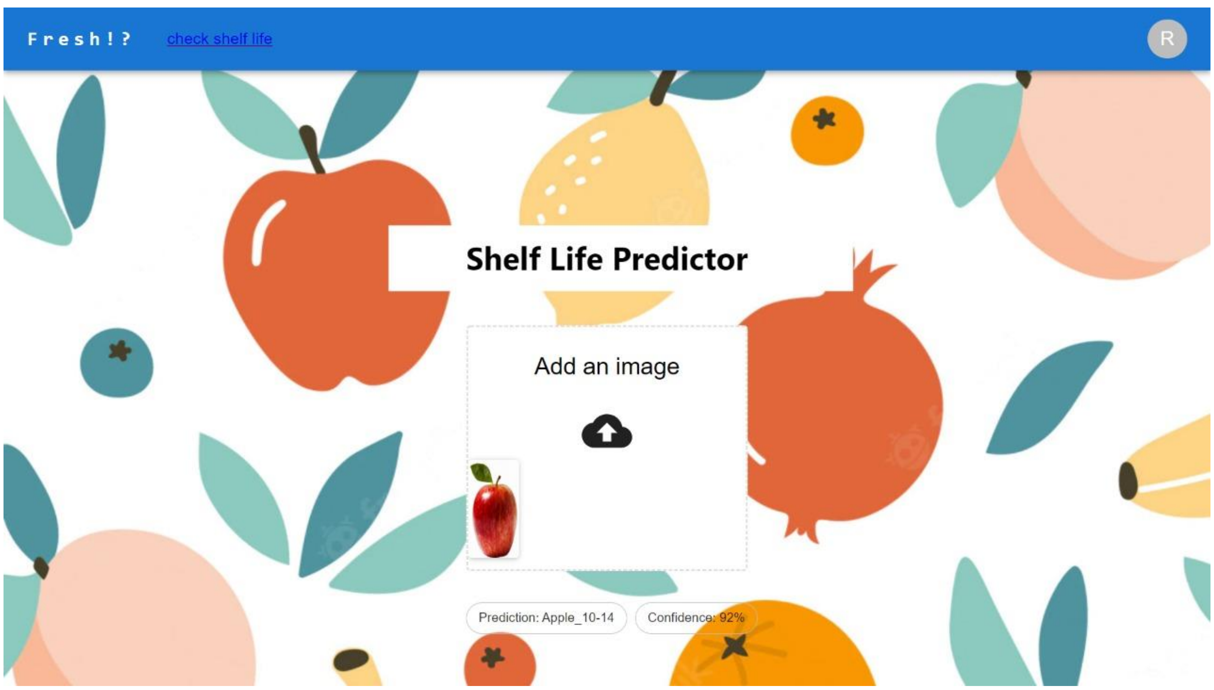


Fig 7.6: Shelf life of fresh apple fruit(10 to 14 days)

CHAPTER 8

RESULTS

The main aim of the project is to check the quality of the fruits as well as predict the shelf life of the same by capturing the image and feeding it to our model. By predicting the shelf life of the fruit and vegetable, we can determine the duration for which the quality of the fruits/vegetables will be fit enough to be sold to the customers. The project has attained high accuracy of 99.58% for image classification and 80.79 % accuracy for shelf-life prediction.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

The project has attained high accuracy of 99.58% for image classification and 80.79 % accuracy for shelf-life prediction. Future plans include training the model on a bigger dataset to predict shelf-life more accurately, as well as implementing the model using hardware to be used in storage facilities.

REFERENCES

[1] Nazrul Ismail & Owais A.Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques",IEEE March 2022.

[2] Mukesh KumarTripathi and Dr Dhananjay D.Maktedar, "A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: A survey ",IEEE June 2020.

[3] Santi Kumari Behera, Amiya Kumar Rath and Prabira Kumar Sethy,. "Maturity status classification of papaya fruits based on machine learning and transfer learning approach " ,IEEE June 2021.

[4] M. Shamim Hossain, Muneer Al-Hammadi and Ghulam Muhammad, "Automatic Fruit Classification Using Deep Learning for Industrial Applications",IEEE October 2018.

[5] Fatma M. A. Mazen & Ahmed A. Nashat, "Ripeness Classification of Bananas Using an Artificial Neural Network",IEEE January 2019.

[6] S. Marimuthu and S. M. Roomi, "Particle swarm optimized fuzzy model for the classification of banana ripeness", IEEE Sensors J., vol. 17, no. 15, pp. 4903-4915, Aug. 2017.

[7] Manali Shaha and Meenakshi Pawar,"Transfer Learning for Image Classification",IEEE: 2nd Conference,April 2018.

[8] Sajja Tulasi Krishna and Hemantha Kumar Kalluri, "Deep Learning and Transfer Learning Approaches for Image Classification", ISSN:2277-3878,Februry 2019.

[9] Shweta S.Deulkar, Sunita S. Barve, "Feature based Fruit Quality Grading System using Support Vector Machine ",3rd IEEE International Conference, pp. 96-104, 2018.

[10] W. Zhang, J. Hu, G. Zhou, and M. He, ‘‘Detection of apple defects based on the FCM-NPGA and a multivariate image analysis,’’ IEEE Access, vol. 8, pp. 38833–38845, 2020.

[11] Yo-Ping Huang;Tzu-Hao Wang;Haobijam Basanta, "Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness",IEEE 2020.

[12] A F. Garcia, J. Cervantes, A. Lopez, and M. Alvarado, ‘‘Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets,’’ IEEE Latin Amer. Trans., vol. 14, no. 7, pp. 3434–3443, Jul. 2016.

[13] M. B. Garcia, S. Ambat, and R. T. Adao, ‘‘Tomayto, tomahto: A machine learning approach for tomato ripening stage identification using pixel-based color image classification,’’ in Proc. IEEE 11th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ., Manage. (HNICEM), Laoag, PA, USA, Nov. 2019.

[14] F. Garcia, J. Cervantes, A. Lopez, and M. Alvarado, ‘‘Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets,’’ IEEE Latin Amer. Trans., vol. 14, no. 7, pp. 3434–3443, Jul. 2016. P. Li et al., "Deep convolutional computation model for feature learning on big data in internet of things", IEEE Trans. Ind. Inform., vol. 14, no. 2, pp. 790-798, Feb. 2018.

[15] Z. Gao, L. Wang, L. Zhou and J. Zhang, "HEp-2 cell image classification with deep convolutional neural networks", IEEE J. Biomed. Health Inform., vol. 21, no. 2, pp. 416-428, Mar. 2017.

[16] C. Hu, X. Liu, Z. Pan, and P. Li, ‘‘Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set,’’ IEEE Access, vol. 7, pp. 154683–154696, Oct. 2019.

[17] X. Liu, D. Zhao, W. Jia, W. Ji, and Y. Sun, ‘‘A detection method for apple fruits based on color and shape features,’’ IEEE Access, vol. 7, pp. 67923–67933, 2019.

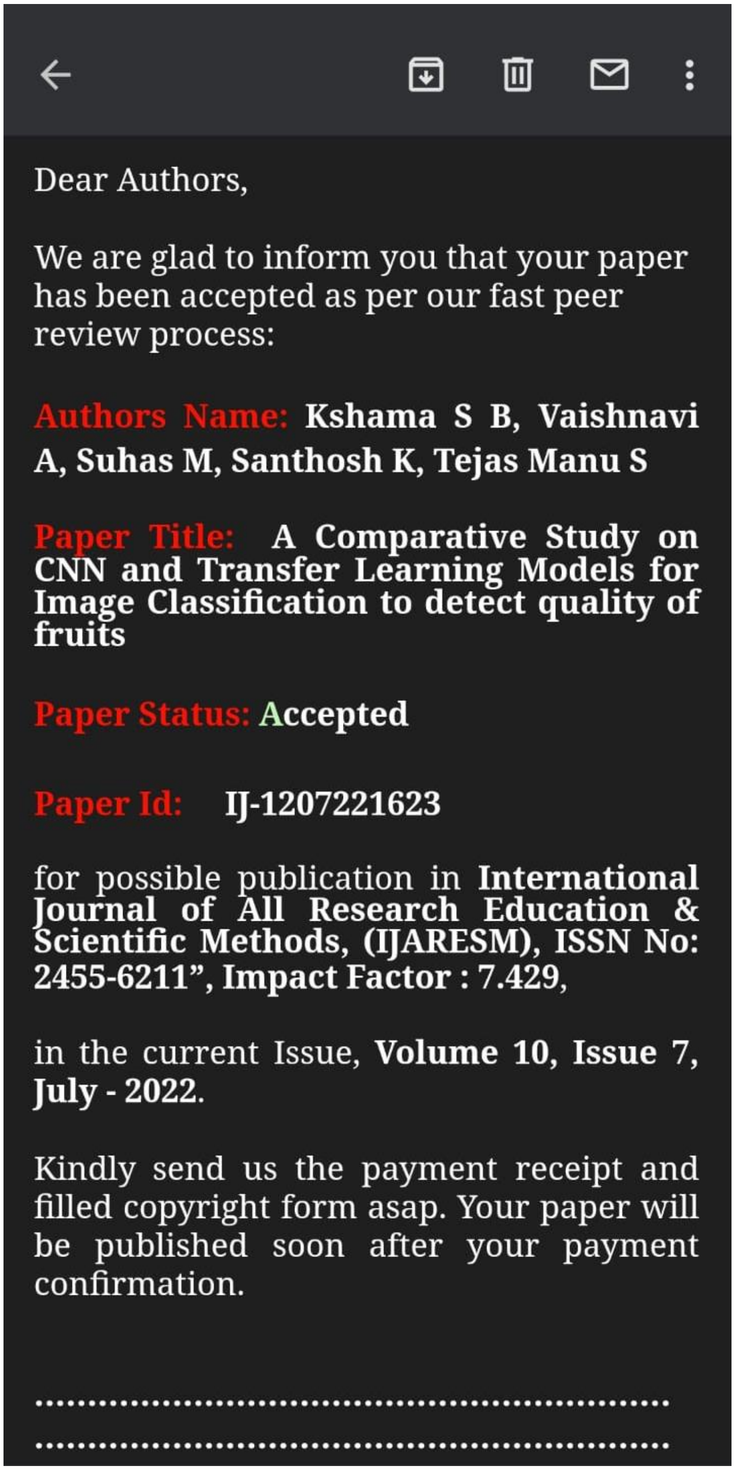
[18] G. Muhammad, "Date fruits classification using texture descriptors and shape-size features", Eng. Appl. Artif. Intell., vol. 37, pp. 361-367, 2015.

[19] V. Leemans and M.-F. Destain, ‘‘A real-time grading method of apples based on features extracted

from defects,'' J. Food Eng., vol. 61, no. 1, pp. 83–89, Jan. 2004.

[20] D. Li, M. Shen and X. Yu, ''Green apple recognition method based on the combination of texture and shape features,'' in Proc. IEEE Int. Conf. Mechatron. Automat. (ICMA), Takamatsu, Japan, pp. 264–269, Aug. 2017.

PAPER ACCEPTENCE SNAPSHOT



Dear Authors,

We are glad to inform you that your paper has been accepted as per our fast peer review process:

Authors Name: **Kshama S B, Vaishnavi A, Suhas M, Santhosh K, Tejas Manu S**

Paper Title: **A Comparative Study on CNN and Transfer Learning Models for Image Classification to detect quality of fruits**

Paper Status: **Accepted**

Paper Id: **IJ-1207221623**

for possible publication in **International Journal of All Research Education & Scientific Methods, (IJARESM), ISSN No: 2455-6211", Impact Factor : 7.429,**

in the current Issue, **Volume 10, Issue 7, July - 2022**.

Kindly send us the payment receipt and filled copyright form asap. Your paper will be published soon after your payment confirmation.