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 likelihood that the image belongs to a "class"). A class is basically a label, like "vehicle," "animal," "structure," and so forth. Raw pixel data was the foundation of early image classification. This implied that computers would dissect images into their component pixels. The issue is that the same subject can appear substantially differently in two distinct photographs. They may have various backdrops, perspectives, poses, etc. 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

The concept behind supervised classification is that a user can choose a sample set of pixels from an image that best represents a given class, and then instruct the image processing software to utilise these training sites as references when classifying 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

Machine learning—a subset of artificial intelligence (AI) that enables machines to learn from data—involves deep learning. Neural networks are a type of computer system used in deep learning. In neural networks, the input is filtered through node layers that are not visible. These nodes each process the input and transmit their findings to the nodes below them. It keeps doing this until it reaches an output layer, at which point the machine responds. Based on how the hidden layers function, many neural network types exist. Convolutional neural networks, or CNNs, are frequently used for image classification in deep learning. In CNN, not all of the nodes in the hidden layers always share their output with all of the nodes in the subsequent layer (known as convolutional layers).

Machines can recognize and extract features the details of the images from deep learning. This implies that they can learn the characteristics to look for in images by examining numerous photographs. Thus, these filters don't need to be manually entered by programmers.

1.4 TRANSFER LEARNING

A model created for one task is used as the basis for another using the machine learning technique known as transfer learning. Given the enormous computing and time resources needed to develop neural network models on these problems and from the enormous leaps in skill that they provide on related problems, it is a common approach in deep learning to use pre-trained models as the starting point on computer vision and natural language processing tasks.

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**. One of the accessible models is a source model that has already been trained. In the pool of potential candidate models from which to choose, several research organisations publish models on significant and difficult datasets.
b. **Reuse Model**. The model that has already been trained can then serve as the foundation for a model on the second task of interest. Depending on the modelling technique employed, this may entail using the entire model or just certain portions of it.
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.

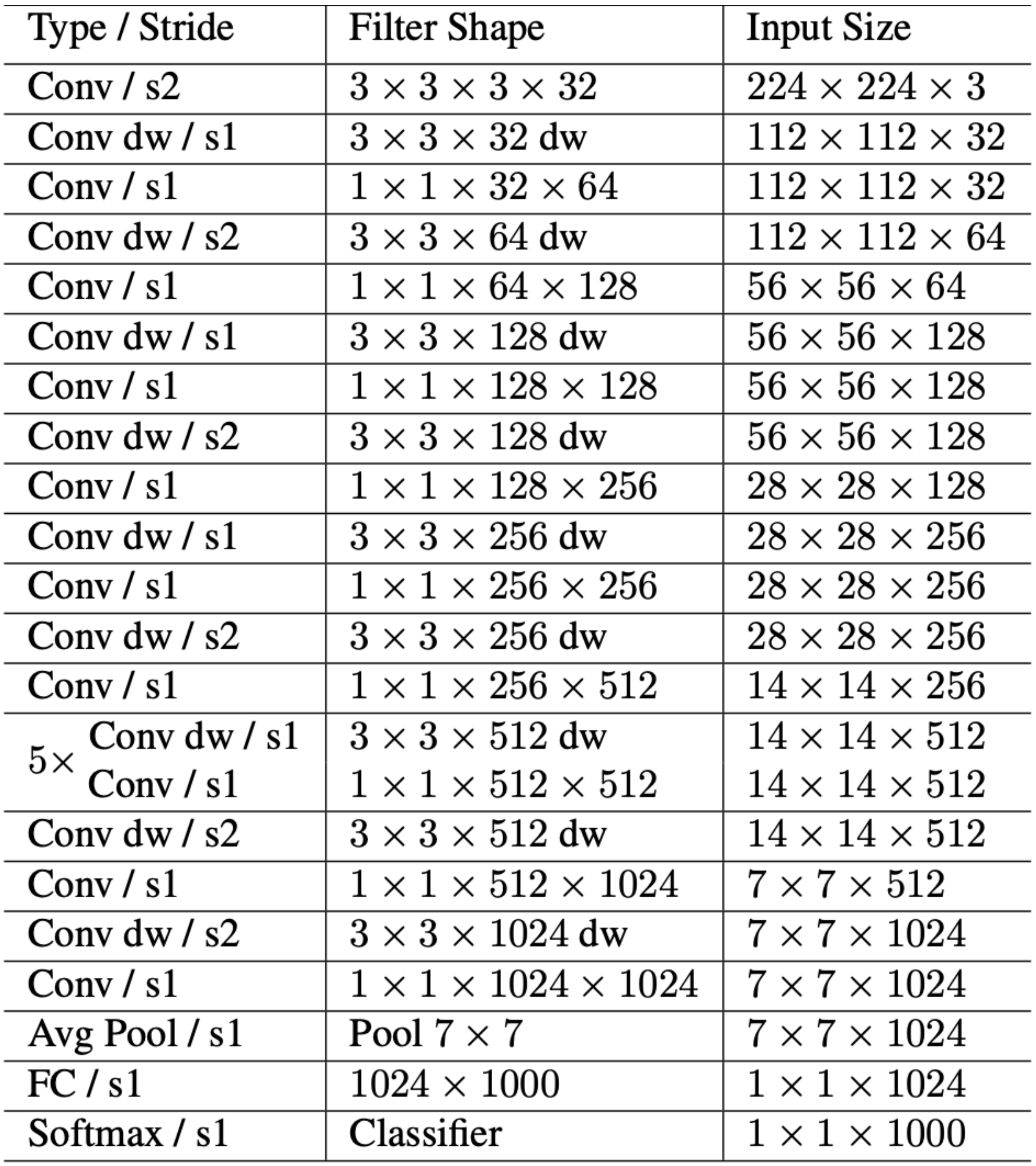
1.5 MOBILENETV1

Convolutional neural networks, such as MobileNet, are specialised for use in embedded and mobile vision applications. They are based on a simplified design that makes use of depth-wise separable convolutions to construct compact deep neural networks with reduced latency for embedded and mobile devices. In order to streamline the computation in the initial few layers of Inception models, MobileNets—which are essentially constructed from depth-wise separable convolutions—are employed. Flattened networks demonstrated the possibilities of extremely factorised networks by constructing a network consisting of entirely factorised convolutions. Each input channel for MobileNets receives a single filter thanks to the depth-wisee convolution.

The outputs of the depth-wise convolution are combined using an 11 convolution after the pointwise convolution. In one step, a conventional convolution filters and combines inputs to create a new set of outputs. This is divided into two layers by the depth-wise separable convolution: a layer for combining and a layer for filtering. The computation and model size are significantly decreased as a result of this factorization.

With only a slight loss in accuracy, MobileNet's 3 3 depth wise separable convolutions require 8 to 9 times less processing than conventional convolutions.

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× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5× 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 method for summarising a classification algorithm's performance is the confusion matrix. If your dataset has more than two classes or if each class has an unequal amount of observations, classification accuracy alone may be deceiving. You can acquire a better understanding of the categorization model's successes and failures by calculating a confusion matrix.

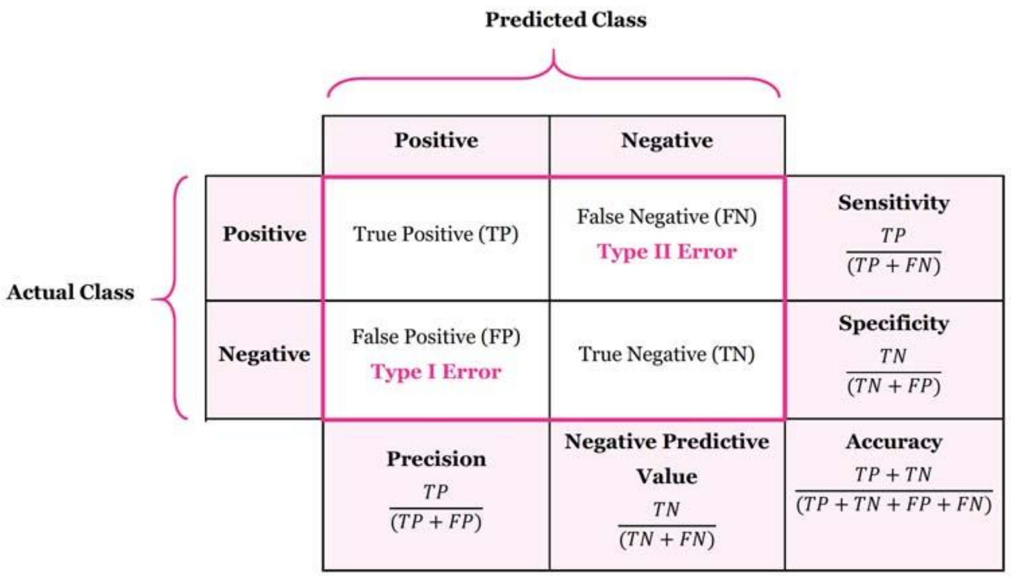


Fig 1.2: Confusion Matrix

1.7 KERAS

On top of the machine learning framework Tensor Flow, Keras is a Python-based deep learning API. It was created with the goal of facilitating quick experimentation. The secret to conducting effective research is being able to move quickly from conception to conclusion.

Keras is

- **Simple** -- Simple, yet not naive. Keras lessens the cognitive strain on developers so you may concentrate on the crucial aspects of the issue.
- **Flexible** -- Straightforward workflows should be quick and simple, whereas arbitrarily advanced workflows should be feasible via a clear path that builds upon what you've

already learned, according to the progressive disclosure of complexity principle adopted by Keras.

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.

Interactive user interfaces may be easily made with React. Create straightforward views for each stage of your application, and when the data changes, React will quickly update and render the appropriate components.

We can use React to describe web user interfaces since Web browsers are capable of understanding JavaScript. With React, we essentially just tell it what we want, and it will construct the real User Interfaces in the Web browser on our behalf. This is why I like to use the term describe in this sentence. Without React or other similar tools, user interfaces would have to be manually built using JavaScript and native Web APIs. Declarative views improve the predictability and debuggability of your code.

1.9 MATERIALS UI

To develop a user interface in our React applications, we can simply import and use several components from the Material-UI package. As a result, the developers can save a lot of time by not having to write everything from start.

MUI provides a complete set of UI tools to help you release new features more quickly. Bring your own design system to our production-ready components, or start with Material UI, our fully loaded component library.

Google's guidelines for creating user interfaces served as a major source of inspiration for Material-UI widgets. Therefore, creating aesthetically appealing applications is simple for developers. Here is where you can read more about Google's material design ideas.

1.10 PYTHON

A multi-paradigm programming language is Python. Many of its features allow functional programming and aspect-oriented programming, and object-oriented programming and structured programming are all fully supported (including metaprogramming and metaobjects [magic methods]). Extensions are available for many additional paradigms, such as design by contract and logic programming.

Python's memory management system combines reference counting and a cycle-detecting garbage collector with dynamic typing. Method and variable names are bound via dynamic name resolution (late binding), which takes place while the programme is running.

A little bit of assistance for functional programming in the Lisp style is provided by its design. In addition to list comprehensions, dictionaries, sets, and generator expressions, it contains filter and map-and-reduce capabilities. Two modules from the standard library—itertools and functools—implement functional tools adapted from Haskell and Standard ML.

Aphorisms like "Beautiful is better than ugly" are part of the document The Zen of Python, which outlines its underlying philosophy.

Explicit is preferable to implicit.

Simple is preferable to complex.

Complexity is preferable to complexity.

Readability is crucial.

Python was made to be extremely extendable via modules rather than having all of its capabilities built into its core. It has gained popularity as a way to add programmable interfaces to existing applications because of its compact modularity. the idea of Van Rossum. Van Rossum's dissatisfaction with ABC, which advocated the opposite strategy, led to his concept of a tiny core language with a huge standard library and easily expandable interpreter.

Python gives developers a choice in their development style while aiming for a simpler, less cluttered syntax and grammar. Python adheres to the "there should be one—and preferably only one—obvious way to do it" tenet as opposed to Perl's "there is more than one way to do it".

1.11 CASCADING STYLE SHEETS

Layout, colour, and font choices can all be differentiated between presentation and content using CSS. This division can enhance content accessibility, offer more flexibility and control in the specification of presentation characteristics, allow multiple web pages to share formatting by specifying the pertinent 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 page load speed between the pages that share the file and its formatting.

Separating formatting from content also enables the presentation of the same markup page in many rendering modes, including on-screen, in print, by voice (using a screen reader or speech-based browser), and on Braille-based tactile devices.

When many style rules match an element, the priority system is used to determine which rule should be applied, hence the name cascading. This priority hierarchy is predictable.

The World Wide Web Consortium maintains the CSS specifications (W3C). RFC 2318 has registered the Internet media type (MIME type) text/CSS for use with CSS (March 1998). For CSS documents, the W3C offers a free CSS validation service.

1.12 MACHINE LEARNING

With the use of machine learning (ML), which is a form of artificial intelligence (AI), software programmes can predict outcomes more accurately without having to be explicitly instructed to do so. In order to forecast new output values, machine learning algorithms use historical data as input.

Machine learning is significant because it aids in the development of new goods and provides businesses with a picture of trends in consumer behavior and operational business patterns.

There are four fundamental strategies: reinforcement learning, semi-supervised learning, unsupervised learning, and supervised learning. The kind of data that data scientists wish to predict determines the type of algorithm they use.

- Supervised learning: For this sort of machine learning, data scientists describe the variables they want the computer to look for correlations between and provide the algorithms with labelled training data. The algorithm's input and output are both given.
- Unsupervised learning: Algorithms used in this sort of machine learning are trained on unlabeled data. The algorithm searches through data sets in search of any significant relationships. Both the predictions or recommendations generated by algorithms and the data used to train them are predefined.
- Semi-supervised learning: These two forms of machine learning are combined in this method. An algorithm may be fed with primarily labelled training data by data scientists, but the model is allowed to explore the data on its own and form its own knowledge of the collection of data.
- Reinforcement learning: Reinforcement learning is frequently used by data scientists to train a system to finish a multi-step process with well-defined criteria. An algorithm is programmed by data scientists to fulfil a goal, and they provide it with positive or negative feedback as it determines how to do so. However, the algorithm generally chooses what actions to take at each stage on its own.

1.13 HYPER PARAMETER TUNING

By integrating data from an existing neural network and utilizing it as an initialization point to make the training process more time and resource effective, fine-tuning deep learning algorithms will help to increase the accuracy of a new neural network model.

The datasets of an existing model and the new deep learning model must be similar in order for fine-tuning to be effective in training new deep learning algorithms.

A model that has already been trained to execute a specific task is fine-tuned or altered to enable it to carry out a second similar task. A deep learning network that has been trained to identify cars, for instance, can be adjusted to identify 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 phase is optional and is based on how closely the two learning models resemble one another. Depending on how close the two models are, you might need to add or subtract a few layers. You must then freeze the layers in the new model once you've added or removed layers depending on the data needed..
4. When a layer is frozen, the data contained therein is no longer subject to alteration. When we train the new model on the new data for the new job, the weights for these layers don't change.
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 EXISTING 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 issue.

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.

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 with 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 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. The first phase describes extracting the image features such as image quality, area, perimeter, mean, variance, color, and intensity. The second phase describes k-means clustering is used for image segmentation. In the last phase, the vectors of color, intensity, segments, and image quality features were utilized for training the support vector machine structure.

2.3. Maturity status classification of papaya fruits based on machine learning and transfer learning approach

This paper 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 also the 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 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 at different ripening stages on the MUSA database.

The artificial neural network-based framework which uses colour, development of brown spots, and Banana fruit ripening stage classification and grading uses Tamura statistical texture feature. Results and the suggested system's performance are contrasted with those of other methods 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 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 a support vector machine classifier. VGG19 CNN Architecture is best compared with the others. Using a deep convolutional neural network, the cell image classification is done.

2.6. Automatic fruit classification using deep learning for industrial applications

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

The framework is based on two different deep learning architectures. The first is a proposed light model of six convolutional neural networks whereas the second is a refined visual geometry group-16 pre-trained deep learning model, there are two layers in the first. Two colour 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, and Classifier. In addition to this k-means clustering is used to achieve better improvement of the accuracy and speed of the classifier.

The purpose to use the HSV (Hue, Saturation, Value) space is because it is possible to chromaticity data can be extracted and processed without the RGB space's unfavourable intensity. On the other hand, before the color is characterized, we complete a selection of the chromaticity that contribute to important data about the fruits.

2.8. A real-time grading method of apples based on features extracted from defects

In this 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 an apple that is apple is examined at 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%.

2.9. Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness

Tomatoes must be manually inspected and harvested, which takes a lot of time and effort. To that purpose, we suggested a fuzzy Mask R-CNN model to automatically determine the maturity degrees of cherry tomatoes. First, a fuzzy c-means model was employed to keep the spatial information of distinct foreground and background objects in the photos in order to automatically annotate them. Once the precise geometric edge positions of the tomatoes were determined, a Hough transform technique was used. Each data point in the image space had a JavaScript Object Notation file attached to it.

Second, Mask R-CNN was trained on annotated photos to precisely identify each tomato. Finally, a hue-saturation-value color model and fuzzy inference rules were employed to forecast the freshness of the tomatoes in order to prevent preharvest abscission. To quickly determine the location of the pedicle head and dissect the fruit, a trigonometric function with Euclidian distance was calculated from the origin of the calyx and stem to the bottom of the tomato. Mask R-CNN achieved a detection accuracy of 98.00% on 100 tomato photos. Overall weighted precision and recall rates for the classification of tomato ripeness were 0.9614 and 0.9591, respectively. Applications for robotic tomato harvesting can therefore help farmers make wiser choices and increase overall production efficiency and yield.

2.10. Date fruits classification using texture descriptors and shape-size features

A method for automatically categorizing dates based on their photos. Different dates have different distinguishing characteristics that can be helpful to identify a specific date. Color, texture, and shape are some of these characteristics. The suggested technique breaks down a visual image of a date into its component colors. The local texture descriptor, such as a Weber local descriptor (WLD) histogram or a local binary pattern (LBP), is then applied to each component in order to encode the texture pattern of the date.

To characterize the image, the texture patterns from each component are combined. The feature set's dimensionality is decreased by using Fisher discrimination ratio (FDR)-based feature selection. To completely characterize the date, size and form features are added to the texture descriptors. We employ support vector machines as a classifier. The suggested approach successfully categorizes dates with an accuracy of greater than 98 percent.

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 a software system is defined by functional requirements.
- The behavior of the system is evaluated when it is presented with particular inputs or conditions, such as calculations, data processing and manipulation.

3.4 Non-Functional Requirements

- **Reliability:** Any computer-related component (software, hardware, or a network, for instance) that regularly functions by its requirements is said to be reliable. It has long been seen as one of three connected characteristics that must be taken into account when creating, purchasing, or utilizing a computer product or component. Any system should be designed with reliability, availability, and serviceability—or RAS, as it is commonly known—in mind. Technical defects are completely absent from a reliable product in theory, but in fact, sellers usually express a product's dependability quotient as a percentage. Since it is presumed that bugs have been fixed in prior releases, evolutionary products (those that have developed through several versions over a substantial amount of time) are typically thought to become better.

- **Performance:** The effectiveness of a particular computer system, or how well the computer performs overall, is known as its performance. An assessment of a computer system's resources and outputs to evaluate whether it is operating at its best is known as a computer performance evaluation. Like a handyman using a voltmeter to check the voltage across a circuit, it is comparable to a voltmeter. The meter checks to make sure the circuit is receiving the proper voltage. Similar to this, a PC's performance can be evaluated using recognised benchmarks to see whether it is operating properly.

- **Portability:** The ease with which a programme can be moved from one computing environment to another is referred to as portability in the context of software. If the effort needed to adapt a computer software application to a new environment falls within realistic

bounds, it is said to be portable. The meaning of the amorphous phrase "reasonable" relies on the context in which it is used and is frequently challenging to describe in quantifiable terms. To port, software refers to changing it so that it can run 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 capacity of IT systems, including networking, storage, databases, and applications, to continue performing effectively when their volume or size changes. It frequently refers to adjusting resources to satisfy either higher or reduced commercial demands. Vertical (scale-up) scalability is adding resources to a physical system, such as more processing power to a server to make it faster and to expand the capacity of hardware or software. When extra capacity is needed, scale-up storage involves expanding an existing system with new components, such as disc drives. Multiple things are linked together to function as a single logical unit thanks to horizontal (scale-out) scalability. This entails adding hardware to connected arrays or clusters for scale-out storage. Numerous nodes (devices) may be present in each cluster, and nodes may be geographically dispersed. Network-attached storage that scales out expands by adding clustered nodes. Performance rises in tandem with storage capacity since each node has processing power, I/O (input/output) bandwidth, and 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 is concerned with preventing damage, theft, and unauthorized use of computer systems and data. Users are routinely attacked because they don't have strong enough security to keep off intruders, and cybercriminals are eager to take advantage of such flaws. Your computers' confidentiality, integrity, and availability are all guaranteed by computer security.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

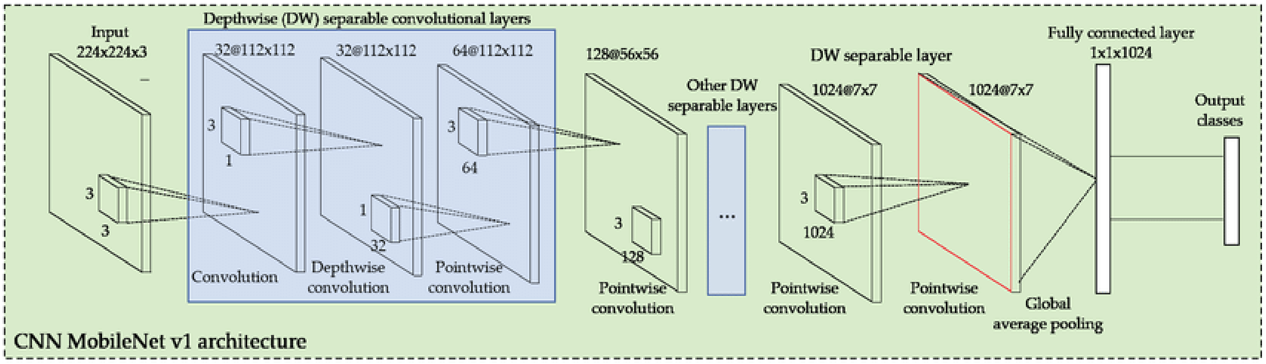


Fig 4.1: MobileNet v1 architecture

To effectively develop lighter models compared to older designs, MobileNetV1 uses depth-wise separable convolutions, which factorise a normal convolution into a depth wise convolution and a pointwise convolution. Additionally, MobileNetV1 introduces the width multiplier and resolution multiplier, two global hyper-parameters that enable a trade-off between latency and accuracy. As a result, the MobileNetV1 is constructed with several depth-wise separable convolution layers, each of which is made up of a depth-wise convolution and a point-wise convolution. By counting depth- and point-wise convolution as separate layers, the MobileNetV1 comprises 28 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.

A global hyperparameter called the width multiplier, symbolized by the symbol is employed to build less complex and computationally expensive models. Its value ranges from 0 to 1. To reduce computation costs and model size at the expense of performance, for a particular layer and value, the number of input channels 'M' becomes * M and the number of output channels 'N' becomes * N. The number of parameters and computation cost are both roughly reduce by a factor of 2. 1,0.75, 0,55, and 0.25 are a few of the often used values.

The resolution multiplier, indicated by the symbol, is the second parameter added to MobileNets. This hyperparameter is used to lower the input image's resolution, which then lowers the input to each layer by the same amount. The input image's resolution changes to 224 * for a given value of. This results in a factor of 2 reductions in the computing cost.

4.2 BLOCK DIAGRAM

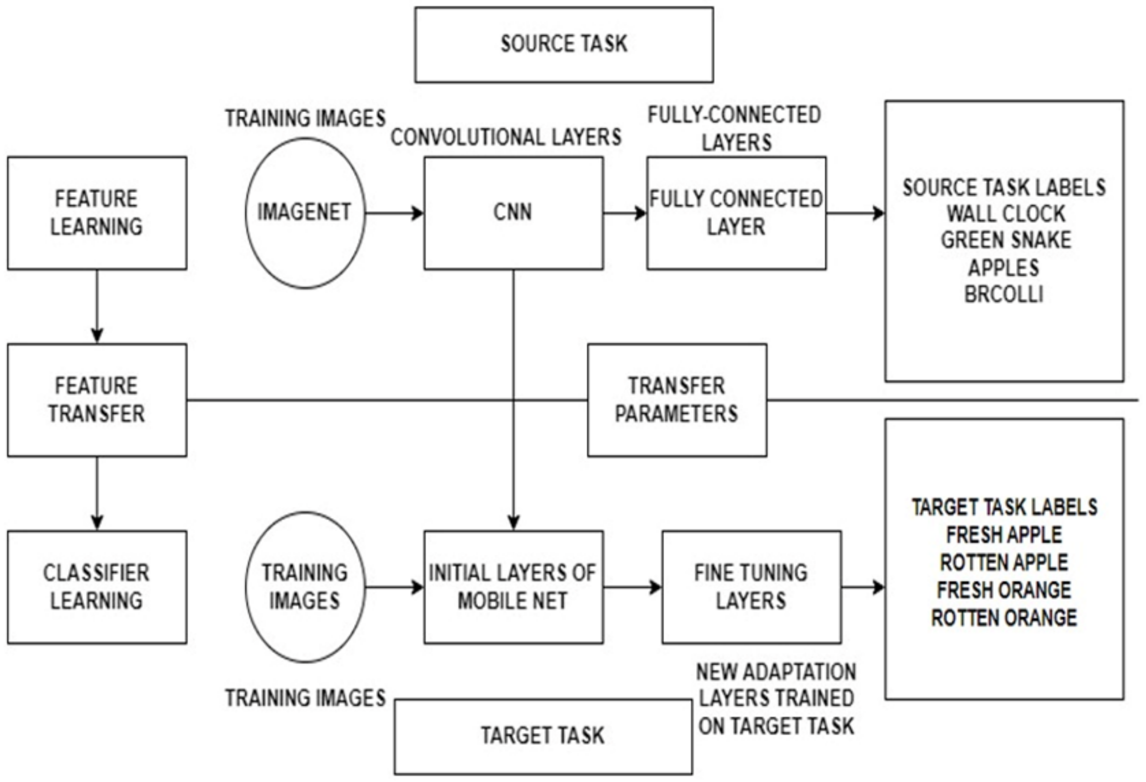


Fig 4.2: Block diagram of Transfer Learning

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 the 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 are 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

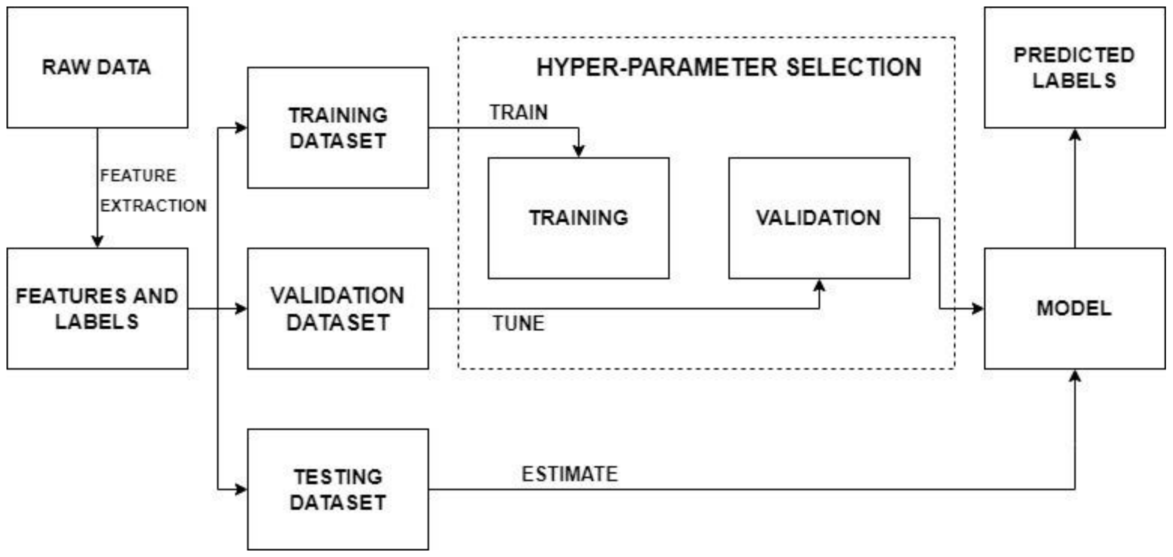


Fig 4.3: Flowchart of Transfer Learning

The training data is pre-processed with the use of a mobile net 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 overfitting or underfitting. 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

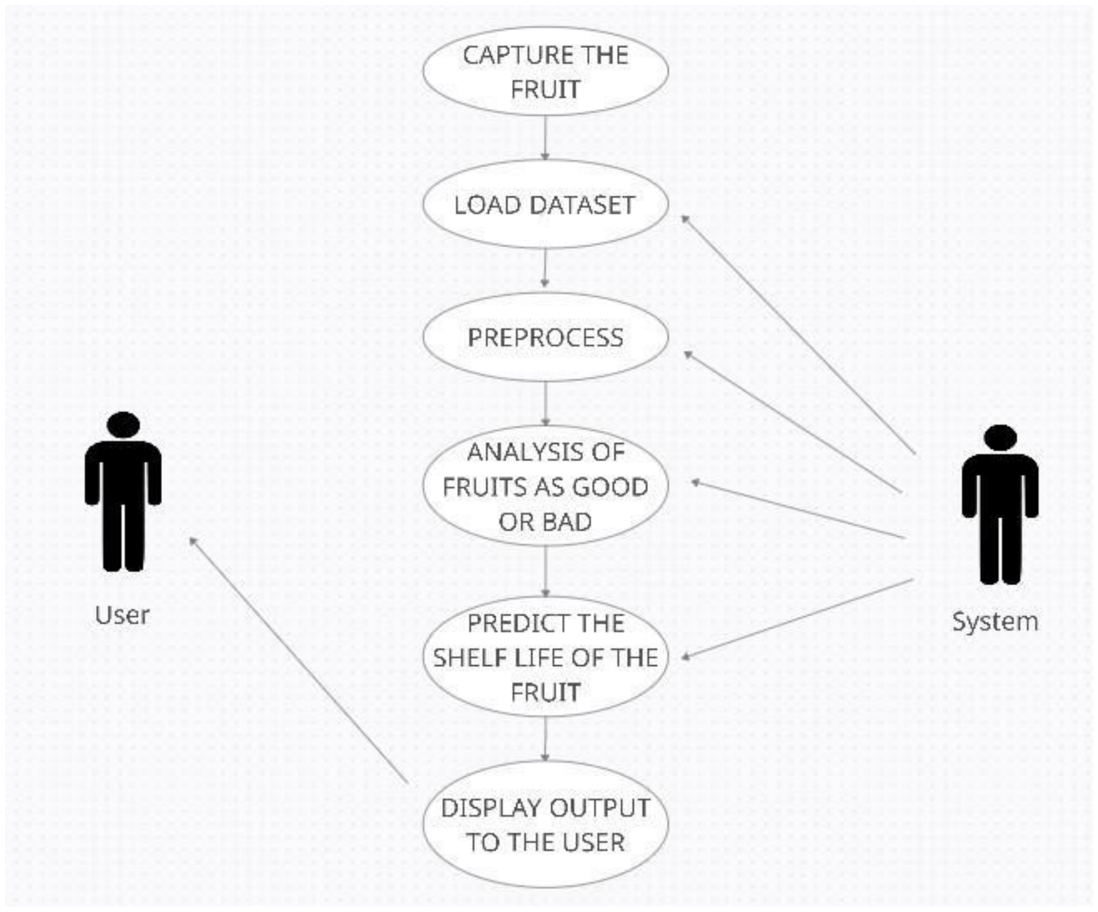

CAPTURE THE FRUIT
LOAD DATASET
PREPROCESS
ANALYSIS OF FRUITS AS GOOD OR BAD
PREDICT THE SHELF LIFE OF THE FRUIT
DISPLAY OUTPUT TO THE USER
User
System

Fig 4.4: Use a case diagram

Fig 4.4 shows the Use-case diagram. The user and the system interaction is represented in the use case diagram. The image of the fruit is captured, and the dataset is uploaded. The system interacts with the class where the dataset is uploaded, and the processing starts. The data is preprocessed where actions such as data cleaning, transformation, feature extraction, etc. are 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 given 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.

CHAPTER 5

IMPLEMENTATION

5.1 FLOWCHART

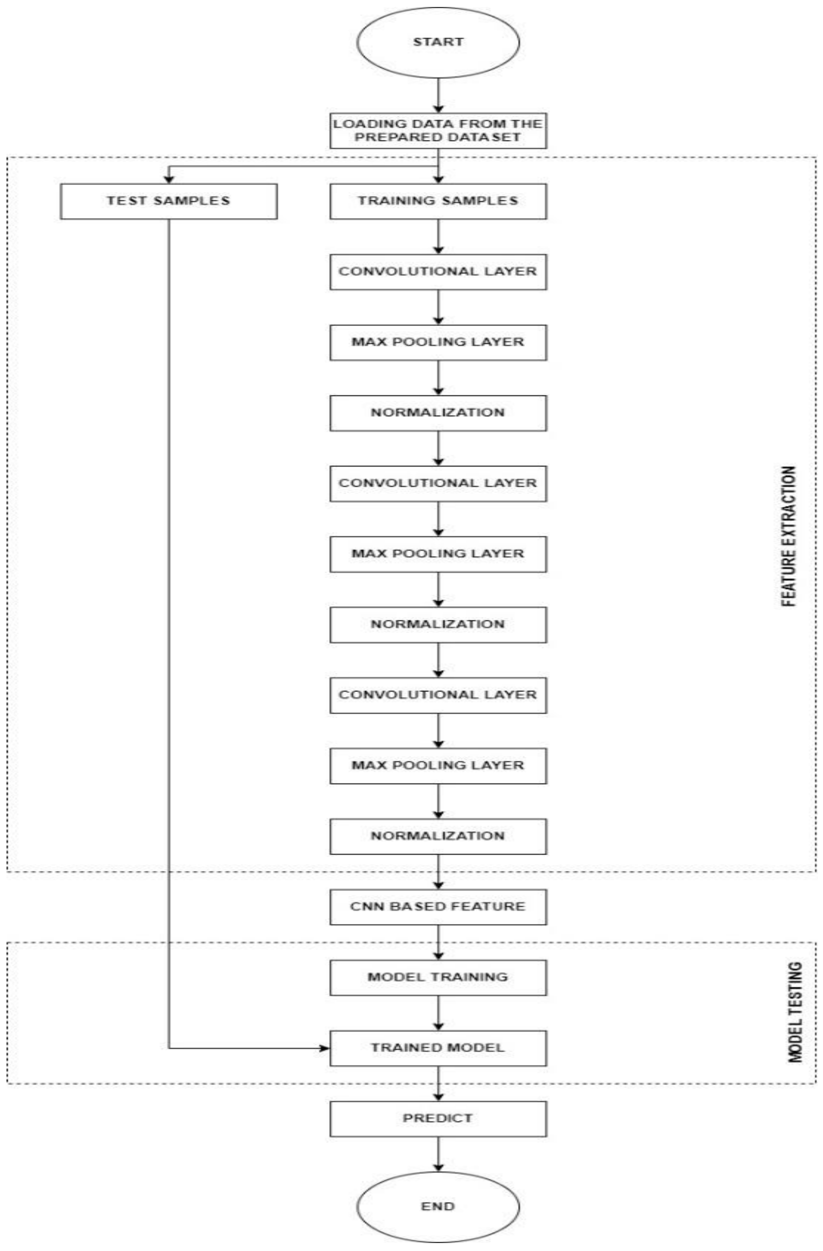


FIG 5.1 FLOWCHART

Fig 5.1. shows the flowchart of a 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 the feature extraction process, the dataset is divided into training, validation and test 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 with the testing datasets. Hence at the final stage, the model gives out the predictions.

5.2 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 mobile net 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 to 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 stacks to provide good accuracy in detecting the quality and shelf life of the fruits and vegetables.

5.3 SOFTMAX ACTIVATION FUNCTION

A vector of numbers is transformed into a vector of probabilities via the mathematical operation known as Softmax, where the probability of each value is inversely proportional to the relative scale of each value in the vector.

The softmax function is most frequently used as an activation function in neural network models in applied machine learning. The network is specifically set up to produce N values, one for each class in the classification task. The outputs are then normalized using the softmax function, changing them from weighted sum values to probabilities that total to 1. Each result in the softmax function's output is understood as the likelihood that a given class exists.

The softmax activation function transforms the raw outputs of the neural network into a vector of probabilities, essentially a probability distribution over the input classes. Consider a multiclass classification problem with N classes. The softmax activation returns an output vector that is N entries long, with the entry at index I corresponding to the probability of a particular input belonging to class.

CHAPTER 6

TESTING

6.1 TESTING

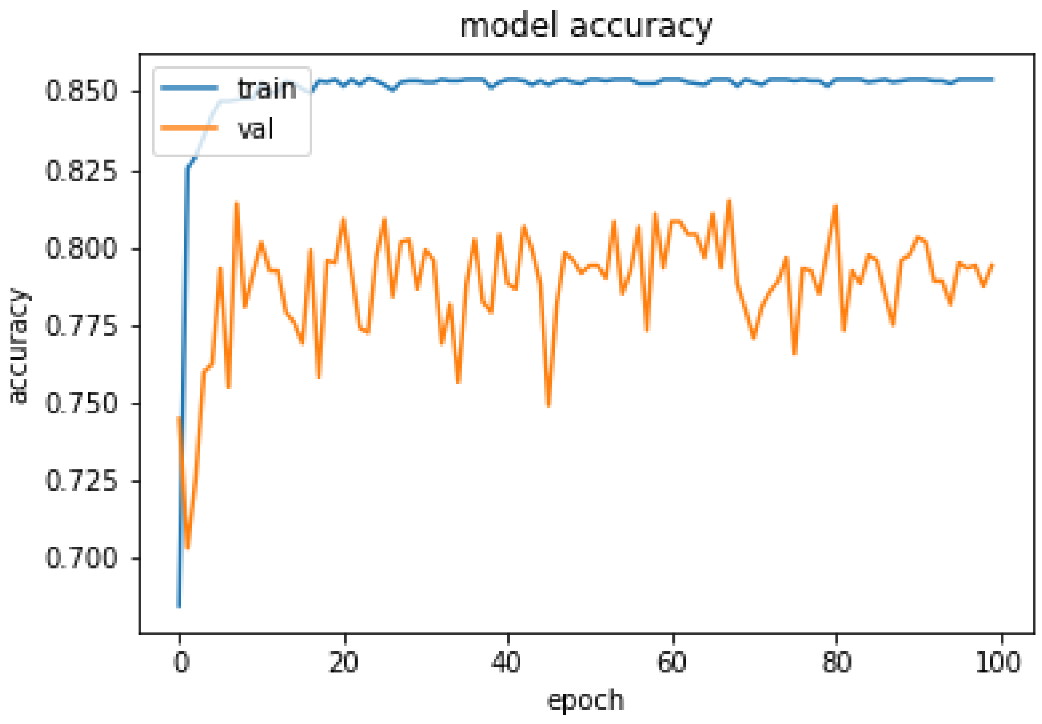


Fig 6.1: Training Accuracy graph after fine-tuning the last 7-layers

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 seen in the graph.

This indicates that the model has not been trained consistently and requires further training and modification to the layers.

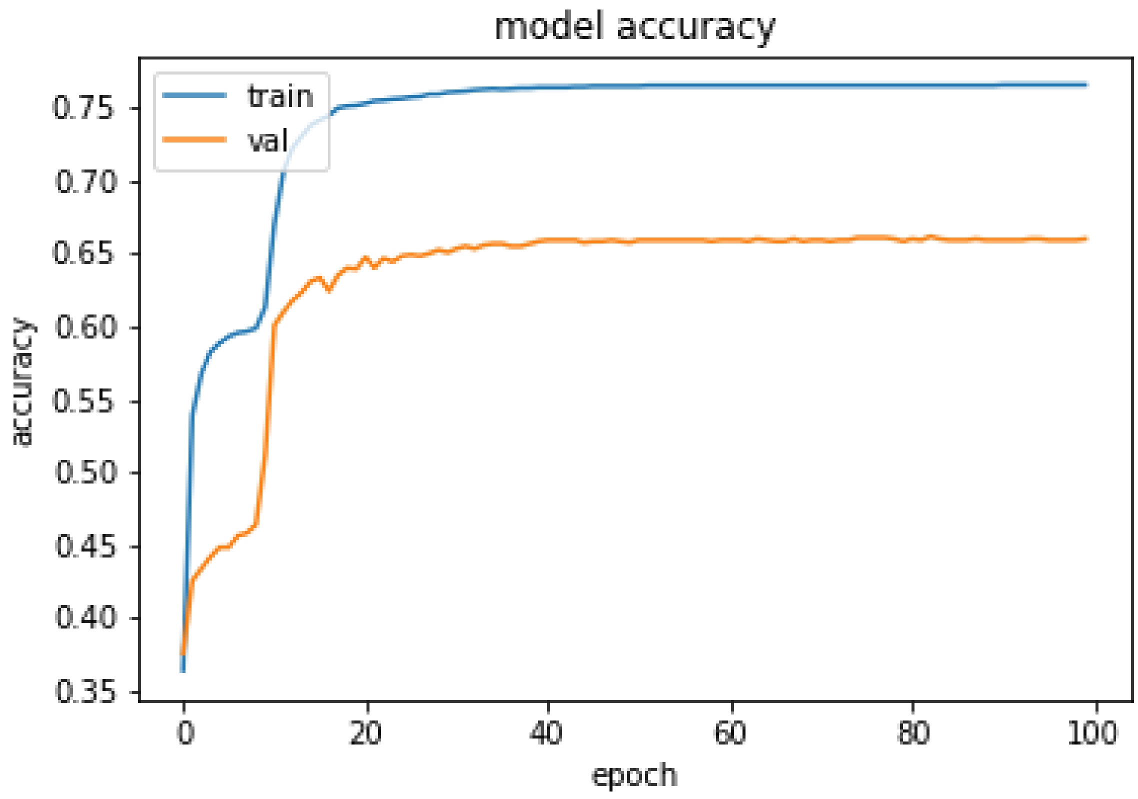


Fig 6.2: Training Accuracy graph after fine-tuning the last 9-layers

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.

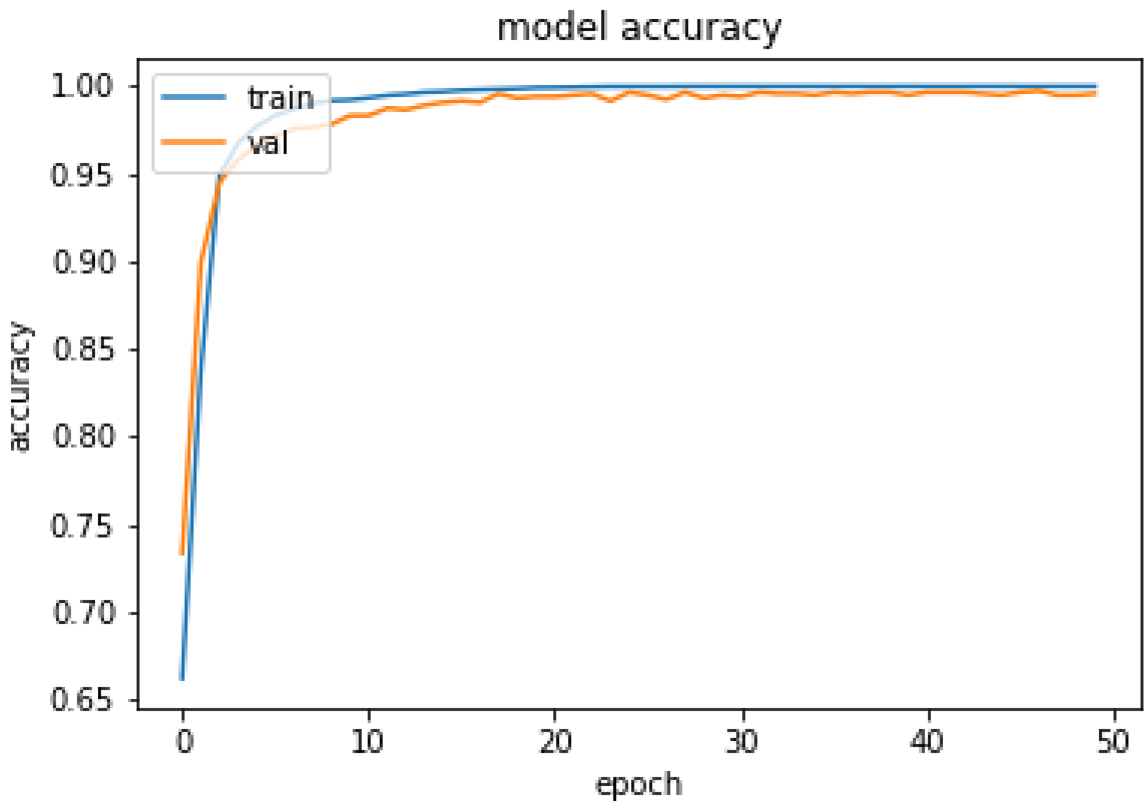


Fig 6.3: Training Accuracy graph after fine-tuning the last 5-layers

3. Fine-Tuning the Last 5 layers

Training the last 5 layers in the data set for 50 epochs leads to a 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.

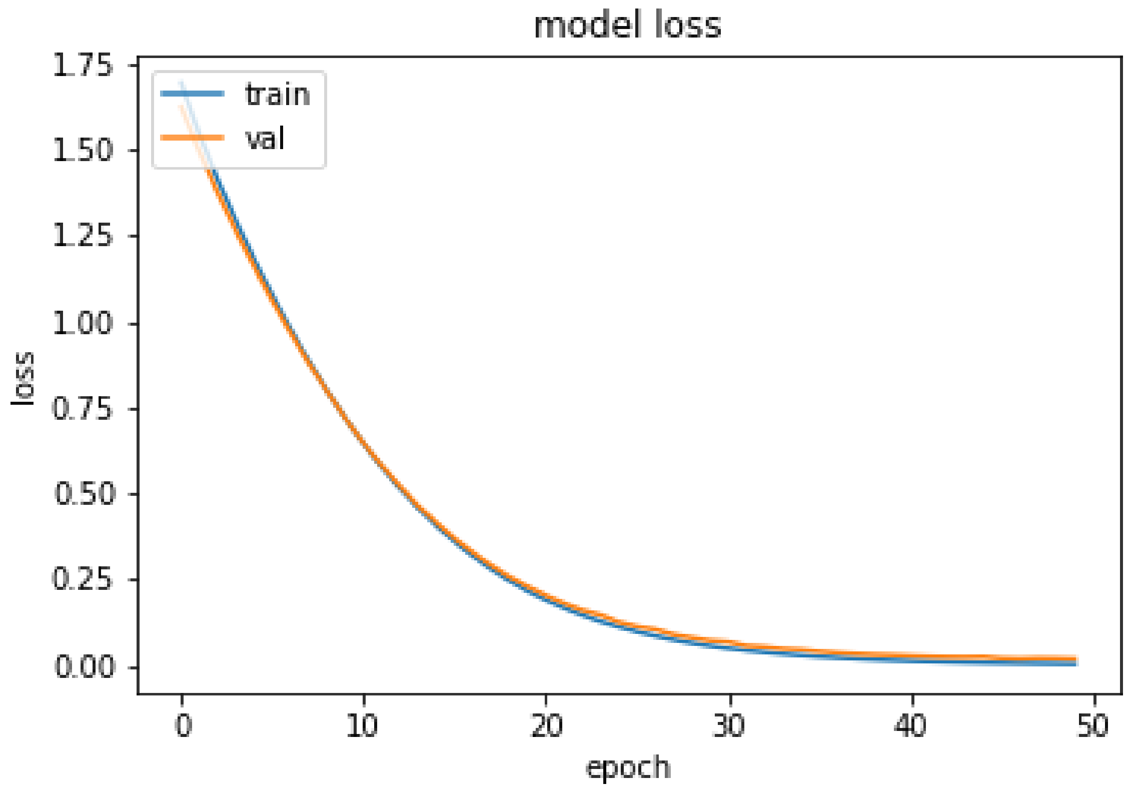


Fig 6.4: Training Loss graph of the model

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

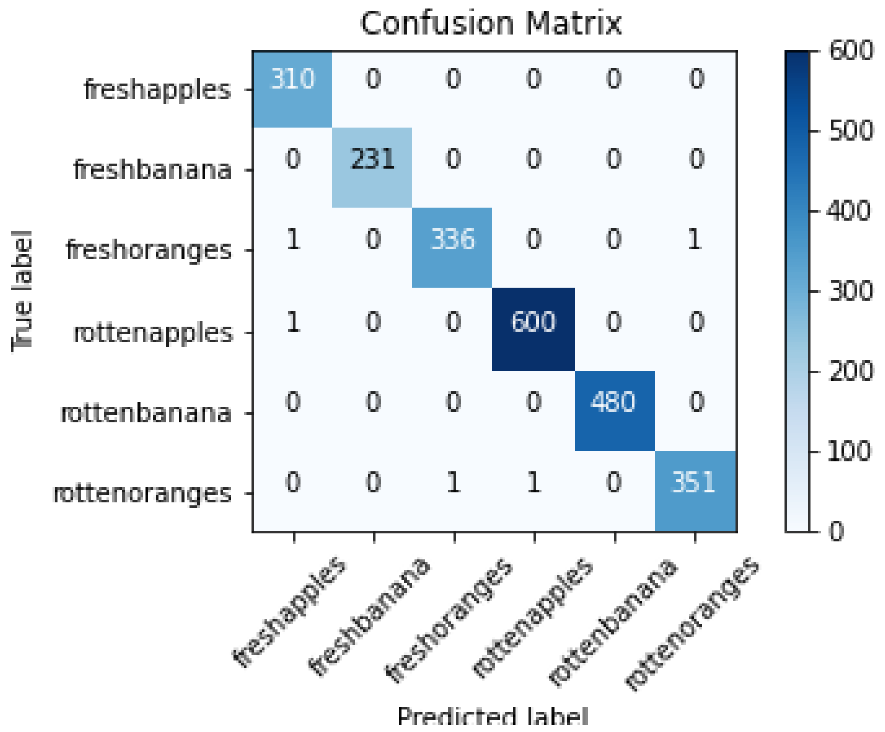


Fig 6.5: Confusion matrix of the trained model

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 to 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

RESULTS

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

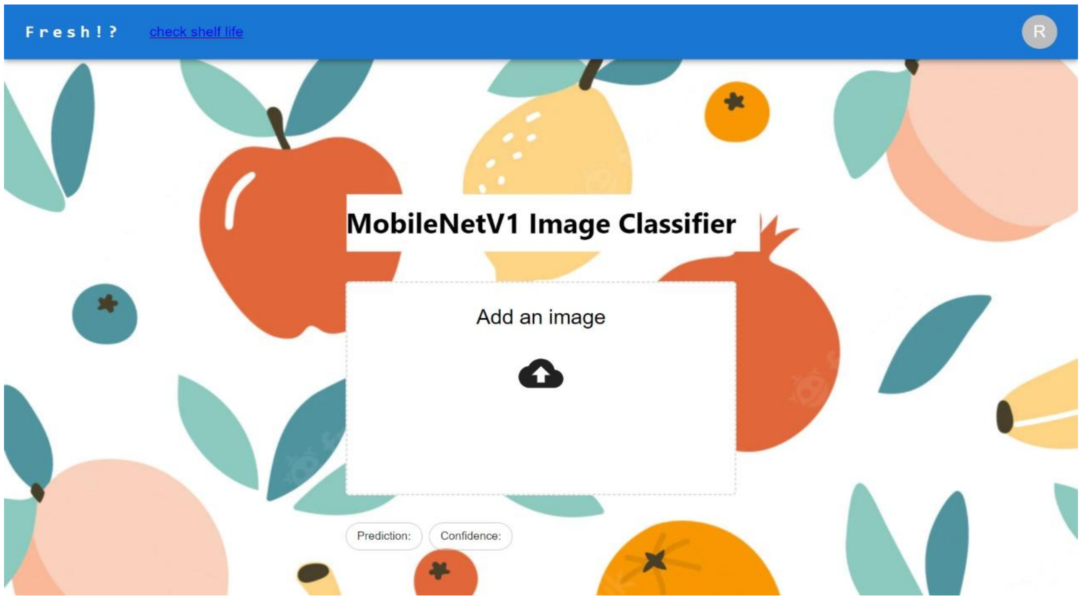


Fig 7.1: Quality Analysis of the fruit (Home Page)

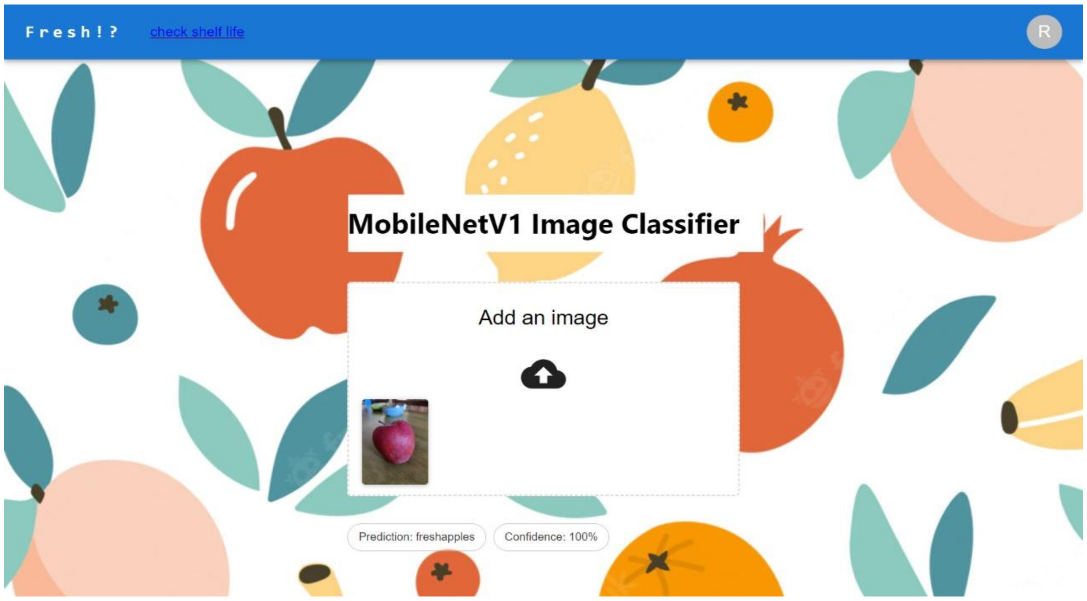


Fig 7.2: Quality of the fruit (shows a fresh apple with 100% confidence)

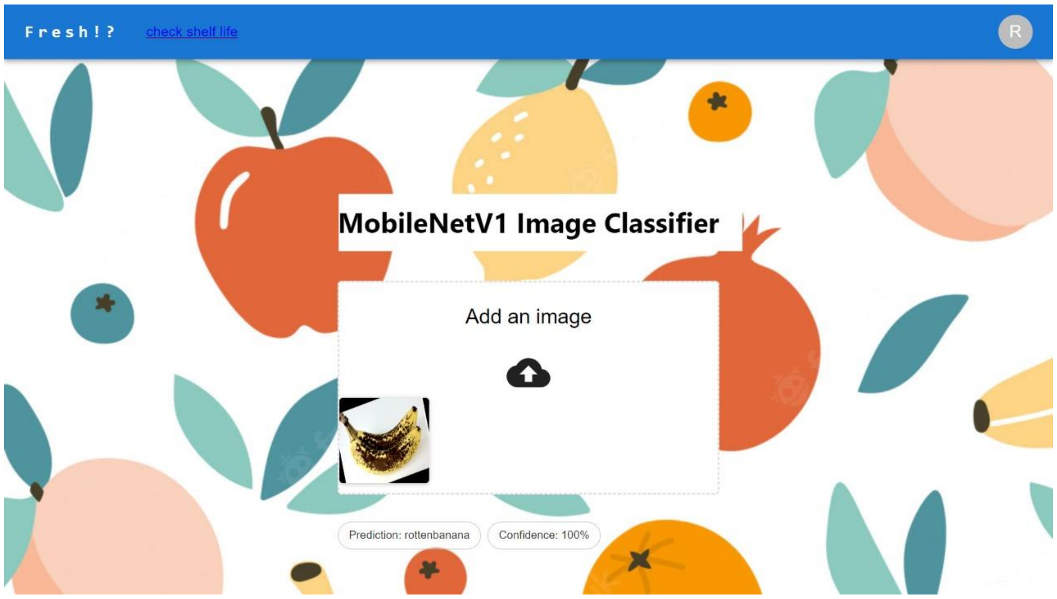


Fig 7.3: Quality of the fruit (shows rotten banana with 100% confidence)

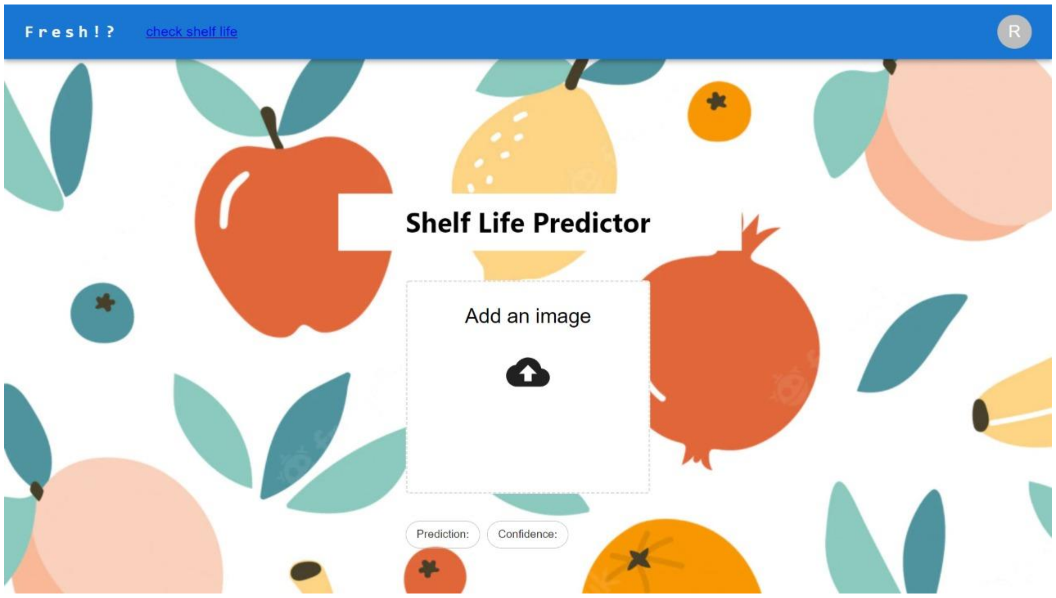


Fig 7.4: Shelf-life predictor

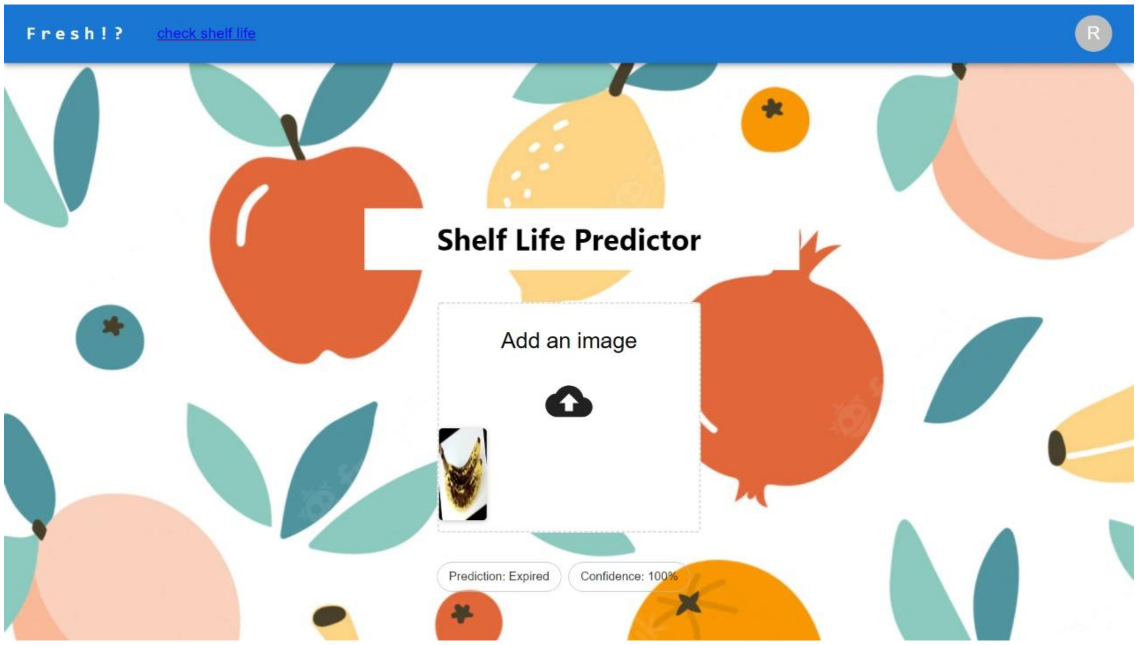


Fig 7.5: Shelf life of rotten banana fruit (shows expired with 100% confidence)

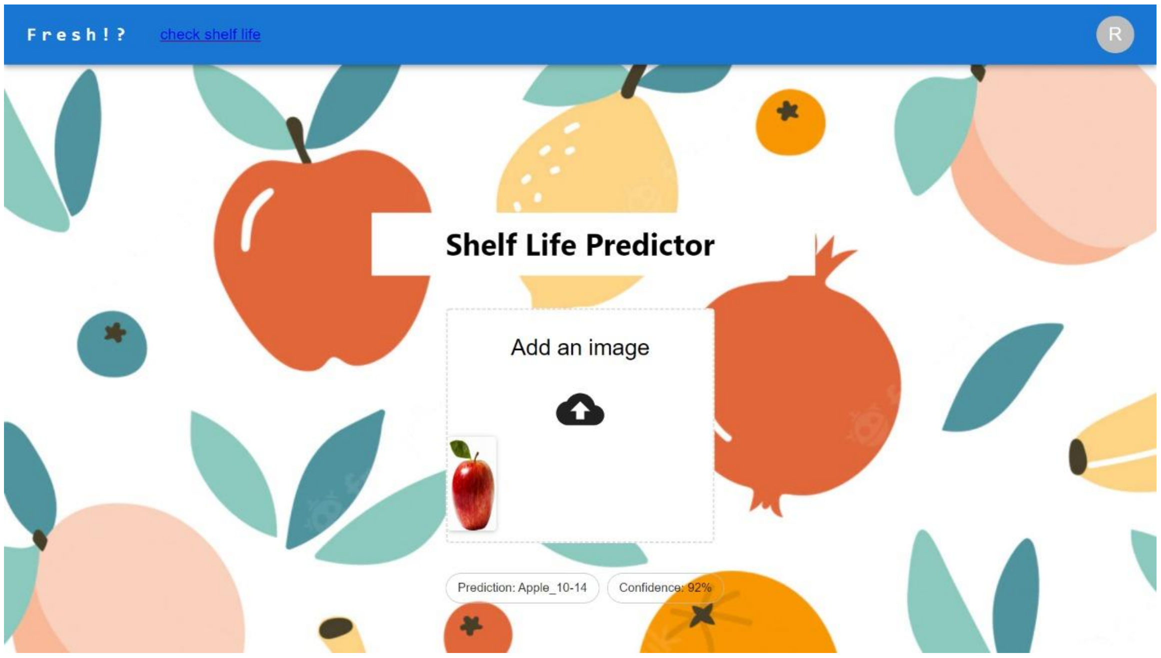


Fig 7.6: Shelf life of fresh apple fruit(10 to 14 days)

- During the training of the model, it can be seen that the model was trained by fine-tuning the last 5,7 and 9 layers. When this was evaluated with the validation and test set it can be found that the model that was fine-tuned for the last 7 and 9 layers did not perform well by showing significant loss and unstable behavior.
- However, when fine-tuned with the last 5 layers, the model can perform extremely well with minimal loss and a very good validation accuracy of 99.58%.
- After the training and deployment of the model on the website, it can be seen that the model is performing very well in the classification between fresh and rotten apple, banana and orange with real-life instance.
- It can be seen that the model can predict this is excellent accuracy ranging from 92% ~ to 100%.
- The confusion matrix derived for the model depicts the performance of the model where it can be seen that the model is able to correctly classify 2308 instances while misclassifying only 5 instances out of a total of 2313 instances.
- As can be seen from the results above, the prediction of the fruit's quality and shelf life utilizing the transfer learning model and fine-tuning categorization not only provides a greater accuracy but also can make predictions in a small amount of time compared to resource-heavy applications such as Convolutional Neural Networks (CNN’s).

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

- It can be concluded that by using a combination of transfer learning and hyper-parameter tuning the project has attained high accuracy of 99.58% for image classification and 80.79 % accuracy for shelf-life prediction.
- When implemented in cold storage facilities and industries, the model will be able to classify fresh and rotten fruits in a small amount of time and with less processing power and resources thereby saving massive amounts of time and money.
- Future plans for the image classification include training the model on a large dataset of a wide variety of fruits and vegetables so that the model will be able to cover and classify a wide range of fresh produce.
- The model will need to be trained on a huge database in the future to forecast shelf-life with a better degree of accuracy.
- To implement the model using hardware such as pi camera, raspberry pi and mechanical arms to name a few, such that it can be used in storage facilities for automated sorting and storage of items.

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, February 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] AnA F. Garcia, J. Cervantes, A. Lopez, and M. Alvarado, ‘‘Fruit classification by extracting colour 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 colour 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 colour 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 the 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.

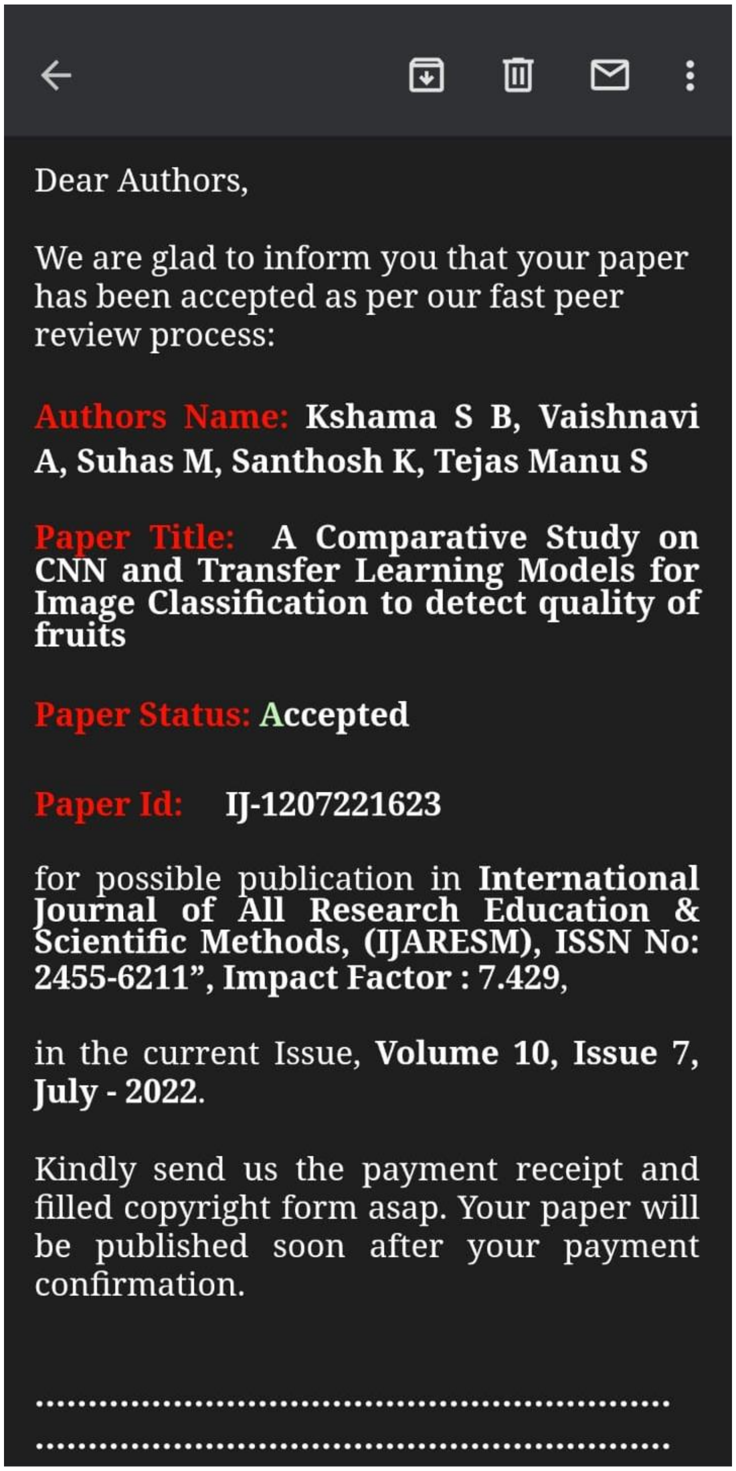
[17] X. Liu, D. Zhao, W. Jia, W. Ji, and Y. Sun, ‘‘A detection method for apple fruits based on colour and shape features,’’ IEEE Access, vol. 7, pp. 67923–67933, 2019.

[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 ACCEPTANCE 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.

...
...