Maturity Detection Techniques in Fruits

1. INTRODUCTION

- Inconsistent human harvesting practices lead to the risk of overripe or underripe fruit. Harvesting dragon fruit prematurely leads to decreased sweetness, flavour, and overall quality, potentially dissatisfying customers and reducing demand and sales, resulting in financial losses, increased labour costs, and lower prices for growers. Physical characteristics like the weight, texture, and external colour of the peel are commonly employed as non-invasive techniques for assessing the ripeness of dragon fruit [1]. Therefore, utilising the computer vision approach this dataset has the potential to develop an automated harvesting system that can empower farmers by delivering accurate advice on optimal harvest times by analysing images of various fruit development stages, consequently lowering labour requirements and minimising financial losses.

- Agriculture being the major source of revenue for the people in the rural areas in a country like India many measures have been taken to improve the revenue of the farmers. Assessing the ripeness of the fruits is also one of the primary responsibilities of the farmer to fix the proper time to glean the fruits and take them to the market. Computer vision and image processing together remains a key methodology in reducing human labour and classifying the different stages of maturity has become very prominent in examining the stages of the fruit's growth. As the colour seems to be the important feature to be observed in determining the different stages of the fruits maturity and colour identification of the tropic fruits is difficult due to the various illuminations and the partial clogging up of the fruits. In order to identify the maturity stage of the fruits the paper proposes a RGBD analysis to examine the stages of the fruits and identify the various stages of maturity.

2.PRELIMINARY

2.1 DATASET

Data Collection: This dataset comprises images that depict different phases of dragon fruit development, encompassing healthy young fruits, ripe fruits, and different decayed specimens.This dataset is crucial for training and validating our model and facilitating accurate detection of dragon fruit stages and qualities. The data was made publicly available on a research paper published in the month of

February 2024.
Reference link to the dataset: https://data.mendeley.com/datasets/2jpzbx8tm6/1

Data Augmentation:
To enhance the effect of the dataset inorder to test the model images underwent various augmentation techniques such as flipping, resizing, cropping, brightness, contrast and noise were applied to generate variations while preserving the essential characteristics of the original images, enhancing model generalisation, and reducing overfitting.

Description of the dataset:

- The dataset comprises a total of 5010 images, categorised into two classes: Mature and Immature, where the number of images in the Mature class are 2010 images and 3000 in the Immature class.

- The images are stored in separate folders for each class. Each image file is labelled, properly assigning to its corresponding class or category. Labelled data serves as the foundation for training and refining deep learning models

2.2 CLASSIFIER MODEL BRIEF:

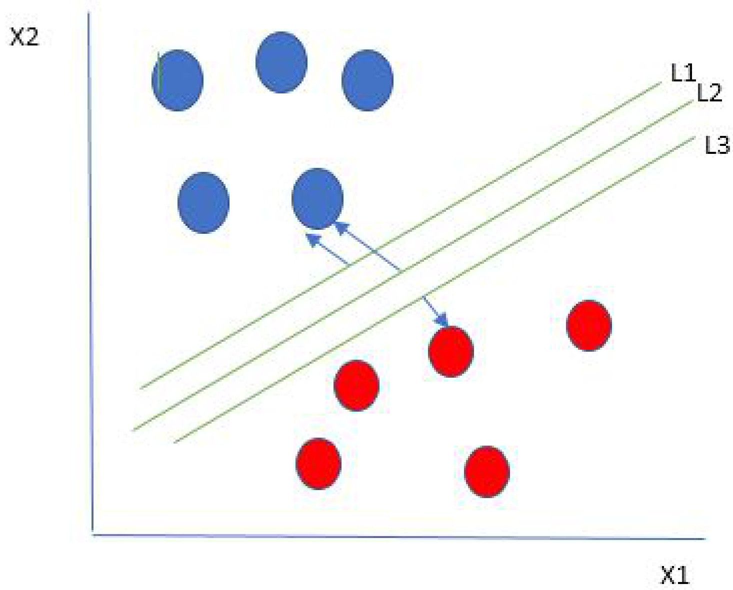
2.2.1
SVM Algorithm:

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

2.2.2

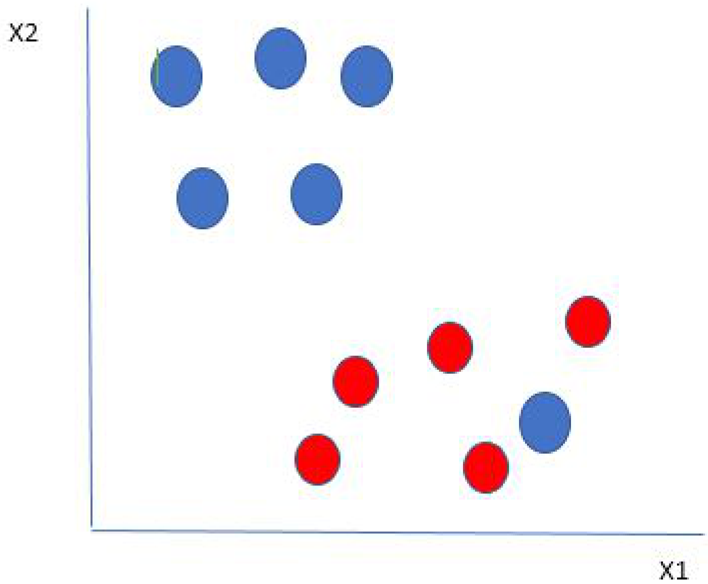
How does SVM work?

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.

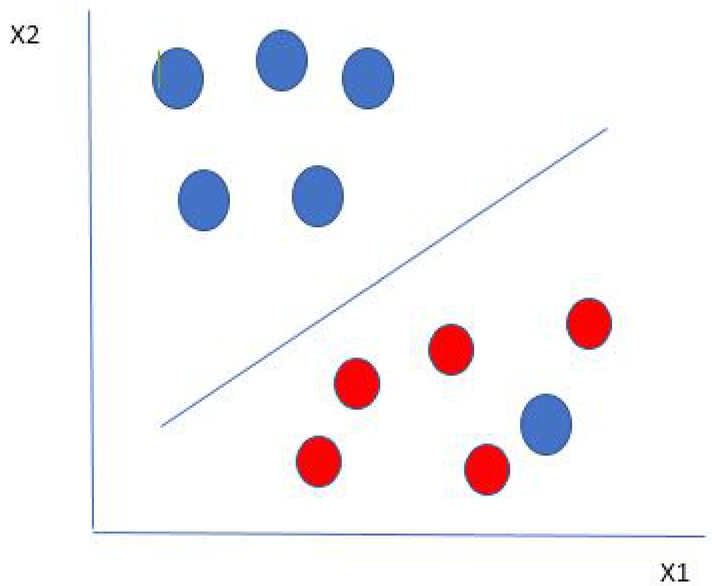


Multiple hyperplanes separate the data from two classes

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximised. If such a hyperplane exists it is known as the **maximum-margin hyperplane/hard margin**. So from the above figure, we choose L2. Let's consider a scenario like shown below

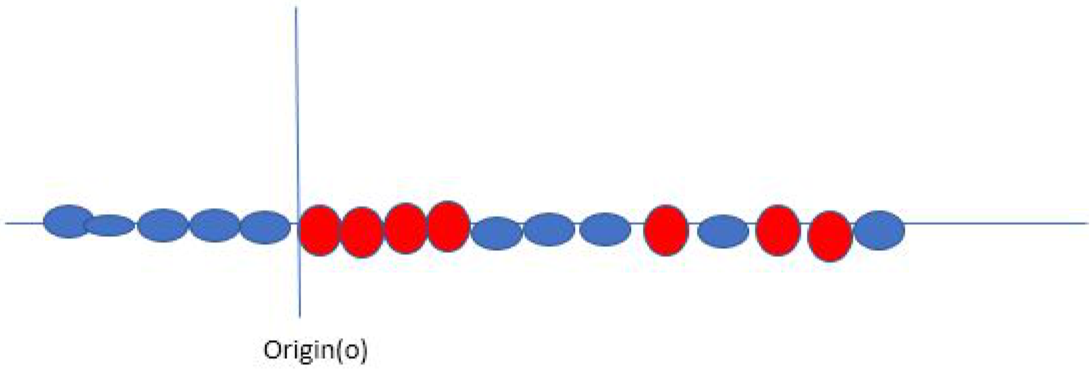


Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It's simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximises the margin. SVM is robust to outliers.

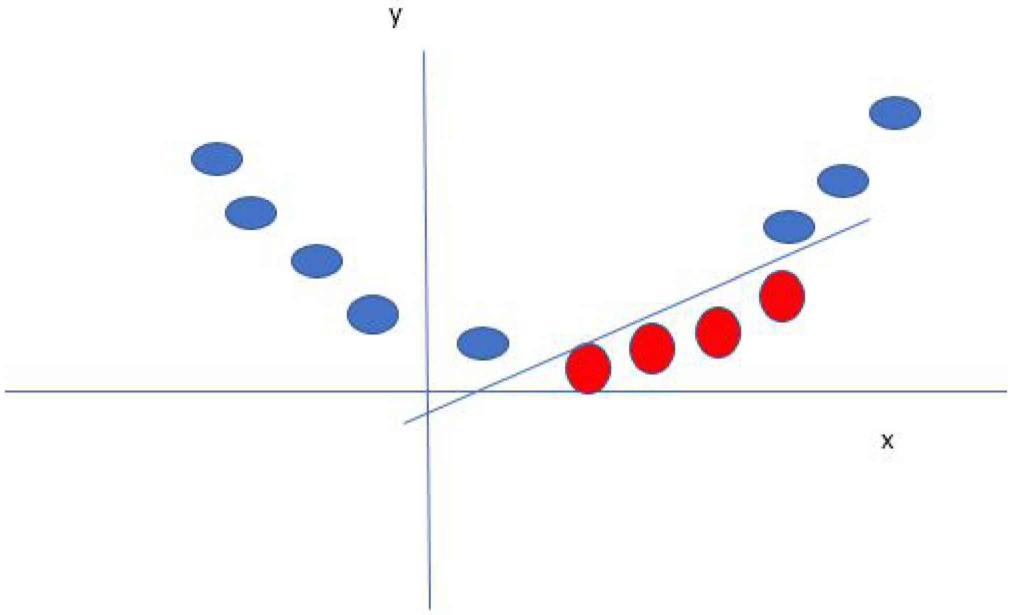


So in this type of data point what SVM does is, finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called **soft margins**. When there is a soft margin to the data set, the SVM tries to minimise $(1/margins \wedge (\sum penalty))$. Hinge loss is a commonly used penalty. If no violations no hinge loss.If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is shown in the figure above. SVM solves this by creating a new variable using a **kernel**. We call a point xi on the line and we create a new variable yi as a function of distance from origin o.so if we plot this we get something like as shown below



Mapping 1D data to 2D to become able to separate the two classes

In this case, the new variable y is created as a function of distance from the origin. A nonlinear function that creates a new variable is referred to as a kernel.

2.2.3

METHODOLOGY INVOLVED:

- The methodology proposed for classifying the Dragon Fruit "*Selenicereus Undatus*" of varied maturity grades is as,the achieved dataset of Dragon fruit images and the camera captured (acquisition) input images were utilised for executing image processing operations viz., image pre–processing, image segmentation, feature extraction for training, and testing the final fruit images. This offers distinctive features information as well as training and testing of fruit images to processing blocks of the algorithms. Then algorithm block compare the camera–captured images with the matching feature's function, inter-
linking the physical and chemical properties of fruit and analyses them into different grades using maturity stages blocks. Thus as useful feature, colour grading of fruit was employed in Python script for classifying fruit into different grades using Support Vector Machine (SVM) algorithm.

- Techniques such as K–mean clustering and SVM algorithms are used inorder to accomplish various image processing operations faster.
    -

3.LITERATURE SURVEY

[PAPER 1]

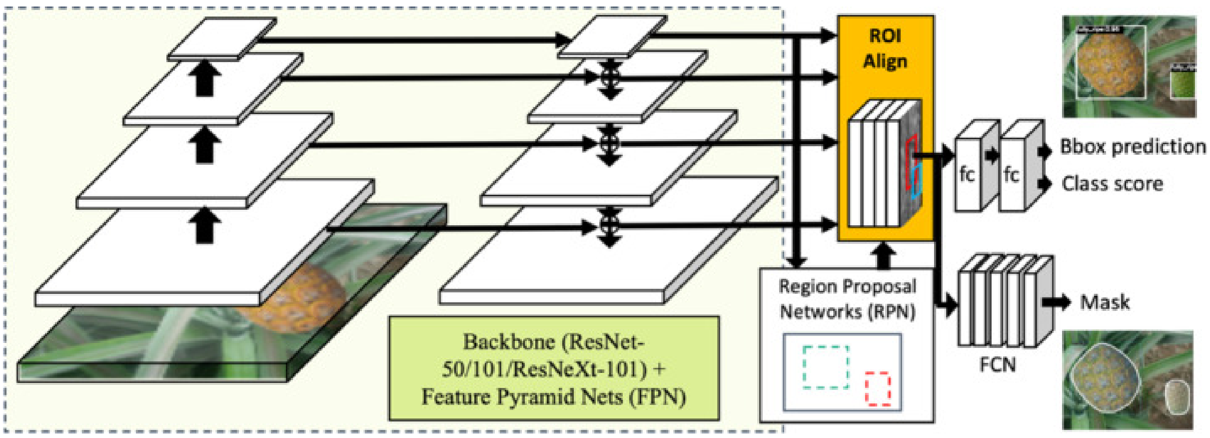
APPROACH USED:

Use A model using single image multi-object sampling and Mask R-CNN(for small datasets)

ML MODEL/ALGORITHM USED :

A Single Image Multi-Object Sampling and Mask R-CNN model for object detection is used, which is designed to address the limitations of solely relying on single-object sampling or standard Mask R-CNN frameworks. This model incorporates a novel sampling strategy that efficiently selects multiple object proposals within a single image, allowing for a broader exploration of object relationships and contextual cues. By integrating Mask R-CNN architecture, which excels in instance segmentation, our model accurately delineates object boundaries and classifies each instance within the sampled regions.

To facilitate decision-level fusion, SIMS-Mask R-CNN assigns adaptive weights to different sampling strategies and fuses the resulting object detection probabilities. This weighted fusion approach ensures a balanced integration of information from various object samples, emphasising regions with high discriminative features and improving overall detection performance.



Mask R-CNN used in our framework:

IMPLEMENTING THE CLASSIFIER MODEL:

We implement the CNN algorithms with the PyTorch framework on the Python programming language.

CONCLUSION:

This paper presents a detection and classification framework for identifying the pineapple ripeness stage using Mask R-CNN. With a small dataset, the model can be sensitive to unseen data, we introduce the optimal thresholding technique based on mAP and detection ratio for Mask R-CNN.Mean average precision of 86.7% with the ratio of ground truth and number of detections is nearly 0.9 (very close to one).

Table 6. Detection Performance of the models trained and tested with original and generated datasets (dataset #2).

Models	mAP (%)	AP50 (%)	AP75 (%)	AP_{unipe} (%)	$AP_{partially\_ripe}$ (%)	AP_{fully_ripe} (%)
CenterMask (VoVNet-99)	14.11	26.62	13.10	52.59	27.36	0.00
CenterMask (VoVNet-57)	30.19	50.86	37.19	50.16	29.72	73.00
Faster R-CNN (ResNet-101)	87.81	93.62	93.62	97.46	87.83	95.89
Faster R-CNN (ResNet-50)	77.72	96.67	93.79	98.52	94.31	97.65
RetinaNet (ResNet-101)	**91.13**	94.21	94.21	99.19	88.02	95.65
RetinaNet (ResNet-50)	90.11	93.09	93.09	99.73	89.96	89.90
Mask R-CNN (ResNet-101)	86.70	**97.98**	**97.98**	99.20	**96.58**	**98.63**
Mask R-CNN (ResNet-50)	82.81	94.76	94.76	97.18	89.36	98.00
Mask R-CNN (ResNeXt-101)	82.98	96.68	96.68	**99.88**	94.11	96.37

PAPER 2

APPROACH USED:

A Model using Artificial Neural Network to detect maturity in fruits

ML MODEL:

ANNs are virtually like NNs therein they contain interconnected PEs (Processing Elements) and are only sanctioned to receive information supplied locally.

ANNs are a kind of parallel computing, which differ from conventional computers within the way they process information. The operation of conventional computers is controlled by one central processing unit (CPU), which holds the computer's recollection and sequentially processes information. Parallel computers, on the antithesis hand, contain several more minute processing elements (PEs) that are linked together. As a result, the computer's recollection is distributed and cognizance is processed in a parallel manner.
These layers are referred to as "hidden" since they're not visible as a network output. The PEs within the sundry layers are either plenarily or partially interconnected. The connections between the PEs are weighted. The vigour of every connection weight is often adjusted; a zero weight represents the absence of a connection, and a negative weight represents an inhibitory relationship between two PEs. The propagation of cognizance through the network commences with the presentation of an input stimulus at the input layer.

The data then permeate and are operated on by the network until an output stimulus is engendered at the output layer (Figure 2).In operation, and each unit of an ANN receives input from other connected units a weighted sum of the input is computed at a given instance of your time. The activation value determines the specific output from the output state of the unit.

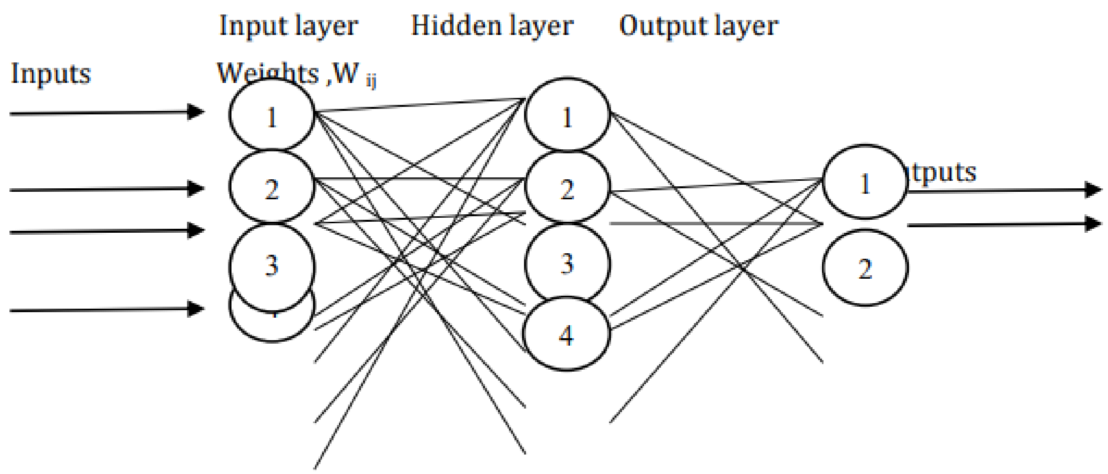


Figure: 2 Basic Layers of ANN

MINIMUM SOFTWARE SPECIFICATION:

a) OPERATING SYSTEM: WINDOWS 8.1

b) FRONT END : MATLAB R2013a

CONCLUSION :

- The technique is started by capturing the fruit’s image using a regular camera or any mobile camera. The features are efficiently extracted from the trained image.
- The extracted features are supported by the parameter's colour, shape, and size. The ANN technique is employed for disease detection. The standard is decided by using fruit features obtained with the assistance of ANN.
- The proposed technique detects the standard of fruits better than the result produced by the prevailing technique. In our project, the identification of excellent and bad fruits supported quality in image processing using MATLAB is completed successfully with more accuracy.
- The utilisation of image processing for identifying the standard is often applied not only to any particular fruit, but we will also apply this method to spot the standard of vegetables with more accuracy.
- Thus, this may enable the technology to be applied to several products. The performance was also found to be reasonably good as compared with human experts and other work.

PAPER 3:

APPROACH:

Tomato Fruit Ripening Classification Using Wavelet-Based Feature Extraction and Multilayer Perceptron

ML MODEL:

Discrete wavelet transform (DWT) and multilayer perceptron (MLP) as the feature extraction method and classifier

A multilayer perceptron (MLP) classifier is an artificial neural network used for supervised learning tasks, primarily in classification problems. It is a feedforward neural network composed of multiple layers of interconnected nodes. MLP classifiers are versatile and can be applied to various classification tasks, including image classification. [11] proposed the MLP-Mixer architecture, an all-MLP approach for image classification. The model replaces traditional convolutional layers with MLPs and mixer layers, demonstrating competitive results on various image classification benchmarks. ResMLP [12] is another MLP-based architecture incorporating a hierarchical structure similar to ResNet. The research demonstrated that ResMLP achieved competitive results on image classification tasks, especially with limited training data. Therefore, this study uses discrete wavelet transform (DWT) and multilayer perceptron (MLP) as the feature extraction method and classifier.

Multilayer Perceptron (MLP) is a fully connected multilayer neural network. An MLP is a typical example of a feedforward artificial neural network. MLP Classifier relies on an underlying Neural Network to perform the task of classification.

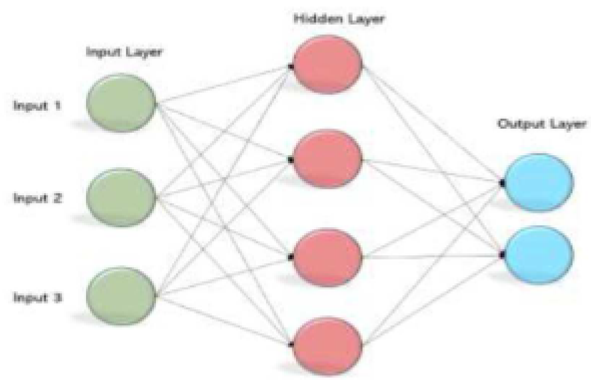


Fig. 8. Architecture of Multilayer Perceptron Classifier

CLASSIFICATION:

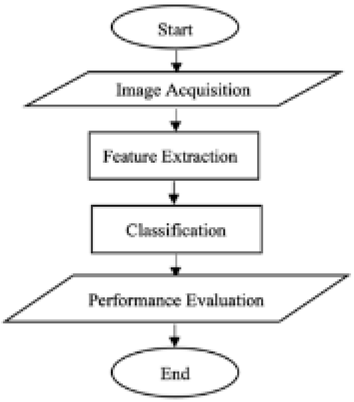


Fig. 1. Phases of Tomato Ripening Classification

CONCLUSION:

This paper proposed the discrete wavelet transform method combined with multilayer perceptron classifier to classify tomato ripeness. Generally, tomatoes' colour, size and shape features play an important role in determining the fruit's quality, grade and ripeness. Since that, those features with different maturities and grades were used as an indicator for tomato classification and obtained using discrete wavelet transform techniques. The mean, variance, skewness, and kurtosis values were retrieved from the approaches, and the tomato classification was performed using a Multilayer Perceptron (MLP) classifier. It was concluded that using a higher DWT decomposition level will not increase the classification performance of all tomato classes. Therefore, for future research, it is suggested to do feature selection before inputting the features into the MLP classifier.

Tomato Class	Precision	Recall	F-Score
Super grade	0.71	1.00	0.83
First grade	0.92	0.92	0.92
Second grade	0.67	0.44	0.53
Unqualified	0.50	0.60	0.55
Damage	1	1	1

PAPER 4:

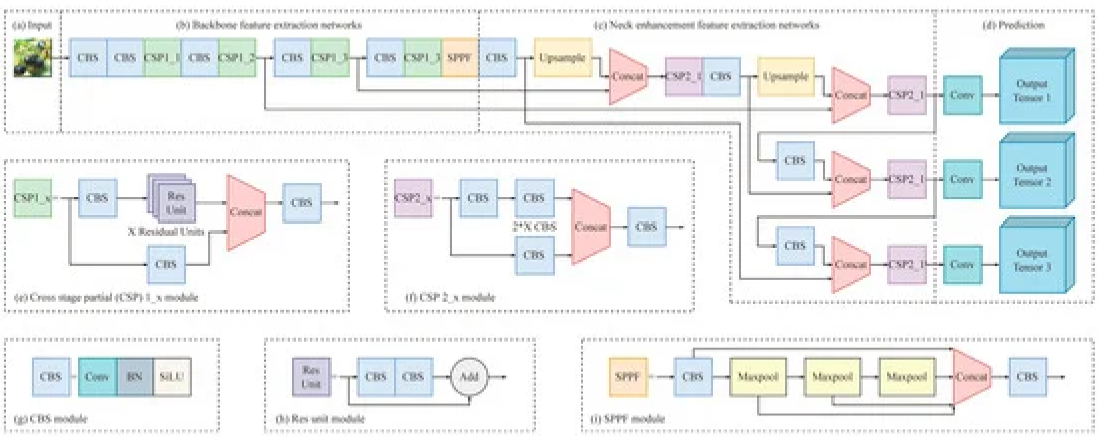
APPROACH:

A Lightweight Detection Method for Blueberry Fruit Maturity Based on an Improved YOLOv5 Algorithm

ML MODEL:

YOLOv5 , CBAM, FASTER R-CNN, SiLU, PANet, FPN

The YOLOv5 algorithm is one of the algorithms in the YOLO series [18]. It is an improvement based on the YOLOv4 algorithm. The YOLOv5 algorithm consists of 10 detectors: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and others. The main differences among them lie in the number of convolutional layers and optimal application scenarios. As the number of convolutional layers increases, the model size gradually increases, while the detection performance improves and the detection speed decreases. This research focuses on the YOLOv5s 7.0 model as the subject. The overall network structure of the YOLOv5s 7.0 model can be divided into four parts: the input layers, the backbone feature extraction networks, the neck enhancement feature extraction networks, and the output layers



It mainly consists of four parts: the input layers, the backbone feature extraction networks, the neck enhancement feature extraction networks, and the output layers. The input layers of the improved YOLOv5 algorithm accept blueberry fruit images with the size of 608 × 608 pixels.

Training deep convolutional neural networks requires a large amount of data [16]. Too little data can result in underfitting or overfitting of deep convolutional neural networks. Therefore, data augmentation needs to be performed on the original blueberry fruit images, this research employs various methods, including mirroring, rotation, scaling, adding noise, and adjusting brightness, to enhance the diversity of the blueberry fruit images that we have collected.

CONCLUSION:

The experimental results demonstrate that the improved YOLOv5 algorithm can effectively utilise RGB images to detect blueberry fruits and recognize their ripeness. The improved YOLOv5 algorithm achieves a *P* of 96.3%, an *R* of 92%, and a *mAP* of 91.5% at a threshold of 0.5. Compared to other detection algorithms such as YOLOv5, SSD, and Faster R-CNN, this method has a smaller model size, smaller network parameters, lower memory usage, lower computation usage, and faster detection speed while maintaining high detection performance.

Table 2. Performance comparison of the various blueberry fruit detection algorithms.

Metrics/Models		YOLOv5	YOLOv5-Ours	SSD-vgg	Faster R-CNN-vgg
P (%)	mature	98.7	97.8	96.0	93.1
	semi-ripe	95.5	96.3	92.7	87.1
	immature	97.0	94.9	96.2	85.6
	mean value	97.1	96.3	95.0	88.6
R (%)	mature	93.5	92.9	96.0	95.8
	semi-ripe	91.3	90.1	89.0	90.1
	immature	93.4	93.0	93.9	93.0
	mean value	92.7	92.0	93.0	93.0
mAP@0.5 (%)	mature	95.1	93.7	95.9	95.6
	semi-ripe	91.0	88.8	88.0	89.1
	immature	93.5	91.9	92.5	91.0
	mean value	93.2	91.5	92.1	91.9
Model size (MB)		13.6	5.65	91.6	521.0
Parameter (M)		7.02	2.85	23.6	136.7
FLOPs (G)		15.8	5.6	246.6	376.5
Speed (fps)		66.2	67.1	44.4	17.0

PAPER:5

APPROACH

Dragon Fruit Maturity Detection Based-HSV Space colour

ML MODEL:

Naive Bayes Classifier method

RGB to HSV method.

- Colour is one of the elements to determine the level of maturity of the fruit because each level of maturity of the fruit has a different colour level. At present, the dragon fruit is only classified based on a visual analysis of the skin colour of the human eye. This study aims to classify dragon fruit using image processing with the naïve Bayes method based on HSV colour space.
- The colour feature used in this study is RGB which has been converted to an HSV value, after getting an HSV value then classified using the naïve Bayes method. Image data used amounted to 120 training images and 30 test images.
- The results of the classification of dragon fruit using the naïve Bayes method based on HSV colour space can show an accuracy rate of 86.6%.

The implementation of the dragon fruit maturity detection system based on skin colour in the HSV colour space with classification using the naïve Bayes method is shown below.

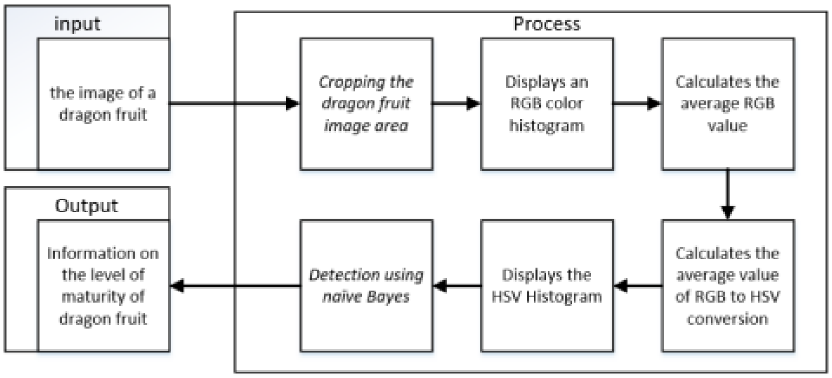


Figure 1. Flowchart of maturity determination program based on HSV color space

CONCLUSION

Through a robust automated system for dragon fruit maturity detection, employing image processing techniques and the naïve Bayes method based on the HSV colour space. Through testing and analysis, the model achieved an accuracy rate of 86.6% when processing 30 images, each representing different stages of fruit ripeness.

The HSV values for;

- Unripe: Minimum:((0.19355; 0.51748; 0.42190) and Maximum (0.27481; 0.85481; 0.70850)
- For Half-ripe: Minimum (0.16776; 0.26961; 0.54466) and Maximum(0.82015; 0.62398; 0.96775)
- For Ripe: Minimum (0.55273; 0.35087; 0.60747) and Maximum (0.93848; 0.70491; 0.98491).

Table 7 Results of the classification of dragon fruit

		Prediction Class			Total
		Unripe	*Half Ripe*	*Ripe*	
Actual Class	*Unripe*	9	1	0	10
	Half Ripe	1	8	1	10
	Ripe	0	1	9	10
Total		10	10	10	30

Then the accuracy of the system can be calculated in a way

$$\text{Accuracy} = \frac{26}{30} x100\% = 86,6\%$$

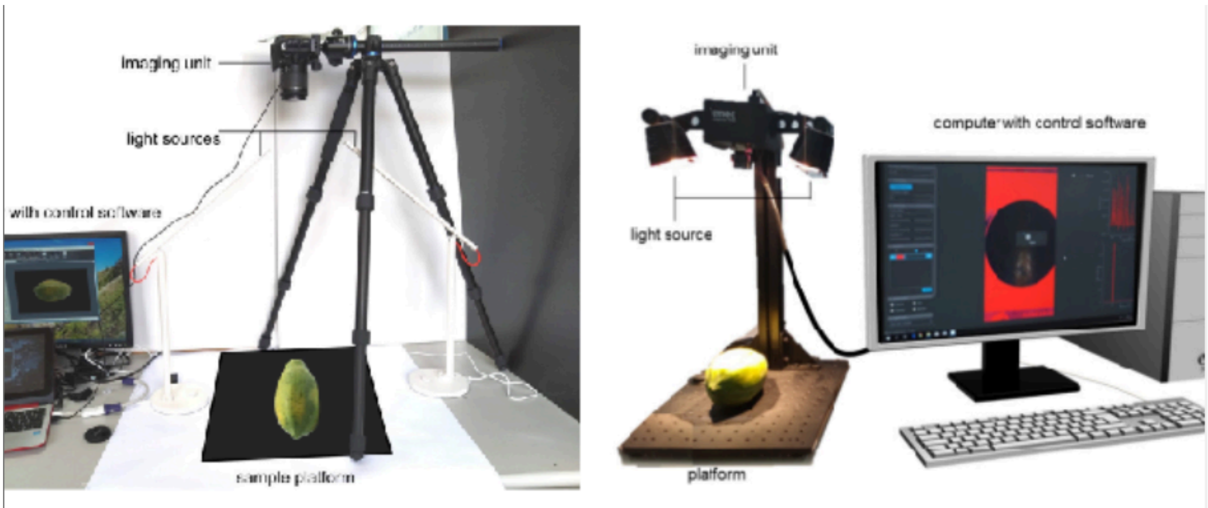
PAPER 6:

APPROACH

Visible-Light and Hyperspectral Imaging for Fruit Maturity Estimation

ML MODEL

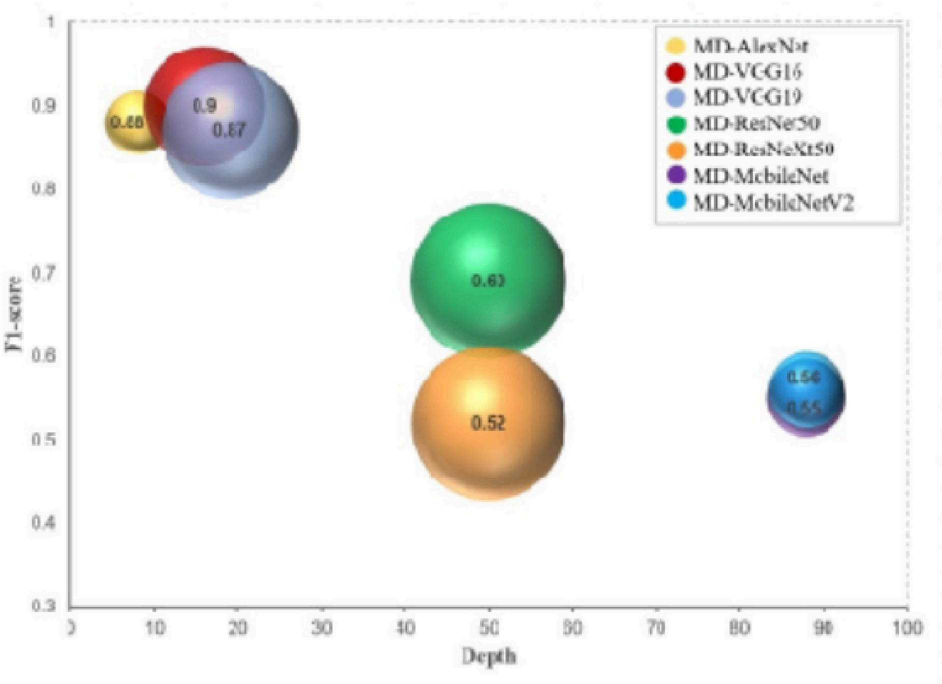
Visible-light imaging captures images based on the colours and visual characteristics of objects, while hyperspectral imaging provides detailed spectral information across a wide range of wavelengths, enabling more comprehensive analysis and characterization of materials.
Maturity stages of papaya fruit samples are defined based on the visual characteristics of the peel colour (papaya stages, description, number of RGB images, and hyperspectral (HS) data cubes) . Two hundred fifty-three (253). At least eight RGB images (one from each side: front, back, left, right; diagonal orientation of each side) and at least two hyperspectral images (two to four sides) per sample were acquired. All samples were classified into six maturity stages: green with a trace of yellow (MS1), more green than yellow (MS2), mix of green and yellow (MS3), more yellow than green (MS4), fully ripe (MS5), and overripe (MS6).
Faster R-CNN and random forest, respectively, while 15,000 images were acquired in [47], which also used Faster R-CNN. [30,48,49] collected and used 557, 300, and 240 HSI images for classification using deep learning methods: ResNeXt, GAN, and AlexNet, respectively. Thus, the number of samples, RGB and HS images gathered in this study, is by far among the highest based on recently published studies.



(a) visible-light imaging setup (b) hyperspectral imaging setup

Conclusion:

This study had developed multimodal variants of deep learning models for a novel non-destructive, refined fruit maturity stage classification. To build the database of multimodal input, hyperspectral imaging and visible light imaging acquired the hyperspectral data cubes and RGB images of papaya fruit samples, respectively, at six maturity stages from unripe stage to overripe stage.the models utilized the multimodal input data cubes and exhibited very promising results in fruit maturity estimation with up to 0.90 F1-score and as low as 1.45% top-2 error rate. Among the seven architectures, MD-VGG16 obtained the best performance, while MD-AlexNet and MD-VGG19 showed comparable outcome.



PAPER 7 :

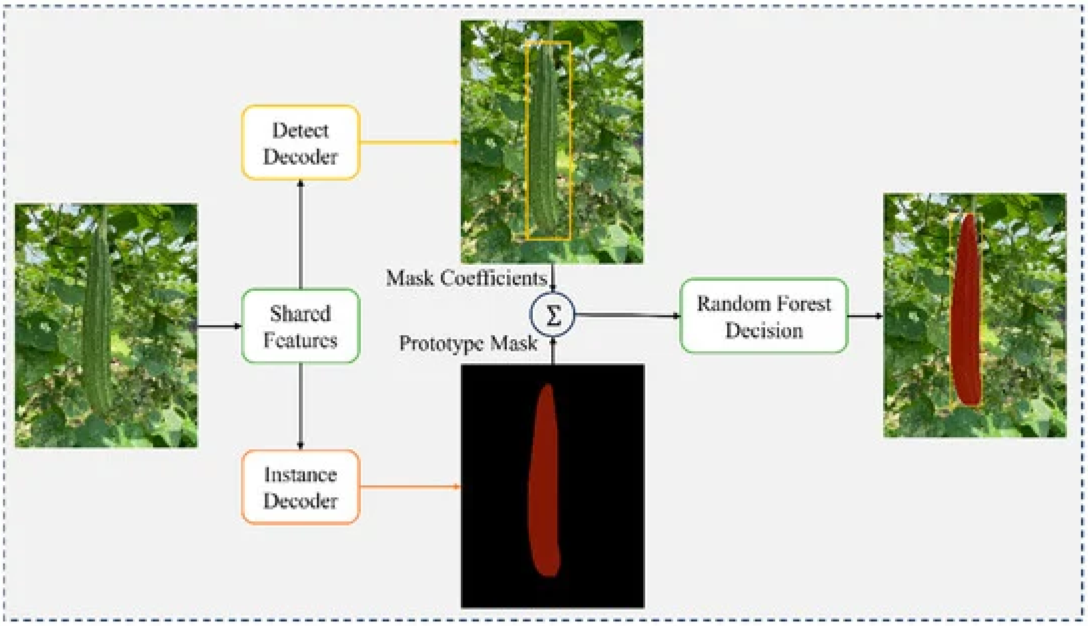
APPROACH

Maturity Classification Method for Loofah employing EdgeNeXt and an enhanced pyramid attention-based feature pyramid network (PAFPN)

ML MODEL

LuffaInst, a one-stage instance segmentation model, and a machine learning-based maturity classification model. LuffaInst employs a lightweight EdgeNeXt as the backbone and an enhanced pyramid attention-based feature pyramid network (PAFPN). To cater to the unique characteristics of elongated loofah fruits and the challenge of small target detection, we incorporated a novel attention module, the efficient strip attention module (ESA), which utilises long and narrow convolutional kernels for strip pooling.
The mean average precision (mAP) on the loofah image dataset has improved and the FPS increased by at least 10.13 f/s compared with Mask R-CNN, Mask Scoring R-CNN, YOLACT++, and SOLOv2, thereby satisfying the real-time detection requirement.

The study comprised of two features: The LuffaInst instance segmentation model and the random forest decision model, the LuffaInst's detection branch decoder and instance segmentation decoder leverage shared feature maps to achieve their respective tasks, thus ensuring the accurate detection and segmentation of the loofah fruit.



We use EdgeNeXt-small as the backbone network, where Pn on the right represents the output feature maps of each respective layer, with output dimensions of 96 × 80 × 80 (P1), 160 × 40 × 40 (P2), and 304 × 20 × 20 (P3). We merge the downscaled output feature maps from these three layers to generate a 560 × 10 × 10 (P4) fourth-layer output feature map.

Table 3. The evaluation metrics of Mask R-CNN, Mask Scoring R-CNN, YOLACT++, SOLOv2, and LuffaInst on the loofah image dataset.

Models	mAP (%)	Parameters	GFLOPs	FPS	Times Per Image (ms)
Mask R-CNN	91.00	43.971 M	1.472 T	9.5	105.31
Mask Scoring R-CNN	89.60	60.656 M	2.874 T	27.34	36.65
YOLACT++	87.00	35.21 M	0.247 T	28.5	35.1
SOLOv2	87.5	46.22 M	0.139 T	17.31	58.2
LuffaInst	**94.20**	**11.91 M**	0.205 T	**38.63**	**25.79**

CONCLUSION:

In this study, we proposed a method for detecting and classifying the maturity of the loofah with the aim of providing a solution for loofah fruit maturity detection in the efficient and intelligent harvesting technology of the loofah. Compared with MaskR-CNN, Mask Score R-CNN, YOLACT++, and SOLOv2, LuffaInst not only maintains the model size but also achieves a higher accuracy of 94.2% compared with other popular instance segmentation models.

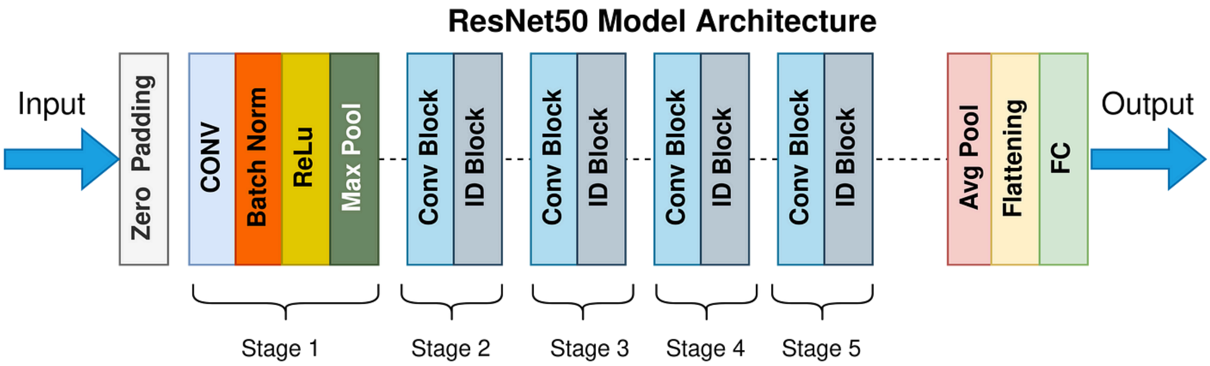
PAPER 8:

APPROACH

Machine learning techniques such as Deep Convolutional Neural Network(CNN)

ML MODEL

- In this research, we have applied the ResNet50 framework with the intention of identifying the ripeness and quality of dragon fruits. ResNet50 is a commonly employed deep convolutional neural network (CNN) architecture that is renowned for its effectiveness in image classification and object detection.
- It is characterised by its depth, consisting of 50 convolutional layers, which enables it to learn intricate features from images. Its deep structure and residual connections contribute to its ability to achieve state-of-the-art results in a variety of computer vision applications.



In ResNet50, batch normalisation is applied after each convolutional layer and before the activation function (e.g., ReLU), ensuring that the inputs to subsequent layers are well-scaled and centred. ReLU introduces non-linearity into the network by replacing negative values with zeros.

Measuring Metrics

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (1)$$

Precision: Precision indicates how accurately the model's optimistic predictions came true. It measures the proportion of real positives to all anticipated positives.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

Recall: The model's capacity to recognize all pertinent instances in the dataset is measured by recall, also known as sensitivity or true positive rate. It measures the proportion of real positives to all actual positives.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

F1-Score: The harmonic mean of recall and precision is known as the F1-Score. When you need to take into account both false positives and false negatives, it provides a balance between these two measures and is particularly helpful.

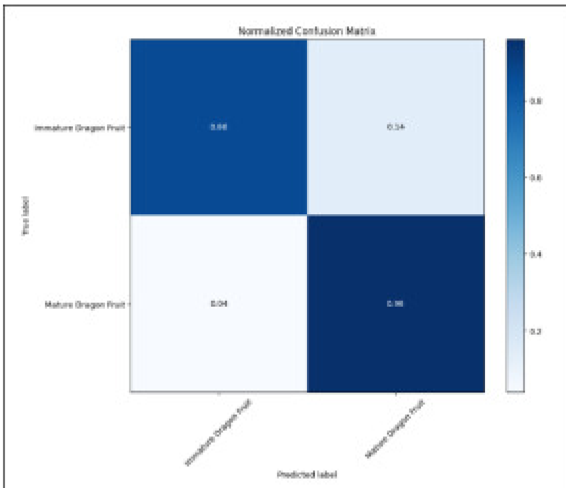
$$F1 - Score = 2 \times \frac{Recall\ \times Precision}{Recall\ + Precision} \quad (4)$$

Conclusion:

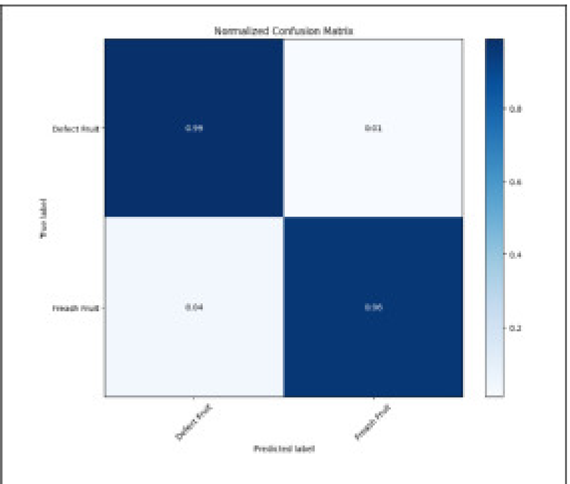
This underlines the ResNet50 architecture's potency in precisely determining the maturity and quality grading of dragons. The model has demonstrated strong performance, achieving a 90% accuracy rate in distinguishing between immature and mature dragon fruit and a 98% accuracy rate in identifying fresh or damaged dragon fruit

Table 4. Classification report for maturity detection and quality grading.

Topic	Class Name	Precision	Recall	F1-Score
Dragon Fruit Maturity Detection Dataset	Immature Dragon fruit	0.97	0.86	0.91
	Mature Dragon fruit	0.82	0.96	0.89
	Accuracy			0.90
Dragon Fruit Quality Grading Dataset	Defect Fruit	0.97	0.99	0.98
	Fresh Fruit	0.98	0.96	0.97
	Accuracy			0.98



(a) Confusion Matrix of ResNet50 model for Dragon Fruit Maturity Detection Dataset



(b) Confusion Matrix of ResNet50 model for Dragon Fruit Quality Grading Dataset

PAPER 9:

APPROACH:
Maturity stage detection based on spectral-spatial detection of hyperspectral image using selected bands.

ML MODEL:

This spectral-spatial processing based on partitional clustering techniques is adopted from Tarabalka et al. (2009) with several changes. There are two major steps in the clustering technique: similarity measure and grouping. In the first step, distance measures include Euclidean distance, Mahalanobis distance, cityblock and cosine. Euclidean distance was shown to be the most suitable for the blueberry detection task after trial and error. The second stage is to group the pixels with the greatest spectral similarity into the same clusters.

- The main procedure includes fruit detection in the spectral domain, morphological operation in the spatial domain, and post-processing. First, SAM was applied to the image. Spectral angles between the pixels and the library were calculated. Since SAM was calculated in the pixel scale with only the spectral information, there were many incorrectly detected pixels that scattered all over the image.
- The result of SAM is an image with each pixel labelled to its best matching class.
- Nested clustering technique and morphological operations are used as two approaches to spatially process the hyperspectral images. Spatial analysis is integrated with spectral analysis using spectral angle mapper. The detection schemes achieve high accuracies, and provide blueberry detection maps with more homogeneous regions and less noise compared to the results using only spectral detection.

Table 1. True positive and false positive rates after each step the spectral-spatial detection based on nested clustering technique.

Class	SAM		SAM+Segmentation	
	TP(%)	FP(%)	TP(%)	FP(%)
Mature fruit	68.7	30.9	65.3	11.1
Intermediate fruit	52.4	43.8	70.5	11.4
Young fruit	75.0	175.0	75.0	25.0

CONCLUSION:

- Two spectral-spatial detection schemes were carried out, and they both improved the detection of blueberry maturity stages using only spectral information. The first method was to combine

segmentation of nested clustering results with spectral detection results, and the second method was to combine the spectral detection results with morphological operations
- Spectral-spatial detection using morphological operations outperformed the detection based on nested clustering by achieving more than 75% true positive rates for all three fruit classes.

PAPER 10:

APPROACH:

A methodology for fresh tomato maturity detection using computer vision.

MODEL:

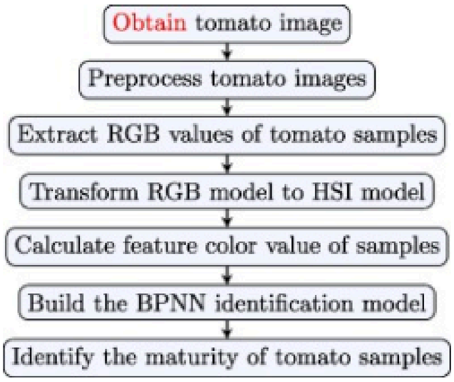
Backpropagation neural network (BPNN) classification technique.

Computer vision system:

The experiments for detecting the required features were conducted in the lab under the normal fluorescent light sources. The camera (SONY NEX-5N, Tokyo, Japan) with lens zoom of between 18 and 55 mm (Optical Steady Shot, Sony, Tokyo, Japan) was installed in the top of the tripod, and the distance between the lens of the camera and the objects was adjusted to 40 cm. The desk was covered with a black cloth as a dark background onto which the tomato samples were placed. All fluorescent lights were turned on while capturing the images of the tomato samples for more uniform light and sufficient illumination. Finally, based on the colour feature values, a BPNN model was built to identify the maturity of the tomato samples using classification.

2.3. Image processing

In this study, a machine vision algorithm was developed to capture the images of the tomato samples, and then it extracted the feature color value to classify the maturity level of the tomato samples. The detection flow chart of the tomato maturity level, based on the machine vision, is shown in Fig. 3.



Technology Used:

An image-processing program was developed using Visual C++ 6.0 on a Microsoft Windows XP operating system (CPU: core i3, 2.8 MHz, memory: Kingston, 2 GB) and the Matrox Imaging Library 9.0 (Matrox, Inc., Dorval, Canada).

RESULT:

Overall, when the input data of the H values are used as the feature color values, the accuracy rate of tomato samples maturity level identification is higher than when the input data of the B values are used as the feature color values. When the neuron number of hidden layers (NHL) is 10, the overall accuracy rate of tomato samples maturity level identification occurs at its highest amount. The accuracy rate of both red and mature tomato samples is 100.00%. The accuracy rate for the green tomato samples is 97.92%, and the average accuracy rate of tomato samples is at 99.31%.

Table 3. The color detection results of the tomato samples using a back propagation neural network.

	Tomato samples	The neuron number of hidden layers and the detection results					
		Blue(8)	Blue(10)	Blue(12)	Hue(8)	Hue(10)	Hue(12)
Red	Number of tomato samples	42	48	48	48	48	47
	Accuracy rate (%)	87.50%	100.00%	100.00%	100.00%	100.00%	97.92%
Orange	Number of tomato samples	32	20	32	45	48	45
	Accuracy rate (%)	66.67%	41.67%	66.67%	93.75%	100.00%	93.75%
Green	Number of tomato samples	45	45	45	46	47	48
	Accuracy rate (%)	93.75%	93.75%	93.75%	95.83%	97.92%	100.00%

4.METHODOLOGY

4.1 PREPROCESSING:

1.Data Organization:

- The first method of the preprocessing involves the directory structure where the image data resides. The structure systematically compiles the image data for subsequent processing.
- The function `uploadFruitDataset ()′ loaded the dataset from specified directory using filedialog.askdirectory()′.

Data Cleaning

Missing Values Handling
Noisy Data Reduction
Outlier Removal

2. Data Transformation:

- Generalisation: Converting low-level granular data into higher-level information using concept hierarchies, enhancing data abstraction and understanding.
- Normalisation: pre–processes them by
resizing to 128 × 128 pixels, converting the colour from BGR to RGB, and then applies the K–means segmentation.
- Attribute Selection: Creating new data properties from existing attributes to enrich the dataset and facilitate effective data mining.

3. Encoding categorical features:

- Label encoding assigns a unique numerical label to each category in a categorical variable. It is suitable for ordinal categorical variables (categories with a meaningful order), as it preserves the ordinal relationship between categories.

```
from sklearn.preprocessing import LabelEncoder
```

- Another possibility to convert categorical features to features that can be used with scikit-learn estimators is to use a one-of-K, also known as one-hot or dummy encoding. This type of encoding can be obtained with the OneHotEncoder, which transforms each categorical feature with n_categories possible values into n_categories binary features, with one of them 1, and all others 0.

4. Data enrichment.

- In this step, data scientists apply the various feature engineering libraries to the data to effect the desired transformations. The result should be a data set organised to achieve the optimal balance between the training time for a new model and the required computation.
- sklearn.utils.shuffle⇒Shuffle arrays or sparse matrices in a consistent way.

4.2 Splitting the Data

Dataset Splitting:

Dataset splitting involves dividing the available dataset into three subsets:

Training Set: The largest portion of the dataset used for training the model.

Validation Set: Used for hyperparameter tuning and model selection.

Test Set: An independent dataset used to evaluate the final performance of the trained model.

2. Purpose of Each Subset:

Training Set: Used to train the model by adjusting its parameters through optimization algorithms like gradient descent. Enables the model to learn the underlying patterns and relationships present in the data. A larger training set allows the model to generalise well to unseen data.

Validation Set: Used for hyperparameter tuning and model selection. Evaluates the model's performance on data not seen during training. Provides an unbiased assessment of the model's performance. Helps optimise hyperparameters without overfitting to the training data.

Test Set: Used to evaluate the final performance of the trained model. Provides an unbiased estimate of the model's performance on unseen data. Assesses the model's generalisation ability to new, unseen examples. Results in an unbiased estimate of the model's real-world performance.

3. Dataset Splitting Process: The dataset is randomly partitioned into training, validation, and test sets while maintaining the distribution of classes or other relevant characteristics. Common ratios for splitting the dataset include 60-80% for training, 10-20% for validation, and 10-20% for testing, although these ratios may vary based on factors like dataset size and task requirements.

4. Implementation:

- The dataset splitting process can be implemented using tools like the train_test_split() function from scikit-learn.
- This function ensures that each subset contains a representative sample of the overall dataset, facilitating comprehensive model development, tuning, and evaluation.

4.3 Model

Support Vector Machine (SVM):

Input Layer:

- SVM models typically don't have traditional input layers like neural networks.
- The input to an SVM model consists of feature vectors derived from the dataset.

Feature Extraction:

- Feature extraction may involve techniques such as:
- Direct use of raw data features.
- Transformation of data into higher-dimensional space using kernels.

SVM:

- The core of the SVM model is the decision function that separates classes by finding the optimal hyperplane.
- The decision function is determined by a subset of training data called support vectors.
- Each training example is represented as a feature vector in the input space.
- Kernel Function:
- SVM models often use a kernel function to map the input data into a higher-dimensional space where classes are more separable.
- Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid.

Regularisation:

- SVM models incorporate regularisation parameters such as C to control the trade-off between maximising the margin and minimising classification errors.

Model Compilation:

- SVM models don't require compilation in the same sense as neural networks.
- However, during model initialization, parameters such as the kernel type, C value, and other hyperparameters are specified.

Model Training:

- Model training is done by using the .fit() function from sklearn.svm model. “X_train” represents the feature vectors of the training data, and “y_train” represents the corresponding labels.
- The training process involves finding the optimal hyperplane that separates the classes with the maximum margin.
- The goal is to minimise the classification error while maximising the margin between classes.

Model Evaluation:

- After training, the model is evaluated using a separate test dataset.
- Performance metrics such as accuracy, precision & recall are computed to assess the model's effectiveness.The model's ability to generalise to unseen data is evaluated to ensure it can perform well in real-world scenarios.

Summary: Support Vector Machine (SVM) models are powerful algorithms for classification and regression tasks. They operate by finding the optimal hyperplane that separates classes in the feature space, making them particularly effective in high-dimensional spaces. SVMs are robust, efficient, and widely used in various domains, including image classification, text classification, and bioinformatics.

4.4 Comparative study

SVM & k-means Model (Accuracy: 99.70%)

- Feature Extraction:
- k-Means clustering efficiently extracts representative features from images, which are then fed into the SVM model.
- This clustering technique helps in reducing the dimensionality of the feature space while preserving essential information.
- Robustness to Noise:
- SVMs are robust to noisy data and outliers, making them suitable for tasks like maturity detection where the image quality might vary & Linear Separability
- enables SVMs to find an optimal hyperplane to separate the classes with a high margin.

CNN for Image Processing(Accuracy: 99.63%):

CNN models, while powerful for image processing tasks, are complex and require a large amount of data for training.

CNNs have many hyperparameters (e.g., number of layers, kernel sizes, learning rate) that require careful tuning to achieve optimal performance.

Limited computational resources or training time may prevent the CNN model from fully converging to an optimal solution.

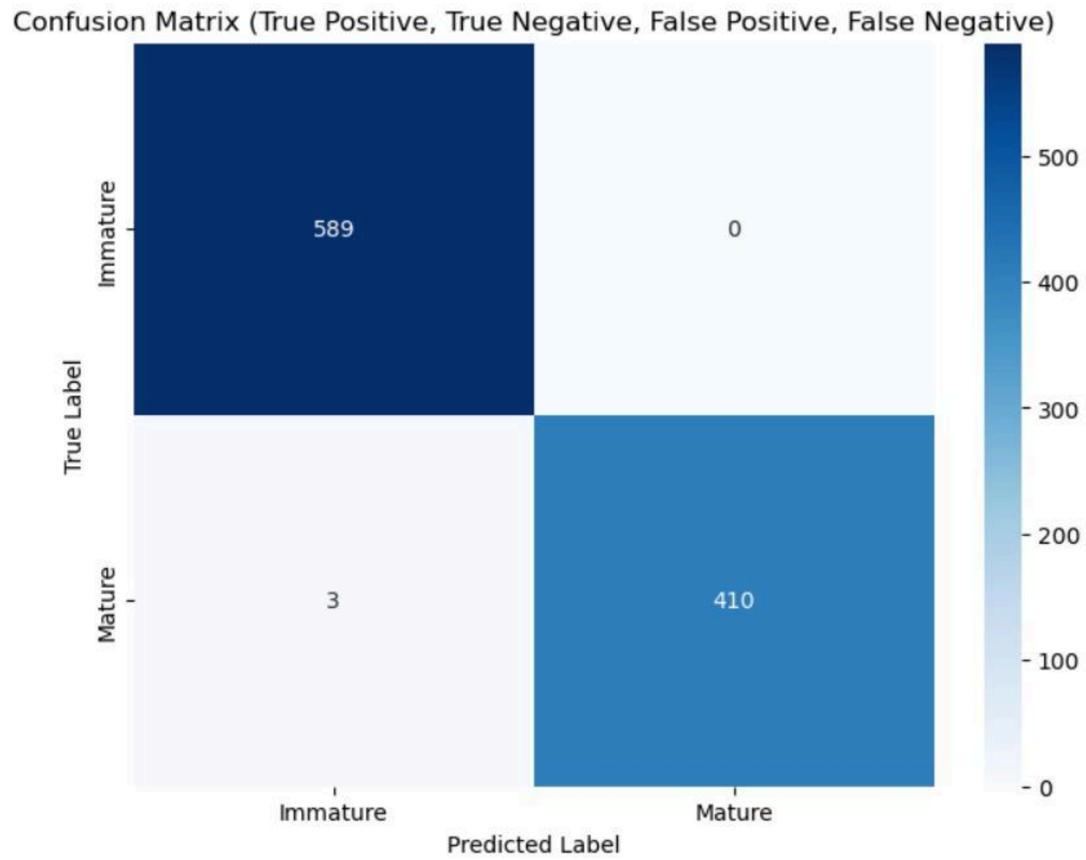
During training vanishing gradients or overfitting may occur, resulting in slightly less accurate data.

Overall, SVM-k means clustering model showed better results among both tested with an impressive 99.70% test accuracy and 99.93% training accuracy with 1.00 precision and 0.992 recall.

5. RESULT

- The SVM model using k-means clustering achieved an impressive 99.27% precision on the training data, indicating its thorough understanding of ripening features. When tested on unseen data, it maintained a high accuracy of 99.7%, demonstrating its ability to generalise effectively.
- These results prove that the model used(SVM) with k-means clustering is a reliable technique for image processing methodologies like maturity detection in fruits.

```
Precision: 1.0000
Recall: 0.9927
True Positive (TP): 410
True Negative (TN): 589
False Positive (FP): 0
False Negative (FN): 3
```



6. REFERENCES:

1) https://www.sciencedirect.com/science/article/pii/S2772375522000958#sec0003

2)https://www.irjmets.com/uploadedfiles/paper/volume2/issue_9_september_2020/3355/1628083132.pdf

3)https://www.mdpi.com/2077-0472/14/3/331#:~:text=The%20study%20explored%20the%20use,of%2072.64%25%20for%20instance%20segmentation

4)https://www.mdpi.com/2077-0472/14/1/36

5)https://iopscience.iop.org/article/10.1088/1757-899X/771/1/012022/pdf

6)https://www.researchgate.net/publication/349286860_Multimodal_Deep_Learning_and_Visible-Light_and_Hyperspectral_Imaging_for_Fruit_Maturity_Estimation

7)https://www.mdpi.com/2073-4395/13/8/2144#fig_body_display_agronomy-13-02144-f003

8)https://www.sciencedirect.com/science/article/pii/S235234092300968X#sec0004

9)https://abe.ufl.edu/precag/pdf/2015-icra%20workshop_CeYang.pdf

10)https://www.sciencedirect.com/science/article/pii/S277227592200017X

11)https://www.sciencedirect.com/science/article/pii/S2772375523002216#:~:text=The%20enlargement%20of%20the%20carpel,when%20to%20harvest%20custard%20apples.