DEVELOPMENT OF A MACHINE VISION SYSTEM TO IDENTIFY MATURED PEPPER SPIKES

by

MEERA T

(2018-18-014)

DEPARTMENT OF FARM MACHINERY AND POWER ENGINEERING

KELAPPAJI COLLEGE OF AGRICULTURAL ENGINEERING AND TECHNOLOGY

TAVANUR - 679 573, MALAPPURAM

KERALA, INDIA

2020

DEVELOPMENT OF A MACHINE VISION SYSTEM TO IDENTIFY MATURED PEPPER SPIKES

by

MEERA T

(2018-18-014)

THESIS

Submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY

IN

AGRICULTURAL ENGINEERING

(Farm Power and Machinery)

Faculty of Agricultural Engineering and Technology

Kerala Agricultural University

Department of Farm Machinery and Power Engineering

Kelappaji College of Agricultural Engineering and Technology

Tavanur - 679 573, Kerala

2020

DECLARATION

I, hereby declare that this thesis entitled "**DEVELOPMENT OF A MACHINE VISION SYSTEM TO IDENTIFY MATURED PEPPER SPIKES**" is a bona-fide record of research work done by me during the course of academic programme in the Kerala Agricultural University and that the thesis has not previously formed the basis for the award of any degree, diploma, associateship, fellowship or other similar title of any other university or society.

Place: Tavanur **Meera T**

Date: - – 2021 2018- 18- 014

CERTIFICATE

Certified that this thesis entitled "**DEVELOPMENT OF A MACHINE VISION SYSTEM TO IDENTIFY MATURED PEPPER SPIKES**" is a bona-fide record of research work done independently by Ms. Meera T (2018- 18- 014) under my guidance and supervision and that it has not previously formed the basis for the award of any degree, diploma, fellowship, associateship to her.

Place : Tavanur

Date : - - 2021

Er. Sindhu Bhaskar

Major Advisor

Assistant Professor

Department of Farm Machinery

And Power Engineering

KCAET, KAU,

Tavanur, Kerala

CERTIFICATE

We, the undersigned members of the Advisory Committee of Ms. Meera T (2018-18-014) a candidate for the degree of Master of Technology in Agricultural Engineering majoring in Farm Power and Machinery agree that the thesis entitled "**DEVELOPMENT OF A MACHINE VISION SYSTEM TO IDENTIFY MATURED PEPPER SPIKES**" may be submitted by Ms. Meera T (2018-18-014) in partial fulfilment of the requirement for the degree.

Er. Sindhu Bhaskar
(Chairman, Advisory Committee)
Asst. Professor
Department of FMPE
KCAET, Tavanur

Dr. Jayan, P. R
(Member, Advisory Committee)
Professor & Head
Department of FMPE
KCAET, Tavanur

Er. Josephina Paul
(Member, Advisory Committee) Asst. Professor
Department of BE & AS
KCAET, Tavanur

Prof. Mohamed Iqbal
(Member, Advisory Committee)
Former Head
Department of Mechanical Engineering
TKM College of Engineering, Kollam

Dr. N. Mini Raj
(Member, Advisory Committee)
Professor
Department of Plantation Crops and Spices
COH, Vellanikkara

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First of all let me thank the Almighty for blessing me and helping me to complete this task as those blessings were invaluable.

My profound gratitude is placed on record to my chairman **Er. Sindhu Bhaskar**, Assistant Professor, Department of Farm Machinery and Power Engineering, KCAET, Tavanur for her deemed support, idea and efficacious advice throughout my project work.

This would be the right moment to thank **Dr. Sathian K. K**., Dean (Ag. Engg), KCAET, Tavanur for his support during the course of the project work.

I am grateful to all the members of advisory committee **Dr. Jayan, P.R**, Professor & Head, FMPE, KCAET, Tavanur, **Er. Josephina Paul**, Asst. Professor, Dept. of BE &AS KCAET, Tavanur, **Prof. Mohamed Iqbal**, Former Head, Dept. of Mechanical Engineering, TKM College of Engineering, Kollam and **Dr. N Mini Raj**, Professor, Dept. of Plantation Crops and spices, COH, Vellanikkara for their help and constructive criticism throughout the work.

It is my pleasure to thank **Dr. Shaji James P**, Former Head, FMPE, KCAET, Tavanur and **Er. Shivaji. K P**, Asst. Professor, College of Agriculture, Ambalavayal and **Dr. Deepak K**, Dept. of FMPE for their advice and guidance rendered during this study.

Let me take this opportunity to thank the selfless efforts of my cousin brother, **Er. Praseed M**, Machine Learning Engineer, SpotDraft Corporate Office, Banglore, all along with me and for his immense help, guidance and support throughout my research work.

Let me also thank all the teachers, technical staffs, office staffs for their support and cooperation and special thanks to computer lab staffs.

I would also like to thank all my friends, juniors and seniors who have helped me in one way or the other for the completion of my research work.

Words are not enough to explain my family's support and prayers throughout my master's period. Thanks to my father, mother, sister and my strong support and reliance, Swathy Chandran, and brothers with substantial helps; Vishnu, Jishnu and Akhil. Special thanks to Vishnupriya and Ms. Sindhu M. for all their love and compassion throughout the work. Finally I acknowledge the financial support of All India Co-ordinated Research Project (AICRP) and Kerala Agricultural University for financially supporting me in completing my research work in the University.

My thanks remain with all those who have helped me in one way or the other, directly or indirectly for the completion of the research work.

Meera T

Dedicated to my Profession

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PLATES

LIST OF APPENDICES

SYMBOLS AND ABBREVIATIONS

%	:	Percentage
+	:	Add
×	:	Multiply
.pbtxt	:	Protobuf Text
2D	:	2 – Dimensional
2K	:	2000
3D	:	3- Dimensional
4500U	:	4500 In U Series
5E	:	5 Enhanced
A	:	Ampere
AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
ANOVA	:	Analysis of Variance
API	:	Application Programming Interface
ARM	:	Advanced Risc Machines
BCM	:	Body Control Module
BPNN	:	Back Propagation Neural Network
C	:	Compiler
CAT	:	Category
CCD	:	Charge Couple Device
CIELAB	:	Commission Internationale De L'éclairage
CMOS	:	Complementary Metal–Oxide–Semiconductor
CMY	:	Cyan-Magenta-Yellow
CMYK	:	Cyan, Magenta, Yellow and Black
CNN	:	Convolutional Neural Network
CPU	:	Central Processing Unit

CUDA	:	Compute Unified Device Architecture
DC	:	Direct Current
DoF	:	Degree of Freedom
DSI	:	Digital Speech Interpolation
ES	:	Expert System
et. al.,	:	And Others
FN	:	False Negative
FP	:	False Positive
FPS	:	Frames Per Second
GA	:	Genetic Algorithm
GB	:	Giga Byte
GHz	:	Giga Hertz
GLCM	:	Gray-Level Co-Occurrence Matrix
GPU	:	Graphics Processing Unit
GUI	:	Graphics User Interface
HD	:	High-Definition
HDMI	:	High-Definition Multimedia Interface
HOG	:	Histogram of Oriented Gradients
HSI	:	Hue Saturation Intensity
HSV	:	Hue Saturation Value
i5	:	Intel Core 5
Int	:	Integer
IoT	:	Internet of Things
IR	:	Infrared
KAU	:	Kerala Agricultural University
KCAET	:	Kelappaji College of Agricultural Engineering and Technology
kg/ha	:	Kilogram per Hectare
KNN	:	K-Nearest Neighbour

LAN	:	Local Area Network
LBP	:	Local Binary Pattern
LDA	:	Linear Discriminant Analysis
LED	:	Light Emitting Diode
MATLAB	:	Matrix Laboratory
MB	:	Mega Byte
MHz	:	Mega Hertz
MIPI	:	Mobile Industry Processor Interface
MLP	:	Multi Layer Perceptron
mm	:	Millimetre
MP	:	Mega Pixel
MTs	:	Metric Tons
NASA	:	National Aeronautics and Space Administration
NCC	:	Normalized Correlation Coefficients
NN	:	Neural Network
Numpy	:	Numerical Python
OpenCV	:	Open Computer Vision
openGL	:	Open Graphics Library
OS	:	Operating System
PC	:	Personal Computer
pH	:	Potential of Hydrogen
pil	:	Python Imaging Library
pip	:	Preferred Installer Program
PLSDA	:	Partial Least Squares-Discriminant Analysis
PNG	:	Portable Network Graphics
QDA	:	Quadratic Discriminant Analysis
R^2	:	Square of Co-Relation Co-Efficient
RAM	:	Random Access Memory
RANSAC	:	Random Sample Consensus

RBF	:	Radial Basis Function
RCNN	:	Region- Based Convolutional Neural Network
ResNet	:	Residual Networks
RF	:	Random Forest
RGB	:	Red Green Blue
RGB-D	:	Red Green Blue – Depth
RoI	:	Region Of Interest
RPN	:	Region Propose Network
RS232	:	Recommended Standard 232
SciPy	:	Scientific Computing Tools for Python
SD	:	Secure Digital
S	:	Second
SK	:	Session Key
SURF	:	Speeded Up Robust Features
SVM	:	Support Vector Machine
Tf - RCNN	:	TensorFlow - Region- Based Convolutional Neural Network
tk	:	Tkinter Module
TN	:	True Negative
TP	:	True Positive
UHD	:	Ultra High Definition
USB	:	Universal Serial Bus
V	:	Volt
VGGNet	:	Visual Geometry Group Network
VOC	:	Visual Object Classes
XML	:	Extensible Mark-up Language

Introduction

Chapter I

INTRODUCTION

Black pepper (*Pipper nigrum*) is the most precious and valuable spices in the world. It is the 3rd most added ingredient in food among the wide range of spices. India is the second largest producer with a production of 61,000 Tonnes, following Vietnam of production, 263,000 Tonnes in 2019. Kerala produced 20,000 Tonnes of black pepper in 2019 (Anon, 2019).

Black pepper is a perennial vine which grows to a height of about 10 m. The fruit also called as peppercorns or pepper spikes are drupes for a diameter of 5 mm. It has berries, which are spherical in shape. Pepper spikes become matured when one of its berries starts to turn yellow. It is normally harvested using hand picking using a single pole bamboo ladder as support for standing. Another method for harvesting is using poles attached with knife for plucking. This method cannot offer identification of black pepper spike at correct maturity stage. The quality of black pepper is assessed on density and size of berries, size of spike and ripeness, at the time of harvest. Black pepper is rich in anti-oxidants, anti-inflammatory compounds and helps to balance blood sugar and cholesterol level. Also it has cancer fighting properties and improves degenerative and damaged brain cells (Meixner, 2019). To ensure the quality of pepper, it should be harvested at its proper maturity stage.

The traditional methods are highly labour intensive and time consuming and it does not have any standardised procedure. The method of harvesting using single pole bamboo ladder is risky and do not have a balanced posture. Only skilled labours can perform such type of task, which is very less in number. In the present scenario, youngsters are more reluctant to extensive and straining works agricultural sector. Farmers harvest fruits based on their visual perspective of maturity, like colour change, size enlargement or shape deformation. This type of maturity detection criteria differs from one farmer to the other, and eventually results in variety of grades in harvested products. There arises a need for standardizing the maturity detection method. This non-uniform distribution affects complex grading processes, unpleasant value addition products, varied preservation processes, difficult and confusing storage and packaging treatments. These factors make harvesting of the fruits a more significant and influencing operation in agriculture. Hence there is an urgent need of suitable mechanisation to harvest the black pepper spikes. The harvester should be able to reach the height of 10 m, identify the matured ones and deposit the harvested produce safely and correctly in the gunny bags.

To achieve correct matured pepper harvesting, latest technologies have been evolved. Artificial intelligence (AI), Internet of Things (IoT), machine learning, image processing etc. are the leading technologies governing the world. Latest technologies employed in robotics ensure less human interference resulting less labour requirement and time saving. Robotic harvesting employing the above technologies is the best option to harvest black pepper spikes at correct maturity level. The main component of a robotic pepper harvesting system is a machine vision system, which can identify black pepper and reduce the human effort to climb up the ladder. The four functions, robotic harvesting system performs are i) identifying the matured fruit, ii) plucking iii) depositing it to a specified location and iv) controlling all the functions. A machine vision system captures and analyses the image features and then utilized for identifying the matured product. It mainly consists of a sensor, processor and a display unit. This system will be an effective tool for identifying the correct maturity stage of pepper spike. The processes involved are non- destructive, accurate and reliable to achieve target of harvesting (Vibhute and Bodhe, 2012). With the development in machine learning, Neural Networks (NN) and GPUs (Graphic Processing Units), the capability of machine vision systems have been improved. So there is a need to develop a machine vision system is to identify the matured black pepper spike to assist the robotic black pepper harvesting system.

Considering the above, a project is carried out, on the development of a machine vision system for identifying matured pepper spike with following objectives:

1. To develop a suitable computer assisted programme for a machine vision system for identifying matured pepper spikes
2. To develop a machine vision system to identify matured pepper spikes
3. To evaluate the developed machine vision system to identify the matured pepper spikes

Review of Literature

Chapter II

REVIEW OF LITERATURE

A brief review of the works done relevant to different aspects of this research is reported here.

2.1 Study of physical properties of matured black pepper spike

Kondo *et al*. (2009) developed a machine vision system for identifying tomato for a robot harvesting. They considered six physical parameters for detection of cluster of tomato. The physical properties of tomatoes included fruit diameter, main stem diameter, main stem angle, peduncle diameter and peduncle length for designing the robotic harvester. The maximum, minimum and average values of the properties were measured in the study. The developed system had an accuracy of 65%.

Ohali (2011) developed a computer vision system for grading date fruits. They considered three grades of dates and studied five external features or five physical properties for detection. They were; flabbiness, size, shape, intensity and defects. They used colour intensity distribution at grey level image for flabbiness measurement. Size was measured using area covered by fruit, shape was measured using outer profile, intensity estimated using number of wrinkles and defects were determined from colour intensity. The system had a maximum accuracy of 80%.

Jun *et al*. (2012) developed a machine vision system for robotic grading by extracting external features of sweet pepper. One variety of sweet pepper was selected for the study and three categories were considered for grading. Mainly four physical parameters were studied; mass, colour, shape and defect. Based on each properties, several grades of sweet pepper were formed and were considered for robotic grading. The overall accuracy was 95%.

Mohammadi *et al*. (2015) developed a system for identifying persimmon fruit using image processing technique. Physical, mechanical and nutritional properties were studied for the detection. The physical properties studied include colour, diameter, equivalent diameter and arithmetic diameter of the fruit, sphericity, surface area and aspect ratio. The system on evaluation had an overall accuracy of 90.24%.

Fashi *et al*. (2019) developed a pomegranate grading system based on image processing. Fifteen parameters of 31 monochrome channels were extracted and were given as input to the Artificial Neural Network (ANN) model. Physical properties studied include length, width, circumference, centroid, texture, area, aspect ratio, diameter of the persistent calyx of the fruit, length of the persistent calyx, ratio of the diameter to length of calyx etc. The system had an accuracy of 98%.

Based on these reviews, a preliminary study on physical properties of matured black pepper spikes were done for this research. The physical properties include, colour, sphericity, length of spike, width of spike and diameter of spike.

2.2 Computer Assisted Programme for Identification

Gao and Lu (2006) developed an algorithm based on image processing for finding pruning positions on a grapevine. The programme was coded in MATLAB 7.0. The images were pre-processed by converting the colour space from RGB into grey scale and thresholding was also done. The canes were having a grey level of 0, that was given a pixel of black colour and background was given white. Then the shape of the canes was extracted using their edges. In this image, the programme searches from the starting pixel and when finds a pixel of 0, which is the cane, the middle point of the cane was identified and eventually its shape. Then the coordinates of this cane was calculated. This algorithm had 85% accuracy.

Kane and Lee (2007) developed a multispectral imaging based yield mapping for citrus fruits. The image acquisition was done at the fall of citrus growing season. They acquired about 500 images using three different filters. They used two feature wavelengths for distinguishing fruit and leaves. The programme was written in MATLAB 7.0. Then the images were pre-processed using histogram stretching and enhancement was done using smoothing functions. Thresholding was done using Otsu's method. One-third of images was used for the testing and rest for training. Each pixel was validated and classified, and this classification percent error was calculated for evaluation. And the average pixel identification was 84.5%.

Kurtulmus *et al*. (2011) developed a machine vision system for detecting citrus fruit. They adopted an Eigen fruit approach for detection. 96 images were captured in daylight. The scanning of citrus tree was done using sub windows. The images were pre-processed and three classifiers were used in this model; an Eigen fruit of intensity, saturation component and a Gabor texture filter. Blob analysis was used for merging detected parts. The model had an accuracy of 75.3%.

Ohali (2011) developed a computer vision system for grading date fruits. They considered three grades of dates. The acquired image is subjected to pre-processing like binary thresholding, and edges were extracted using Sobel edge operator. The features and physical properties like flabbiness, size, shape, intensity and defects were studied. These feature values then fed to BPNN. This was the classifier used. They created two BPNN models of detection for a comparison study. The difference in the two models was in the number of neurons in the input layer. In the results it was found that second model with colour and diameter as features had higher accuracy.

Patel *et al*. (2011) conducted a study for efficient detection of fruits using improved multiple features based algorithm. They used a combination of colour and texture for the fruit detection. The acquired image was filtered, then segmented using Bayesian discriminate analysis, and then region labelling was done. Various features were extracted from the images. The intensity and colour features were extracted at first; the orientation features were extracted using Gabor filters and then edge features were also extracted. Then a feature map was computed. Laplacian Pyramid Transform and fuzzy logic were used for classification. This model was then evaluated and had a detection efficiency of 90%.

Wang *et al*. (2011) conducted a study for judging based on a vision system for tomato maturity. As the maturity detection was crucial for a tomato robotic harvester, the vision system was needed to be more efficient. The programme was written in MATLAB. Images were taken in normal lighting of the greenhouse. Images were pre-processed using Otsu thresholding, converting to RGB colour space and histogram analysis. Then the feature values like area and colour were extracted. The colour values like Hue mean and red-green difference mean were exploited for maturity degree identification. The model had 96% accuracy.

Jun *et al*. (2012) developed a machine vision system for a mobile fruit grading robot. This system was to extract the external features of sweet peppers. They used four groups of sweet pepper to sort. The programme was written in MATLAB. The images were captured with lighting devices. The images were converted to HIS model, and then binarised and noise was removed. Then dilation and erosion filters were used for fill the space in RoI. Then the physical parameters like area, perimeter, maximum and equivalent diameter, roundness etc. were studied and extracted. Defects were identified using HIS distribution. These feature values were given as input to the two-layer perceptron neural network. The model resulted in detection of more than 90% in all groups.

Barbedo (2014) conducted a study to count the number of whiteflies on soybean leaves. Dataset was comprised of 748 images of leaves having various stages of whitefly; nymphs, adult whiteflies, empty exoskeletons and lesions. The programme was coded in MATLAB. The image colour space was transformed to CMYK as pre-processing, different stages of whiteflies had different channels of colour space. Then it was threshold, thus the RoI were obtained. The classifiers used for the programming includes ANN, SVM and deep learning. The developed system was fast and counted the flies efficiently.

Li *et al*. (2014) conducted a study for identifying blueberry maturity stages. They considered four stages of maturity. A set of 46 images were acquired. And from that 23 images were used for training and rest for testing. They adopted the 'Colour component analysis based detection (CCAD)' algorithm for detection. To explore the possibility of utilizing smaller number of colour components, the forward feature selection algorithm (FFSA) proposed by Whitney (1971), and Kumar *et al*. (2001), was used. Different classifiers were used to distinguish between the blueberries, it include KNN, Naïve Bayesian Classifier (NBC) and Supervised K-means clustering classifier using weighted Euclidian distance (SK-means) and a cross validation of these were also done. It resulted that the KNN classifier had highest accuracy of 85-98%.

Thendral *et al*. (2014) conducted a comparative study between two segmentation methods; edge based and colour based. Twenty images of orange were acquired from the internet. For the edge detection based algorithm Canny edge detector was used. In this, firstly the gradient magnitude of the image was computed, using this maximum suppression process is done, and finally the hysteresis process. A binary image is the final product. In case of colour based algorithm, image was pre-processed, converted colour space, converted to binary image and extracted the fruit region. When evaluated, it was found that the colour based algorithm had better results.

George (2015) developed a machine vision system for sorting fruits and vegetables. The programme was coded in MATLAB. Images were acquired using a mobile camera. The images were pre-processed by transforming it to L*a*b* colour model, and then segmented using k-means clustering. Then two features, colour and shape were used for detection. Fuzzy logic was used as the classifier. The model was able to successfully distinguish fruits and vegetables.

Yongsheng *et al*. (2015) conducted a study to locate apples using stereoscopic vision. The programme was written in VC++ 6.0 environment. The pre-processing steps like noise removal, and thresholding were done. Then the features were extracted from the contour image of apple. The circle detection was done using random ring method (RRM). And stereo matching was done both in feature based and area based. This model resulted 89.5% of accuracy.

Barnea *et al*. (2016) conducted a study for the detection of fruit based on colour-agnostic shape. In this study, they had used the multispectral signals for detection. The colour features of the image were used to identify the shape of the fruit and the 3D space of this is analysed. 3D surface features, 3D plane reflectance symmetry features, image plane highlights were used to detect the fruit. Also SVM classifier was used for the classification.

Blok *et al*. (2016) did a research to develop a machine vision system to identify broccoli heads, which usually is a labour-intensive process. Total 7008 images were acquired by this system. For ground truth data, 200 images were selected and labelled. They used the MVTec Halcon (vl2) machine vision software. The images were segmented using colour and textural features. Filters were then applied to the images. Overall accuracy was found to be 92.4%.

Chung *et al*. (2016) developed a system based on a computer assisted programme in MATLAB for identifying a disease called Bakanae disease in rice varieties. They took two varieties of rice; Tainan 11 and Toyonishik. The disease occurred due to contaminated seeds. So the system was developed to identify infected seedlings in early stages of growth. Images of inoculated seedlings of infected and non-infected seedlings were acquired using two flatbed colour scanners. Using image processing algorithms, anatomical points in the scanned images were found. Pre-processing like thresholding and noise removal were done to images. The morphological and colour features were used for distinguishing the seedlings. SVM classifier was used for classification. The developed system had an accuracy of 87.9%.

Dutta *et al*. (2016) conducted a study for image processing based classification of pesticide applied grapes. The images were segmented, converted to grey scale and then, statistical and textural features were extracted from the RoI. The wavelet domain features were more discriminatory, so those features were extracted using Haar filters. Then these were fed to the SVM classifier for distinguishing pesticide treated and non-treated grapes. The evaluated accuracy of this model was found to be 100%.

Guanjun *et al*. (2016) developed a multi-template matching algorithm for recognising cucumber. It was difficult to recognise cucumber in the complex similar background of green leaves. So they incorporated spectroscopy to the system. A series of cucumber images were prepared from the captured images by scale and angle transformation. For identification the template library were used to calculate the matrix of normalized correlation coefficients (NCC). This NCC was used to form a threshold for identification of the cucumber. If NCC was above a threshold value, then there is a cucumber in the image frame. This algorithm was tested, and it resulted in 98% accuracy.

Maldonado and Barbosa (2016) did a study to count the number of green oranges on the tree. They used 1328 images captured in different time and in different lighting conditions. The programme was coded in C++ using OpenCV library. Images were first converted to HSV then the H channel was threshold and V channel was undergone histogram equalisation. The V channel were filtered using Gaussian filter, within that Sobel operators were used for smoothing and Laplacian operator was used edge filtering. Then in feature extraction, the bas-relief representation (which is the Gaussian blur), and Laplacian and sobel masks helped in fruit recognition. SVM was used as a classifier, and counting of fruits were also done. The model resulted in an accuracy error of only 5%.

Nguyen *et al*. (2016) conducted a study for the detection of red and bicoloured apples on tree using a RGB-D camera. The detection algorithm was written in C++ and using Point Cloud Library (PCL). In the first phase, the acquired image was pre-processed, and filtered. In filtering, distance and colour filters were used. In second phase, clustering segmentation was done using Euclidean clustering algorithm. In the last phase, the clusters were split using a Circular Hough Transformation (CHT) algorithm and the location and diameter of the apples was estimated using RANSAC algorithm. When evaluated, the algorithm detected 100% visible apples and 82% partially occluded apples.

Puttemans *et al*. (2016) developed an automated system for identifying fruit for harvesting. They used boosted cascades of weak classifiers for fruit detection. They worked it on strawberry and apple. Image acquisition was done using different cameras for strawberry and apple. Firstly, they made a cascade of weak classifiers. It was built in grey scale. It could detect fruit but not ripe fruit. So they incorporated colour feature and detected ripe fruits. Then they detected the fruit from the clusters. So they used two methods; watershed based segmentation and Trinocular stereo triangulation based segmentation. In the first method, in the cluster region detected, the centres of this were identified. The watershed based segmentation split large blobs into separate fruits. In the second one, Difference of Gaussians (DoG) filter was used in the detected RoI which had found the seeds, and on this seeds 3D triangulation was done and hence separated. It was found that, in the experiments both in strawberry and apple, when the scene specific colour information was used, the detection also improved.

Dorj *et al*. (2017) studied an orchard yield estimation using image processing approach. The dataset included 84 images from 21 citrus trees. Images were pre-processed; colour space was converted to HSV, noise removed, and then threshold using histogram. Two watershed segmentation methods were adopted for obtaining RoI; Distance Transform Watershed Segmentation and Marker-Controlled Watershed

Segmentation Algorithm. Then the citrus were counted by considering them as blobs. In evaluation between human PNG counting and the developed model, a correlation R^2 of 0.93 was obtained.

Khazaei and Maleki (2017) developed an algorithm for grape cluster segmentation using colour features. The images were acquired and a dataset of grape clusters, leaves and other backgrounds were obtained. Images were pre-processed and converted to matrix in the first phase using a MATLAB programme. In the second phase, the Genetic Algorithm (GA) was used to determine the colour features, which defines the layers in ANN, classifying algorithm. In the last phase the model was evaluated. And the overall accuracy was found to be 99.40%.

Malekabadi *et al*. (2017) developed a machine vision system to measure the mechanical properties of onion. Two varieties of onion were selected. The programme was coded in MATLAB (version 2015a). The images acquired were pre-processed. Then useful information like stress, strain, Poisson's ratio and modulus of elasticity were measured from the onions. These values were measured and recorded using conventional methods in order to do the performance evaluation of the machine vision system. The results of the evaluation showed that there was not much significant difference in the conventional method and machine vision system.

Moallem *et al*. (2017) developed a computer vision algorithm for apple grading. In this, the acquired image which was captured from controlled environment was segmented for background removal; stem end, defect and calyx detection and refining the defect. Then from these the features like statistical, textural and geometrical were extracted. These feature vectors were then fed to the classifiers; SVM, MLP and KNN in two manners. In the first one, they considered two categories of apple; healthy and defected, whereas in the second, they considered three grades. On evaluation the SVM classifier was found to be efficient with an accuracy of 92.5% and 89.2% in both the manners respectively.

Momin *et al*. (2017a) developed a machine vision system for soybean grading. They considered five categories for detection including normal, split, contaminated, detect and stem/pods. They used a compact, rugged, low cost camera for acquisition.

They provided both backlighting and front lighting. Morphological feature could distinguish stem/pods and beans, surface features distinguished contaminated ones. The algorithm was coded in Visual C++ and using OpenCV library. The images were converted to HSI model and thresholded. Watershed transform was used for de-bridging images. On evaluation the model had better accuracies in all categories.

Momin *et al*. (2017b) did a study on grading mangoes based on mass using image processing. The images were acquired using XGA format colour camera. Three varieties of mangoes were taken. The programme was written in Visual C++ using OpenCV library. The images were converted to HIS model and segmented. After this, features like area, perimeter, Feret diameter and roundness of the mango were extracted. The area of mango was estimated using an OpenCV library function called cvFloodFill. The perimeter was also measured using the boundary pixels; from these values other features were calculated. Using these features, the mangoes were classified into grades using filters. In evaluation, the model had an accuracy of 97% in projected area.

Wang *et al*. (2017) developed an algorithm for fruit segmentation in varied illuminations. Images of grapes, dates and litchi were captured from three different orchards. The programme was coded in MATLAB (version 8.3). The images were first transformed to RGB, and using the wavelet transform (two-dimensional Mallat algorithm) the colour channels were converted to low and high frequency components. Then these were equalised using histogram. The retinal-based image enhancement was then used as the next step for enhancing. Then the k-means clustering was applied for segmentation. The results proved that, this algorithm was beneficial for efficient segmentation in different illuminations.

Ambika and Supriya (2018) studied detection of vanilla species by employing image processing. They took two vanilla species; *Vanilla planifolia* (cultivated species) and *Vanilla andamanica* (wild species). They used Java language and OpenCV library for algorithm. They employed four geometric features for distinguishing, namely aspect ratio, form factor, perimeter ratio of length and width, and width-perimeter ratio. Image acquired was converted to HSV model, and then threshold. Then the geometric parameters like width, length and area were measured and from these the feature indices were calculated. These feature values were then fed to the classifier (SVM). From the results, it was clear that when the number of features considered is more, accuracy is also high.

Gan *et al*. (2018) conducted a study for citrus fruit detection using colour and thermal images. The programme was coded in Python language. A process called image registration was done to fuse correspondence of pixels between the colour image and thermal image. In this, SURF was used to obtain feature points then a 3D world coordinates were estimated and these were back projected to thermal image. Faster RCNN was used for the fruit detection, aided by the thermal images information like size, pixel intensity and mean image intensity. Hough circle detection was used in thermal images for detection. The bounding boxes of the images were then transformed to thermal image. On evaluation it was found that combining thermal images improved the detection to an accuracy of 95.5%.

Gongal *et al*. (2018) developed a machine vision system for identifying and estimating the apple size in an orchard. Identification was done using image processing through MATLAB software, then histogram equalisation, Wiener filtering, Otsu's thresholding and Circular Hough Transform (CHT). They measured the size of identified apple as the major axis length, for this they used 3D coordinates of pixels and 2D size of pixels. In 3D coordinates method, maximum distance between pairs of pixels in the apple region was calculated in terms of their 3D coordinates. In the 2D method calibration was done. For this a number of checkerboards were taken, and distance to the centroid of a square was measured using a sensor and then also actual pixel size within the square was measured. Using these values a regression model was developed which predicted pixel size based on pixel coordinates. The accuracy was found higher in the 2D estimation method.

Ke-ling *et al*. (2018) conducted a study to select good quality pepper berries using a machine vision system. The machine vision system included a scanner which captured PNG images, then pre-processed, and the features like length, width, projected area, weight and density of the seeds were extracted. 400 kernels of pepper seeds were taken for this. The selected features were statistically analysed. The classification was done using multilayer perceptron (MLP). They took a MLP of single hidden layer. The MLP with 15 features were found to be more efficient with an accuracy of 99.4%.

Pereira *et al*. (2018) conducted a study to predict the ripening stage of papaya using image processing. The papaya samples were collected from market and two images of each papaya were taken. Image acquisition was done using a digital camera. The images acquired were then used for physical and chemical analysis for obtaining information like pH, firmness, soluble solids, total carotenoids etc. They considered three maturity stages. The programme was coded in MATLAB (version R2015a). Images were pre-processed by normalisation, then colour space transformation, and then thresholding. The feature values were then obtained from these images. They were then fed to random forest (RF) algorithm and classification was done. And the model achieved about 94% accuracy in two different dataset evaluations.

Singh, V. (2018) developed a custom object detection classifier. He employed TensorFlow as main library and Faster-RCNN as classifier. On evaluation he obtained 9% accuracy in detection.

Yun *et al*. (2018) conducted a study to detect and locate the wolfberry branches. They acquired 20 images using a CCD camera under black backlight. Gauss smooth filtering was used for normalising the images as pre-processing and then segmentation was done using mathematical morphological method and rest of the noise were removed using minimum area method. Segments were discontinuous due to occlusion of leaves, so it was connected by convex hull theory. The detection rate was found to be more than 60%.

Beyaz *et al*. (2019) developed a methodology to detect fly sting in olive fruit. The image acquisition was done using a CCD camera. The coding was done in MATLAB. The acquired images were pre-processed and threshold. The features like morphological and colour values were extracted and then fed to classifiers like QDA, LDA and SVM. They studied the performances of different classifiers. The PLSDA model had best results with accuracy of 80%.

Cavallo *et al*. (2019) proposed an algorithm for a computer vision system for quality evaluation of grapes. They took two varieties of grapes and three different classifications. Dataset included 400 images of each cultivar of grapes acquired by a CCD camera. These images were pre-processed and segmented using thresholding. Then two sets of features were extracted; first was statistical features in the CIELAB colour space and second was centroid based colour segmentation algorithm. These extracted features were then fed to the Random Forest classifier for training and classification. The model achieved an accuracy of 90% in cultivar Victoria and 100% in cultivar Italia.

Chen *et al*. (2019) developed a machine vision system for the inspection of coloured rice quality system. The images of *Red Indica* rice were captured. These images and near-infrared images were then pre-processed using median filter, and Otsu thresholding was also done. Then by fitting invariant moment ellipses of the rice kernels, each of its major axis was determined, which was one of the physical feature. Some geometric features like area, perimeter, aspect ratio and grain size were also extracted. Chalkiness from the near-infrared images was also extracted for detecting head rice. These feature values were taken to SVM classifier. Two models of SVM were used; one for broken kernel and other for head rice. The proposed system were able to detect very fast; i.e.; in 0.15 seconds.

Fashi *et al*. (2019) conducted a study to develop a system for grading pomegranate fruit, based on their colour and size. They measured and studied physical properties of 200 fruits for this study. They considered three grades of pomegranate. Photographs of both pomegranate and its arils were taken. Three arils of a pomegranate with different colours were taken, and their average had given the mean colour of the parent pomegranate. The images were then pre-processed, segmented and edge detected. Modelling was also done using Adaptive Neuro Fuzzy Inference System (ANFIS) and Response Surface Methodology (RSM). The maximum classifying accuracy of 98% was obtained in ANN.

Fu *et al*. (2019) conducted a study on banana detection using colour and textural properties. Image acquisition was done using a digital camera and 700 images were taken. The algorithm consisted of a coarse algorithm and a fine algorithm. The coarse algorithm included colour space analysis, which is converting to HSV space and then thresholding. The coarse algorithm was to save the detection time. The fine detection algorithm included HOG feature extraction and LBP feature extraction. This model is then classified by SVM classifier and AdaBoost classifier. They did experiments in classification with these classifiers and with different texture features. It was found that HOG+LBP and SVM combination had the best results.

Ireri *et al*. (2019) developed a machine vision system for tomato grading based on RGB feature. The acquired images were subjected to pre-processing and the background removal was done using histogram thresholding technique. Then for the calyx and stalk scar detection 50 images of stalk scar was taken and an algorithm was created. 500 images of defects were taken; their features were extracted and fed to RBF-SVM classifier for defect detection. From the segmented images of tomatoes features like colour, shape and textural were extracted. They operated it with two models SVM model and ANN model. And it was found that the RBF-SVM model had better accuracy.

Lin *et al*. (2019) developed a programme to detect and locate citrus using RGB-D images. Total 506 RGB-D images were captured. The images were then segmented using depth filter and a Bayes classifier. This gave large number of significant points and excluded insignificant points. In the next step they clustered these points using a density clustering process. This density clustering algorithm can reveal the centres of the clusters based on the large density and distance, where these are the two characteristics of each point. They employed the SVM classifier, and it was applied to the feature vectors of the points. So, after detection, the location was determined using 3D coordinates of the cluster. The developed algorithm had a good performance on evaluation.

Nasiri *et al*. (2019) developed an automated sorting for date fruit using image based deep neural network. They considered four grades of dates. Image acquisition was done using a Smartphone. Images were then pre-processed augmented and features extracted. Visual Geometry Group network (VGGNet) was used for the CNN construction. It had subsections like Max-Pooling, Dropout, Flatten, Batch Normalisation and a fully connected layer or dense layer. Then fine tuning of the work was done. Then this model was evaluated and achieved an accuracy of 96.98%.

Rehman *et al*. (2019) developed a system for detecting weed in wild blueberry. The programme was done in C in visual studio 2010. The colour space was converted to HIS and segmented. For image extraction a set of 13 textural features were extracted from these images. 70% of data was taken for training and rest for testing the model. The quadratic classifier was used for classification. The image segmentation and classification is shown in Figure 2.1. And the accuracy resulted to be more than 90%.

Yu *et al*. (2019) conducted a study to improvise the fruit detection in a strawberry picking robot using Mask-RCNN. Image acquisition was done using a hand-held digital camera. About 2000 images were acquired in different light intensities and periods. Then the images were given annotations to distinguish ripe and unripe using Labelme. This will generate mask images. These mask images were then exploited for detection and classification.

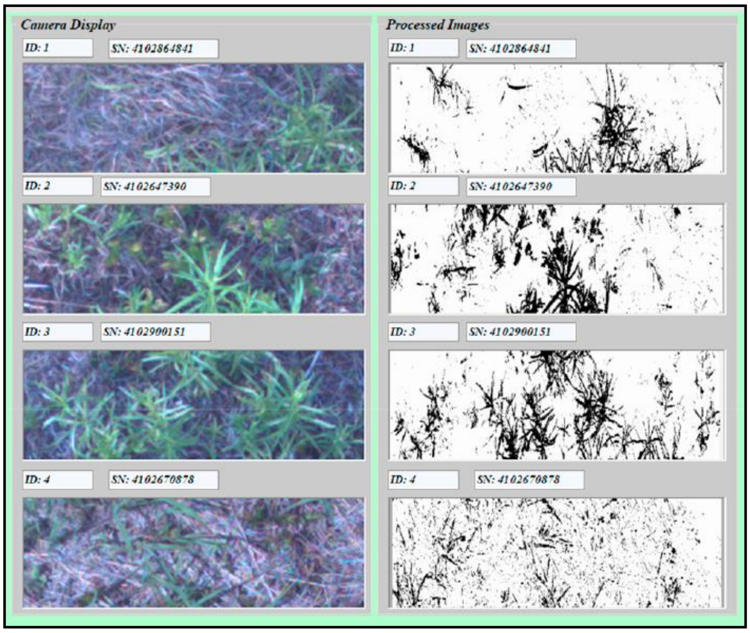

Figure 2.1 Colour co-occurrence matrix of the acquired image and quadratic classifier-based algorithm (Rehman *et al.,* 2019)

Azarmdel *et al*. (2020) conducted a study to grade mulberries based on their ripeness by image processing. Three categories of ripeness were considered. The acquired images were pre-processed and threshold using Otsu thresholding. Geometrical, colour and textural features were extracted from the images. Then the selected features were fed to classifiers, ANN and SVM. The ANN model had higher accuracy when evaluated.

Chaaro and Anton (2020) developed a system for identifying different crops and weeds. MATLAB was used as man software and python and FarmBot were used for image acquisition process. For classification purpose they used RCNN. Background removal and histogram equalization were done as pre-processing. On evaluation the accuracy for classification was found to be 78.10%.

Habib *et al*. (2020) developed a machine vision system to detect disease in papaya. The images were acquired using a mobile camera. The two third of the dataset was used for training and the rest for testing. The images were pre-processed by histogram equalising, converting to L*a*b* colour space and then segmented using k-means clustering. Then the features like statistical features and gray-level co-occurrence matrix (GLCM) features were extracted. The statistical features helped with the defect detection and the GLCM features with textural characteristics. The SVM classifier was used for classification. And the model achieved more than 90% accuracy in classification.

Monhollen *et al*. (2020) developed a corn kernel loss assessment system, which quantify the loss using a machine vision system. The image acquisition was prompted in response to a signal produced by camera cart wheel at definite intervals. Then the detection was done using Faster RCNN. They retrained Resnet-50 CNN architecture for corn kernel detection using ImageNet database. The programme was coded in MATLAB and using Deep Learning Toolbos Model.

Zhang *et al.* (2020) conducted a study to estimate the shaking locations for a robotic apple harvester. A multi-class object recognition algorithm was developed for this. They used faster-RCNN for the detection and three different pre-trained deep learning networks; AlexNet, VGG16 and VGG19. Images were obtained from ImageNet and COCO and were processed in RGB and CIELAB colour space. On evaluation they obtained an accuracy of 72.7% for shaking locations with an average time of detection of 0.45 seconds.

2.2.1 Computer Assisted Programme for Detection of Different Maturity Stages

Clement *et al*. (2013) developed a computer algorithm for classifying cucumber based on their degree of curvature relative to the length. They used C++

language and OpenCV 2.3.1 library. They used two methods for determining the length and curvature. In the first method; ellipsoid approximation, the cucumber was considered as an ellipsoid, the long axis and short axis of the cucumber will give an idea about the curvature and thus they identifies the edge of the cucumber. The other method; active transformation, they identify an axis which is equidistant from both the sides. This was done by calculating distance transform of the image. In the experimental results the active transformation method had 15% less error than ellipsoid approximation method.

Mohammadi *et al*. (2015) conducted a study using image processing technique for detecting maturity of persimmon fruit. The physical, mechanical and nutritional properties were studied and used for classification. Mechanical properties were firmness and elasticity and nutritional properties were total soluble solids (TSS) and titratable acidity (TA). Image processing code was written in MATLAB (version 2008a). The acquired image was filtered, segmented and features were extracted. Using intensity of colour feature ripeness was determined, whereas other features were used into the linear (LDA) and quadratic Discriminant analysis (QDA) based classifiers to identify the maturity stage. The QDA classifier resulted better accuracy of 90.24% in detection.

Zhao *et al*. (2016) conducted a study to detect ripe tomatoes in a greenhouse using AdaBoost classifier. Images were acquired in natural light. . The programme was written in MATLAB (version R2013a). Eighty images were randomly selected for training. The images were classified to sub-windows. The haar-like features of each sub-window were extracted. AdaBoost algorithm was used for training process. They also used the average pixel value (APV) based colour analysis for tomato detection. The detection accuracy was about 96%.

Mim *et al*. (2018) did a study for automatic detection of mango ripening stage. They considered six maturity stages as per United States department of agriculture (USDA) standard classification. Images of 100 mangoes of different maturity stages were acquired. Then images were pre-processed, segmented and twenty four features were extracted. From these, significant features were selected based on the correlation.

Then the classification was done using decision tree. The algorithm is shown in Figure 2.2. The result had 96% accuracy in maturity stage detection.

Taofik *et al*. (2018) developed a system to detect ripeness of tomato and chilli. They considered four ripeness categories. Images were pre-processed, and then segmented using a K-Means Clustering, to identify nature of the fruit. Then the colour features were extracted. Then these were fed to Fuzzy Logic, which detected the ripeness stage. In the evaluation, the detection of tomato had an accuracy of 80% and chilly of 90%.

Tu *et al*. (2018) developed a machine vision algorithm for detecting passion fruit and its maturity stage. They used natural outdoor RGB-D images for the study. They divided the maturity, stages into five; young, near- young, near mature, mature and after-mature. They did the detection in two stages, firstly, faster region-base Convolutional neural networks (Faster RCNN) integrated with colour and depth images were used for passion fruit detection. Secondly, from the detected, features were extracted using the dense scale invariant features transform (DSIFT) algorithm along with locality-constrained linear coding (LLC). These features were then classified using support vector machine (SVM) classifier. They obtained 92.71% detection accuracy.

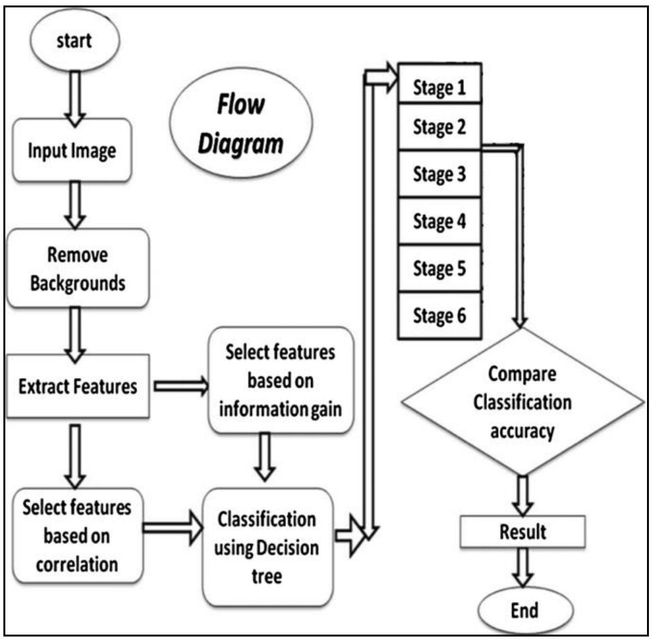

Figure 2.2 Algorithm for grading (Mim *et al*., 2018)

Tan *et al*. (2018) conducted a study to detect blueberry with maturity stage. Here, they had three maturity stages; mature, intermediate and young. The algorithm of their development included three steps; first they created a dataset of 1374 original colour images second, from the dataset, for each images HOG (Histogram Oriented Gradients) feature vectors were calculated and then trained using linear SVM to detect the blueberry. Finally a* and b* features of colour space were used to categorise the maturity stages in the detected space. They also used KNN (K-nearest Neighbour) and newly developed TMWE (Template Matching with Weighted Euclidean Distance) classifiers for the identification. The study found to be efficient in recognition and the KNN classifier yielded the best accuracy.

Wan *et al*. (2018) developed a programme for a computer vision system for identifying maturity stage of tomatoes. The image acquisition was done in the lab under normal fluorescent light. Image processing programme was done in Visual C++ 6.0 and Matrox Imaging Library. Threshold segmentation, noise cancellation, image contour extraction and boundary fill algorithm were used for processing. After identifying the region of interest, an inscribing circle was created and this feature extraction area was then divided into five concentric circles. The colour values were then calculated from these sub domains and then given as input into the back propagation neural network (BPNN). This detects the maturity stage of the tomato and it had an accuracy of 99.31%.

Based on the above reviews, programme to identify matured black pepper spike were developed in two different platforms, one with OpenCV library and Haar cascade classifier and second with TensorFlow library and Faster-RCNN ANN model.

2.3 Machine Vision System for Identification

Kane and Lee (2007) developed a multispectral imaging based yield mapping for citrus fruits. They acquired images using a monochromatic near-infrared camera along with interchangeable optical band pass filters. On evaluation identification accuracy was found to be 84.5%.

Kondo *et al*. (2009) developed a machine vision system for a tomato cluster harvesting robot. The robotic system consisted of a vision system, a manipulator and an end effector. The image acquisition setup included two colour cameras, filters and lighting devices. The image acquisition system was assembled on a mobile platform. Three categories of tomato were considered. More than 70% accuracy was obtained on evaluation.

Rajendra *et al*. (2009) developed a strawberry harvesting robot with a machine vision system algorithm. The system consisted of DoF manipulator, an end effector, vision system and a rail type travelling system. Colour CCD cameras were used for acquisition of images. Five categories of strawberries were considered in this research. A robot with a 4-degree of freedom manipulator was developed in this. Two sensors were fitted in a suction pad for detecting strawberries. The cluster of strawberries, their position and the peduncles were also detected. From the results, it was found that 75% of the strawberries were detected in the real situation.

Ohali (2011) developed a computer vision system for grading date fruit. The image acquisition system consisted of two cameras and lamps. There was an image capturing chamber, consisted of two Logitech cameras and had an illumination system also. A PC was connected as display unit. After capturing the image they were sent to a processor, for processing. The results showed 80% of accuracy in detection.

Ji *et al*. (2012) developed an automatic recognition vision system for an apple harvesting robot. Images were acquired using CCD camera in day light. Other components include industrial computer with Intel Pentium4 1.7 GHz processor and 512 M memory was associated with it. The recognition success rate was 89%.

George (2015) developed a system for sorting fruits and vegetables. An Arduino microcontroller processor was used as part of the hardware in the system. A sensor was used for the detection of fruits. A motor was connected for transporting the fruit. On evaluation, fruits and vegetables were successfully identified.

Arakeri and Lakshmana (2016) developed a computer vision based fruit grading system for tomato. The system had hardware and a software part. The hardware was composed of a conveyor belt, camera, computer system and bins for collection. The software part or image processing part consists of image processing libraries or modules, filters, and artificial neural network. The tomatoes were laid and moved on the conveyor belt and it stops in front of the camera for image acquisition. The sensor here used was digital camera, the image is sent to the processor through USB port. After classification of image the signal was send through RS232 serial transmission. As evaluation result, an accuracy of 100% and 96.5% were obtained for defect and maturity detection respectively.

Blok *et al*. (2016) did a research to develop a machine vision system to identify broccoli heads and evaluated them based on two metrics. The model was evaluated based on Dice similarity coefficient (DSC) and individual broccoli head detection. DSC was used for evaluating pixel segmentation. And the performance evaluation of the system for detection was done using confusion matrix. This matrix represented the TP, TN, FP and FN values. From these values parameters like, sensitivity, specificity, precision and accuracy were calculated. On evaluation almost all parameters had above 90%.

Vithu and Moses (2016) stated that the use of computer vision, near-infrared spectroscopy, magnetic resonance spectroscopy, electronic nose, spectroscopy using the Fourier transform in infrared light, X-ray and hyperspectral images are some of the techniques that can be used to overcome limitations like poor detection and error in recognition.

Ahmadabadi *et al*. (2017) developed an online grading machine vision system for peeled pistachio kernels and peels. The system was composed of conveyor belt, lighting unit, camera, processor and sorting unit. A CCD colour video camera was installed in the acquisition chamber. The conveyor belt powered using a motor. The display unit used was a computer. ATMEGA16 microcontroller was as the controlling unit. The overall accuracy was found as 94.33%.

Malekabadi *et al*. (2017) developed a machine vision system for measurement of mechanical properties of onion. Two cylindrical probes were attached to the testing machine along with two cameras in two different planes, one inside the probe and other one outside. Another sensor was employed for measuring the load on objects. The hardware setup is shown in Figure 2.3. The acquired images from the camera were transferred to computer which was used as a display and image analysis unit. On evaluation, it was revealed that statistically there was no significant difference between human inspection and developed machine vision system.

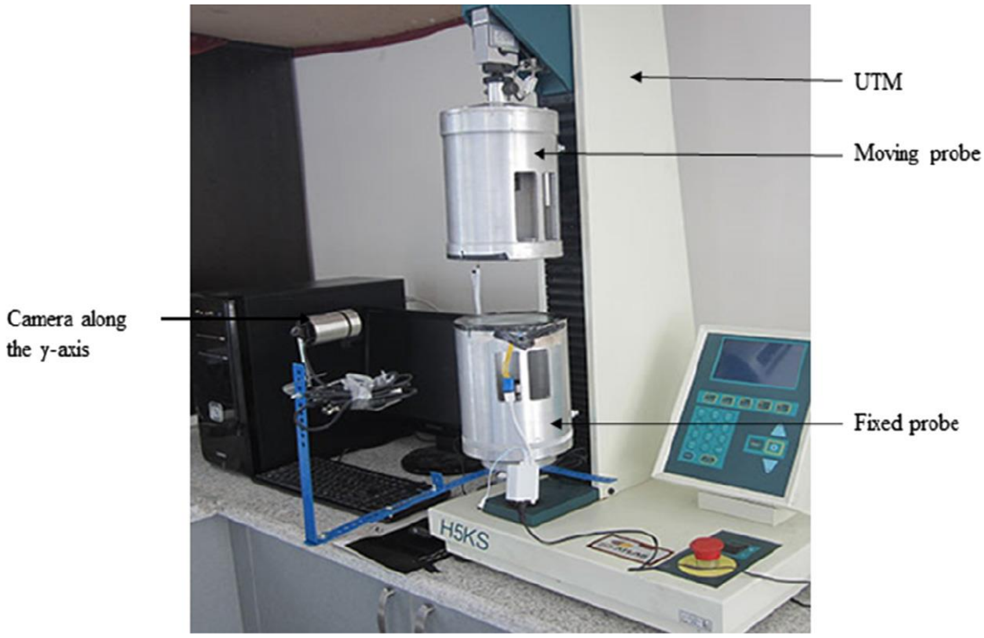

Figure 2.3 Probe and camera setup (Malekabadi *et al*., 2017)

Momin *et al*. (2017a) developed a machine vision system for grading soybean. They used web camera for acquisition with lighting systems. Logicool Webcam Software V-1.1 (Logitech. Com., Romanel-surMorges, Switzerland) was employed for setting up of the camera's operating system. A PC was used as a display unit and for image processing. The developed system had an accuracy of 96%.

Momin *et al*. (2017b) conducted a study on grading mangoes based on mass using image processing. For image acquisition two webcams for top view and side view were used. The top view camera was held at 440 mm above horizontal background and side view camera at a distance of 285 mm from vertical background. Computer was used for display and as an image processing platform. The system had 97% of accuracy.

Gongal *et al*. (2018) developed a machine vision system for identifying and estimating the size of apple. The image acquisition unit had CCD cameras and time-of-flight light based camera. The sensors included a laser sensor also for measuring distance to centroid of apples. The developed system had an accuracy of 84.8%.

Pereira *et al*. (2018) did a study to predict the ripening stage of papaya using image processing. The image acquisition unit had a lighting unit, a digital camera vertically held over the background at a distance of 17.5 cm from the object. It was connected to the processor via a USB port. The system had 94.3% of classification accuracy.

Cavallo *et al*. (2019) developed a computer vision system for quality grading and evaluation of grapes. As sensor they used 3 CCD digital cameras with dedicated CCD for colour channels. A Linos MeVis 12 mm lens system also was used and kept perpendicular to background. The developed system had an accuracy above 90%.

Chen *et al*. (2019) developed a machine vision system for a coloured rice quality system. The image acquisition system consisted of a CCD camera (Microvision CO., Ltd., China), a near infrared backlight and a power unit. The display unit was a computer which processed the acquired images. The overall accuracy was obtained as 96.4%

Ireri *et al*. (2019) developed a machine vision system based on RGB feature for tomato grading. The acquisition was done using a Hikvision Mini Camera. The camera was connected through an Ethernet to the processor. The processor was an Intel core i5-4500U CPU, 4 GHz and 16 GB physical memory. And the display unit used was a Microsoft Windows 10 PC. The developed system achieved overall accuracy of 98.9%.

Rehman *et al*. (2019) developed a system for detecting weed in wild blueberry. The system consisted of VR sprayer consisting of four colour cameras, desktop computer and a shuttle computer. The sensor were of four µEye colour cameras, fixed 0.18 m ahead of the sprayer nozzles to provide distance and time for image analysis. The display was provided by fan-less desktop computer and an eight-channel computerized variable rate controller was used as controller. The computer and camera was connected using 12.2 m USB cables. Also a flow rate control mechanism consisting of Dickey John Land Manager-II controller (LMC) module was attached to trigger the spraying. The accuracy obtained for the quadratic classifiers was 94%.

Williams *et al*. (2019) developed a robotic kiwi fruit harvester with four arms to operate in orchards. The machine vision system part of the harvester included a pair of cameras (Baslar ac1920-40uc USB 3.0). An autonomous multi-purpose modular platform was created first. The systems are all controlled using servo controllers. Each axes were actuated using ROS nodes on commodity computing hardware. For the machine vision system the sensors were colour cameras, which were held at the centre of picking part of arms. The detected images were then passed through blob detector to find the centres. The positioning was done using stereo point matching method. The whole harvester was undergone performance evaluation and it was found that vision system had an accuracy of 89.6%.

Kumar *et al*. (2020) developed a pepper harvester based on image processing. They used the Haar Cascade algorithm for the detection. They created a XML file using acquired images. A cutting mechanism was there which included a scissor actuated by a DC motor, a rack and pinion and trail mechanism for sliding and a funnel to collect the cut pepper, the vision system had a camera as sensor and processor was Raspberry Pi 3. When operated, it was found that, the detection time was much lesser than that of ordinary inspection. In 0.3 seconds pepper was detected.

Figure 2.4 Pepper harvester (Kumar *et al*., 2020)

Monhollen *et al*. (2020) developed a system for identifying corn kernels. The system comprised of a trailer, image acquisition part, processor, display unit, a rotary encoder, microcontroller and a supportive frame. Camera height and trailer pitch were adjustable according to imaging plane. Rotary encoder was to measure distance between images. The sensor for machine vision system was a 12MP RGB camera. CMOS photodetector and rolling shutter were capable of 2.5 fps with maximum resolution. A host computer was used for processing and display. Image acquisition was triggered based on the movement and signal provided by camera cart wheel, where the wheel rotations were measured by encoder. The processing unit was Arduino and a Nano microcontroller. Connection was taken through USB serial communication. This system had an overall accuracy of 91%.

From the above reviews of previous works, the sensor of the system is selected as a webcam, the processor is Raspberry Pi and the display unit selected is Raspberry Pi LCD Display Module.

2.4 Performance Evaluation of Machine Vision System

Dutta *et al*. (2016) did a study for image processing for classifying pesticide applied grapes. The developed model were evaluated and TP, TN, FP, FN values were measured. Based on these values sensitivity, specificity and accuracy were calculated. The accuracy values for different SVM, Random Forest, Naïve Bayes, ANN and LOOCV classifiers were calculated and a comparative analysis were done. And except LOOCV classifiers others had 100% accuracy. But the SVM had high sensitivity and accuracy values.

Zareiforoush *et al*. (2016) developed an Automatic Control System (ACS) to control performance of rice whitening machines. The ACS acquires information about the qualitative indices of rice discharged by sampling. The evaluation was done using two parameters; degree of milling and percentage of broken kernels. The sampled rice kernels were transferred for image acquisition. The images were then processed and information were obtained and fed to the processor, which was then compared with previous information. Fuzzy logic approach was taken for simulating the inference structure of human operator in the machine adjustments. The accuracy of ACS was 89.2%.

Ahmadabadi *et al*. (2017) developed an online grading system for peeled pistachio kernels and peels. A comparative evaluation of SVM classifiers with different kernel functions was done. In the performance evaluation of developed model, when the pistachios were conveyed on the belt, a narrow strip of product was considered which fell from the conveyor. In this way, possibility of missing product was able to reduce. To analyse the performances, two statistical indices were used; correct classification rate (CCR) and accuracy (AC). Accuracy was calculated using true positive, true negative, false positive and false negative values. And CCR calculated using number of samples classified and total number of samples. The overall accuracy of the sorter was 94.33%.

Roshanianfard *et al*. (2018) developed a harvesting robot for pumpkin. It had a robot tractor as a platform, robotic arm, an end-effector and a controlling system. The performance evaluation of the harvester was done on eight parameters, working space, system resolution, harvesting possibility zone, accuracy and repeatability, harvest success rate, cycle time and damage rate. The working space and harvesting possibility zone was measured and compared to required value, system resolution was calculated by moving the control system over 20 squares, then the values of length of square, number of squares and tolerance of the system were used to determine system resolution. Accuracy and repeatability was calculated using positional accuracy and number of repetitions and harvest success rate was calculated by the number of successful harvests. Cycle time included the average time for complete harvesting along with localization, fruit grasp, transport of pumpkin and transport to next fruit. The damage rate was the number of intact harvested pumpkins per total harvested pumpkins. On evaluation of the harvester, had a success rate of 94%.

Chen *et al*. (2019) developed a computer vision system for a coloured rice quality system. The system was evaluated on the basis of three parameters; recall, precision and accuracy. These were calculated from the observation like TP, TN, FP and FN. These were evaluated for all the cases of broken kernel, head rice, damaged rice etc.

Ireri *et al*. (2019) developed a machine vision system based on RGB feature for grading of tomatoes. The evaluation of the system was done using the parameter of accuracy. On evaluation observations like TP, TN, FP and FN were measured and then accuracy was calculated. They evaluated the system for different classifiers like SVM, ANN and RF.

Azarmdel *et al*. (2020) did a study to grade mulberries based on their ripeness using image processing technology. The performance of both the classifiers were evaluated based on parameters called Mean Square Error, calculated using number of data, values observed, and values predicted. The performance of algorithm was evaluated using parameters like, sensitivity, specificity and accuracy; calculated using TP, TN, FP and FN. On evaluation ANN model with four features had high accuracy.

Monhollen *et al*. (2020) developed a corn kernel loss assessment system, which quantify the loss using a machine vision system. The accuracy of the programme was validated by an Average Precision (AP) metric. A comparative evaluation of programme generated and manually drawn boundary boxes was done. Also another assessment of image analysis was done using two subsets of images. The subsets were processed and evaluated for two types of errors; in programme identified bounding boxes and in kernels not identified by the programme. As observations in evaluation TP, FP, TN and FN were measured. From those values parameters like precision, recall, average precision and a metric, Combined Accuracy (CA) were calculated. The system had overall accuracy of 82% in evaluation.

From the reviews above mentioned the performance parameters selected for this research are; sensitivity, specificity, accuracy and time taken for detection.

Materials and Methods

Chapter III

MATERIALS AND METHODS

This chapter explains the processes, materials and methodology used for the development and performance evaluation of machine vision system for identification of matured black pepper. The whole process is explained under four subtitles viz. study of physical properties of black pepper, development of computer assisted programme for identification of matured black pepper spike, development of machine vision system for identification of matured black pepper spikes, and performance evaluation of machine vision system for identification of matured black pepper spikes.

3.1 Study of Physical Properties of Black Pepper Spikes

The study of physical properties of black pepper spike was carried out to measure and analyse properties and their values which can directly and indirectly affect the design and development of the machine vision system. The physical parameters studied include colour, sphericity of berries, length of the spike, average diameter of spike and diameter of berries (Pereira *et al.,* 2018). Two different varieties of black pepper, *Karimunda* and *Panniyur 1* were considered for the study. The samples were collected from KCAET, Tavanur, Randathani, Malappuram and Kattappana, Idukki. The procedure and methods adopted for measuring colour, sphericity, length of spike, diameter of spike and diameter of berries are explained here.

3.1.1 Colour

In image processing, colour is an important attribute for detection and decision making. Colour has different methods of representation in image processing. A machine vision system for horticulture products requires the ability to capture, process and analyse colour images, where algorithms are suitable to detect, extract, and quantify the attribute of colour as much as a customer does (Sandoval *et al*., 2018). Some of the commonly used colour models are RGB, HSV, CMY and CMYK. The colour value was measured from 20 images using the RGB colour model. The RGB value was measured using a programme in python. The colour model is shown in Figure 3.1.

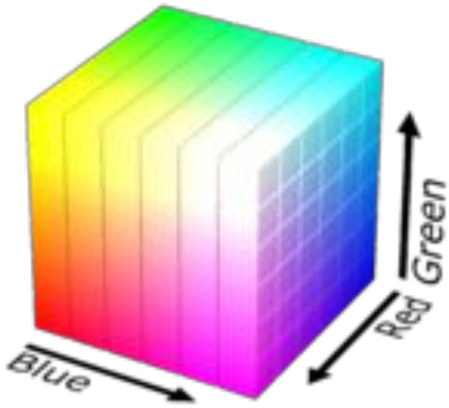

Figure 3.1 RGB colour model

3.1.2 Sphericity

Sphericity is an important engineering property of a biological material which is a result of the shape of the entire commodity. It is the degree of roundness of an object. Sphericity is the ratio of diameter of inscribing circle to diameter of circumscribing circle. Sphericity was measured for 20 pepper berries using the projector microscope. From the observations obtained, sphericity was calculated using the following equation (1).

$$\text{Sphericity} = \frac{\text{Diameter of inscribing circle}}{\text{Diameter of circumscribing circle}} \qquad \text{…. (1)}$$

3.1.3 Length of spikes

Lengths of black pepper spikes were taken for the estimation of the size of a pepper spike. It was measured as the total length from the top end of black pepper to its bottom tip, with and without peduncle as shown in Figure 3.2. It affects the design of cutting tool, conveying unit and storage unit of the harvester. It was measured using a steel rule from 20 different black pepper spikes.

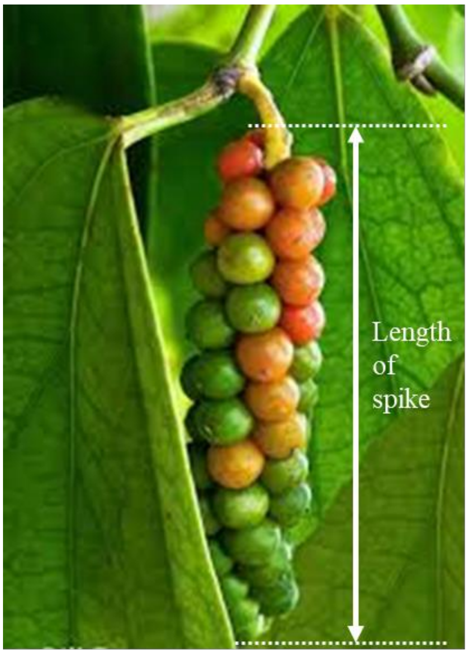

Figure 3.2 Length of spike

3.1.4 Diameter of spikes

Diameter of the pepper spike at three levels viz top end, middle and bottom end was measured for size estimation as shown in Plate 3.1. The average of these three values were recorded as average diameter of spikes. The diameter of spikes was measured using a vernier caliper from 20 black pepper spikes.

Plate 3.1 Diameter of spike

3.1.5 Diameter of berries

Diameter of the berries was measured using vernier caliper. This was measured using 20 replications of pepper berries.

3.2 Development of Computer Assisted Programme for Identification

For the development of programme, image acquisition process, the selection of image acquisition process, selection of platforms, programming language, development of programming codes and selection of classifiers are discussed here.

3.2.1 Image Acquisition

Images of black pepper spikes were collected using a mobile camera of resolution 5 Megapixel at a distance of 30 cm. Mobile camera was selected, as in similar studies carried out by George, (2015) and Habib *et al*., (2020). The images of black pepper, of different maturity stages viz matured and non- matured were collected for different varieties, mainly *Karimunda* and *Panniyur 1*. Images were taken in the natural environment of normal day light intensity (Ji *et al*., 2020). The different maturity stages of the black pepper acquired is shown in Plate 3.2. A black pepper is said to be matured when one or two of the berries start to turn yellow (Thomas, L. and Rajeev, P, 2015). The image number 1 in Plate 3.2 is unmatured and image number 2 to 8 are matured.

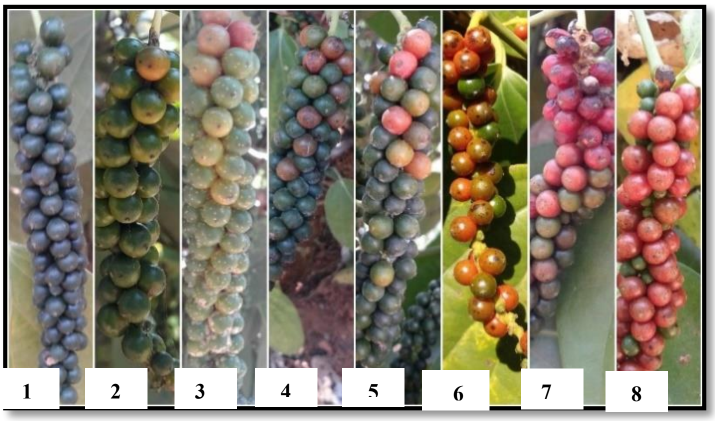

Plate 3.2 Maturity stages of black pepper

3.2.2 Selection of libraries

OpenCV 2.4.13 and TensorFlow 2.0 libraries were selected for developing computer assisted programme for identifying matured black pepper.

3.2.2.1 OpenCV

OpenCV is the library used in OpenCV-Haar cascade platform, following the procedure used for the development of a pepper harvester by Kumar *et al.* (2020) as in article 2.2. This is a huge library supporting lots of functions and algorithms which can perform machine learning applications.

3.2.2.2 TensorFlow

TensorFlow is the library used in Tf-RCNN platform, following the procedure used for the object detection by Singh, V. (2018), as in article 2.2. TensorFlow is a free and open source math library prominent in machine learning created by Google.

3.2.3 Programming Language - Python

Python is the programming language used, following the procedure used for the development of a pepper harvester by Kumar *et al.* (2020) as in article 2.2. It is an object-oriented language and helps hugely to build an object detection programme (Python (programming language), 2020).

3.2.4 Classifier for Detection

The different programming classifiers used for the identification of matured black pepper spike are discussed below.

3.2.4.1 Haar Cascade Classifier

Haar cascade is the classifier used, following the procedure used for the development of a pepper harvester by Kumar *et al.* (2020) as in article 2.2. In this classification approach, the model is trained using a large number of positive and negative images. Xml files of the dataset was created using training and a cascade file was received as an end product, which is then used for the detection. (OpenCV, 2020). OpenCV already contains many pre-trained Haar feature based classifiers for many objects. Those XML files are stored in opencv/data/haarcascades/folder. That is because Haar cascade is best when associated with OpenCV platform (Mordvintsev, A. and Abid, K 2013).

3.2.4.2 Region-Based Convolutional Neural Network (R-CNN)

R-CNN was selected as classifier for the Tf-RCNN platform. Region-based Convolutional neural networks or regions with CNN features (R-CNNs) are a pioneering approach that applies deep models to object detection (Goh, 2017).

3.2.4.2.1 Faster R-CNN

Faster-RCNN is the classifier used in Tf-RCNN platform, following the procedure used for the object detection by Singh, V. (2018), as in article 2.2. Faster RCNN consists of a RPN, a base net and anchors, and the basic network is shown in Figure 3.3. This is usually used for training on smaller datasets. And TensorFlow, is the platform which can train and run deep neural networks for classification, and image recognition. So Faster-RCNN has better performance with TensorFlow platform (Yegulalp, S. 2019).

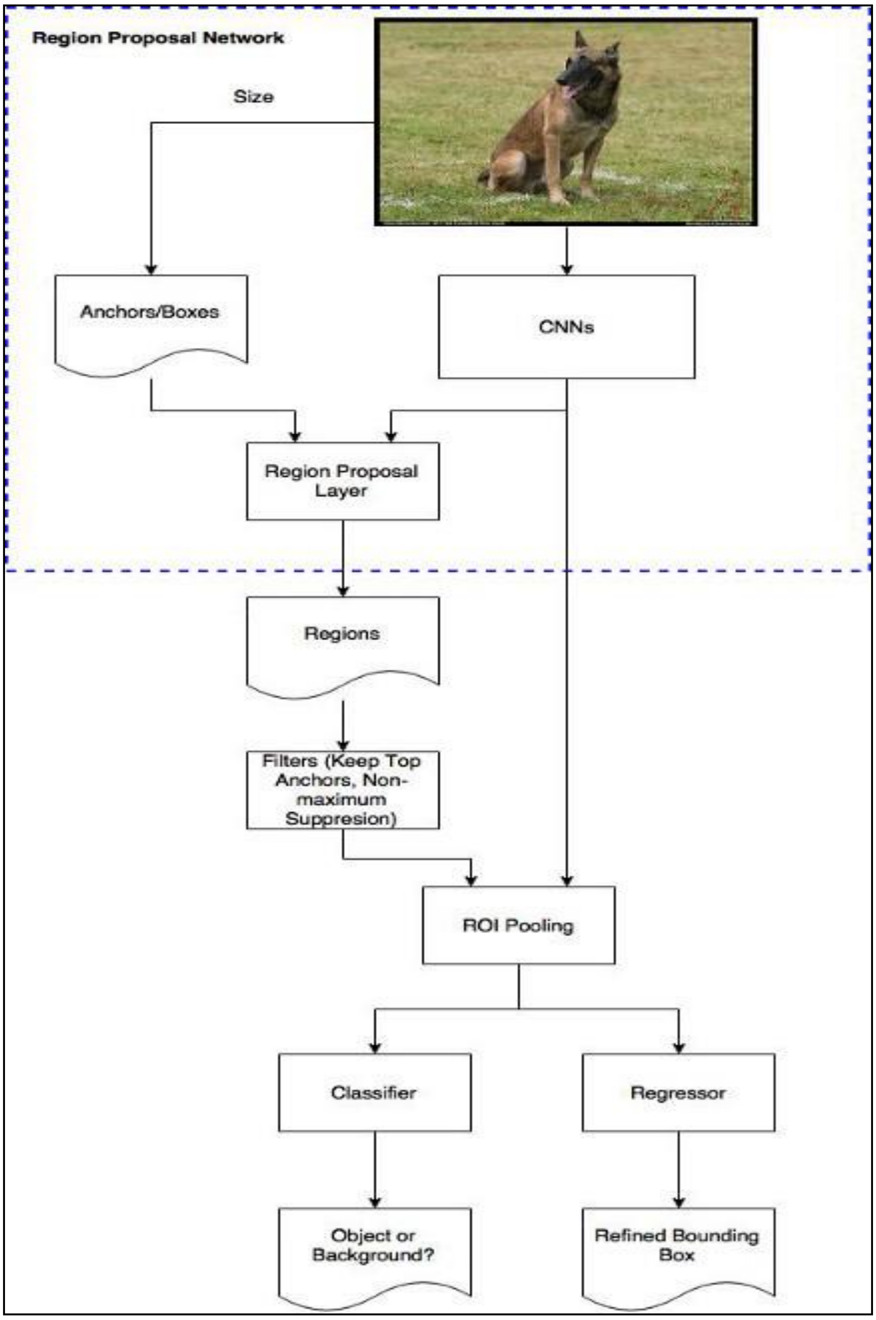

Figure 3.3 Architecture of Faster R-CNN (Gao H, 2017)

3.2.5 Development of Codes

The methods adopted for programming codes are discussed here.

3.2.5.1 OpenCV – Haar Cascade Method

The programme based on OpenCV- Haar Cascade platform is carried out using the below depicted algorithm shown in Figure 3.4.

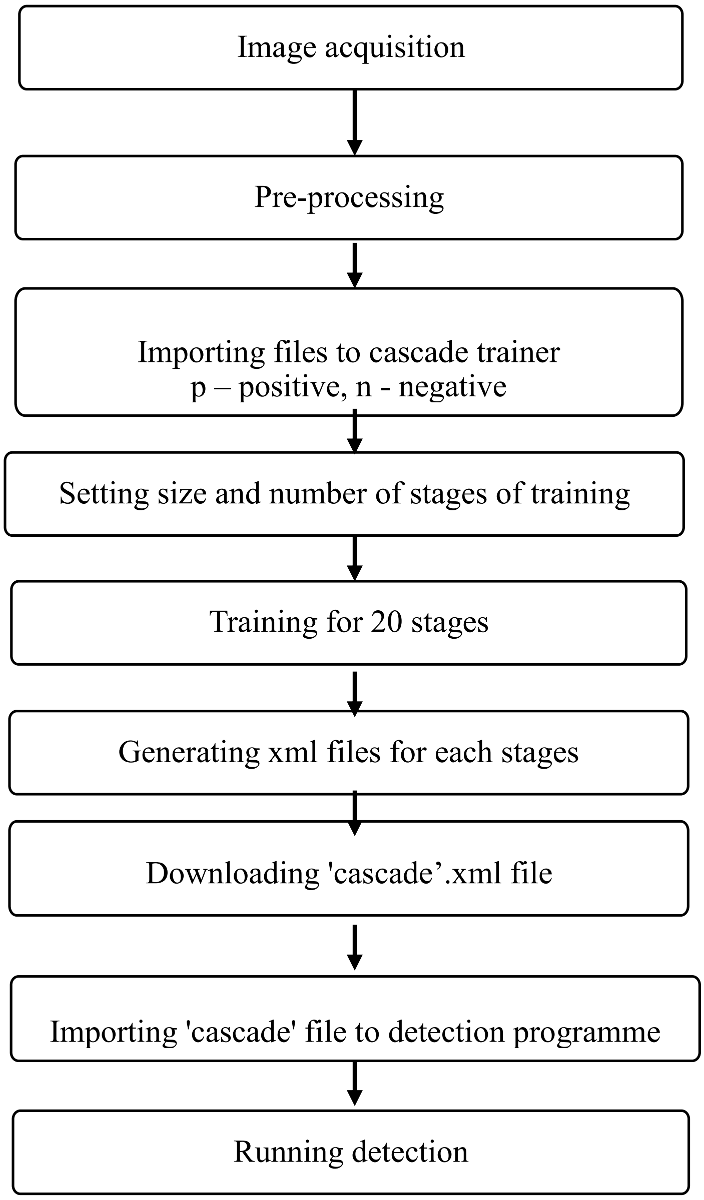

Figure 3.4 Algorithm of OpenCV-Haar cascade platform

In this, the dataset is comprised of positive and negative images. In the dataset images, 400 positive and 400 negative images were used. Combinations of different number of positive and negative images were tested for best object detection. On conducting trials in different combinations, it was found that a dataset of 400 positives and 400 negative images were obtained as best combination. Positive images are the images of matured black pepper and negative are the images without matured black pepper. Negative images included images of leaves, stems, and other plants and no trace of any matured black pepper spike and is shown in the below Plate 3.3 and Plate 3.4.

Plate 3.3 Positive samples

Plate 3.4 Negative samples

The positive images acquired were filtered and noises were removed using MS paint. All the background of the images was removed by editing as shown in Figure 3.5. And all the images were reduced to uniform sizes of 200×200 pixels. The image size was determined based on trial and error method.

Figure 3.5 Pre-processed image

Training of haar cascade classifier was carried out using an application called Cascade Trainer GUI created by Amin Ahmadi, (2016). This application is a GUI which helps to set the training parameters for training. Using this trainer, the cascade classifier can be trained, tested and improved. Before starting the training, a folder was created for the classifier, within it two folders n and p for negative and positive images respectively were created. Then in the trainer, in the Train tab folder paths, number of stages, size of the images and feature type were also inserted. The training was done for 20 stages. Then the training were started and when the training was completed a new folder 'classifier' was created. From the folder the end product of the training, 'cascade' an xml file was downloaded and saved. Using this Cascade file, an object detection programme was created and executed for identifying matured black pepper spikes.

A matured pepper identification programme was developed in python language using OpenCV library functions. The programme created for identifying matured pepper was evaluated by videos. OpenCV functions enabled the video loading, displaying and detecting the matured black pepper spikes in every frame. This programme was used for evaluating the developed platform.

3.2.5.2 TensorFlow - RCNN Method

The training of Tf-RCNN platform was carried out using the following algorithm as shown in the Figure 3.6.

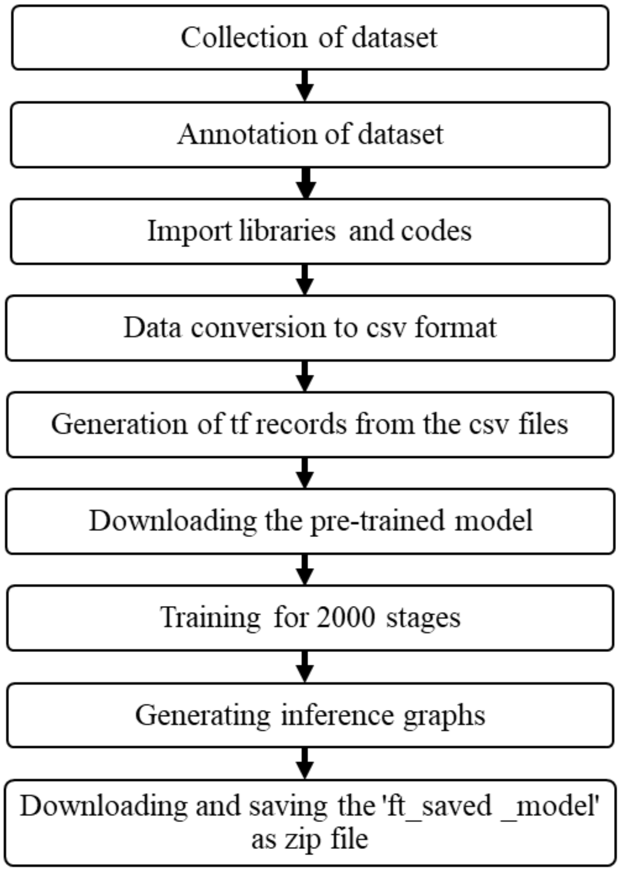

Figure 3.6 Training algorithm of Tf-RCNN platform

A dataset of 150 positive images were collected. The annotation of images was then carried out using LabelImg.exe. This tool uses the python language and it saves the output files as xml files in PASCAL VOC format. Inside this xml files the details of the object in every image will be stored. From the dataset, separate folder for the train images was created. All the training images were resized to 600×800 pixels. The size of dataset and pixel size of the images were decided based on a trial and error method done for determining suitable training size of images. Then they were copied to this folder. Another folder for annotated images was also created. Then LabelImg application was opened and the directory to the train images folder was also opened. Each image was displayed and using the 'Create \nRectBox' option, rectangles were drawn around the objects to be annotated as shown in the Figure 3.7. Once drawn, a pop-up box was appeared to enter the name of the object. Each of the annotations given was saved after this as an xml file in the chosen folder.

Figure 3.7 LabelImg window

A part of pre-processing of the dataset and training and testing were carried out in Google Colab. After labelling of the dataset, the folders of these labelled images were uploaded to the Colab as a zipped file. Then two codes for converting the xml data to csv format and another one for generating TensorFlow record files were uploaded. Before executing these codes, a label map was created in '.pbtxt' format which comprised the label and dictionary of the objects. Then the uploaded xml files in the dataset, was converted first to csv format and then that csv format was used to generate the tf record files for each test and train folder. Some associated helper packages like protobuf- compiler, python- pil, python-xml and python-tk Cython, jupyter and matplotlib were also used.

TensorFlow object detection API was used for detection of matured black pepper spike. The API was cloned into Colab and then repository models, dataset with test images and train images with annotation to be trained was uploaded. An IG folder which saves the inference graph of the trained model at the end of the training, checkpoints (CP) folder in which the checkpoints of the trained model will be saved, two python codes for evaluation and training and a configure file for the chosen model were also added. The training was done for 2000 stages (Singh, V. 2018), till the loss value became zero. The checkpoints were saved in the CP folder and then the inference graphs were generated. The inference graphs and the training results throughout the stages of training were clearly displayed. The end product of the training was then downloaded and saved as a zip file.

3.3 Development of Machine Vision System for Identification

A machine vision system is a complex compilation of lighting, optical and electronic computer equipment. Simply, a machine vision system comprises of a processor, a sensor and an output display unit. Here in this research the processor is Raspberry Pi and sensor is a web cam and display unit is a LED monitor.

3.3.1 Concept of a Machine Vision System

Agriculture is a sector which needs more visual inspection for decision making. Every phase in agriculture is equally influential for the end product. So the quality of the products relays on the processes carried out and their decision making in each phase. A machine vision system is such a decision making system which has become a common method in agriculture due to the automation. In agriculture, it is noteworthy that computer vision applications have grown due to reduced equipment costs, increased computational power, and increasing interest in non-destructive food assessment methods (Mahajan *et al*., 2015).

A machine vision system utilises an image for getting maximum information from it, and conveyed to main system and then to actuators. The extracted information can be used for many actions in the robotic system. The conventional methods of inspection, plant protection and harvesting have less accuracy, time and labour consuming.

3.3.2 Selection of components

For the development of machine vision system for identifying matured black pepper spikes, the components include a sensor, a processor and a display unit.

3.3.2.1 Sensor

Webcam is the image acquisition sensor selected for this machine vision system, following the procedure used for soybean grading by Momin *et al.* (2017a) as in article

2.3. The webcam model used in this study is a USB 2.0 HD PC camera of QUANTUM, with a resolution of 640×480 Pixels and 5 of 0.3 Million Pixels. It had a cable length of 1.4 m copper transmission wire which enabled quick image transmission and is shown in Plate 3.5. It had a focal length of 3 cm to infinity. This camera had higher resolution and highly portable. It could be easily connected to a PC as well as a monitor and when connected to USB hub it is extendable, thus providing a wider view. It also resists overheating.

Plate 3.5 Webcam

3.3.2.2 Processor

Raspberry Pi is the processing unit selected for this machine vision system, following the procedure used for the pepper harvester by Kumar *et al.* (2020) as in article 2.3. It is a small single board computer developed by Raspberry Pi foundation, United Kingdom. This is mainly used for applications like games, word processing, spreadsheets etc. This model of processor was widely accepted and became a common and basic robotics and computer science hardware. There has been a variety of models of Raspberry Pi released so far, which were designed and created for specific and different robotic performances. Every Raspberry Pi model released consists of a Broadcom System on a Chip (SoC) with a CPU and an on-chip graphics processing unit (GPU).

For this system, we used a processor model of Raspberry Pi 4 Model B. This is a version of Raspberry Pi 4 series. This unit have a 64-bit quad-core processor, 4 GB RAM, 2.5 GHz and 5 GHz 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, Gigabit Ethernet port, along with several ports including 2 USB 3.0 ports, two micro HDMI ports enabling 4K UHD video, 2-lane MIPI CSI camera port for connecting a Raspberry Pi camera, 2-lane MIPI DSI display port for connecting a display, 4-pole stereo output and composite video port, micro SD port and a 5V/3A DC power input. The processor speed ranges from 700 MHz to 1.4 GHz.

Raspberry Pi is widely accepted because of its low cost, high processing power, availability of different interfaces including USB, HDMI, Ethernet, Wi-Fi and Bluetooth, and it is easily available, and mainly it have a compact size and shape and also it is USB powered which makes it very suitable for field work and portable. This version of Pi is very energy efficient; fast networking, upgraded USB capacity and have different choices of RAM. But the most advantageous is that it supports different operating systems and computer languages facilitating different and various robotics performances and projects. Because of all these features Raspberry Pi 4 was selected as the processing unit for the research work. The selected Raspberry Pi is shown in Plate 3.6.

Plate 3.6 Raspberry Pi

3.3.2.3 Display Unit

The display unit used was a Raspberry Pi LCD Display Module with a 320x240 resolution display. It was of the model 3.2 LCD-V4 with 7 x 5 x 1.5 cm side lengths. Its signal input included SPI interface. This monitor had brighter and sharper images with a good resolution display. It had a slim design with high resolution and flicker free images. It also had less power requirement. It was suitable to Raspberry Pi versions of B+, 2B, 3B and 3B+. In this research, this display unit was thus selected due to the compact and wireless nature, being suitable for field works.

3.3.3 Installation of Machine Vision System

Raspberry Pi works in an operating system called Raspbian OS. This system should be installed using a graphical SD card writing tool called Raspberry Pi Imager developed by Raspberry Pi itself. A Strontium class 10 SD card of 16 GB storage capacity was used for this purpose. The SD card was inserted to a PC using card reader, then in the Raspberry Pi Imager the appropriate OS for the display unit was selected from the list and SD card was chosen. Then by clicking on the 'WRITE' option the OS was installed into the SD card.

The OS installed SD card was then inserted into the Raspberry Pi. Then the Pi was connected to the display unit using the HDMI to micro HDMI cable. The cable had a 2K resolution and was enabled for 1080 ultra HD. The cable was of length 3 m. The ethernet cable and the webcam were connected to the Pi through the USB ports. Ethernet cable of type CAT 5E of 1 m length was used, so that it could give more feasibility. Then the USB power supply was turned ON and the Pi and monitor were powered up. The Raspberry Pi desktop was displayed in the monitor. Then the Python 3.6.9, OpenCV, TensorFlow, numpy, imutils and accessory libraries were installed into the system through terminal. The programme was then executed through Ubuntu terminal and then Jupyter Notebook. Using the video capture programme, the code was run.

3.3.4 Block Diagram of Machine Vision System

The block diagram of the machine vision system is shown below in Figure 3.8.

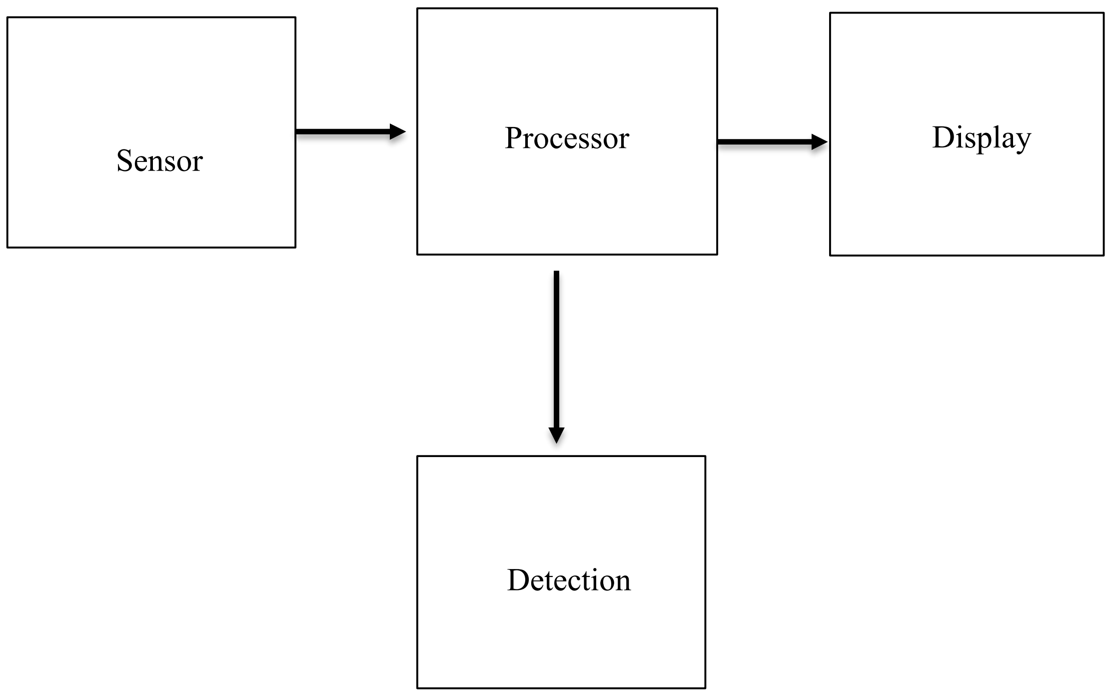

Figure 3.8 Block diagram of machine vision system

3.3.4.1 Algorithm for detection

The detection programme of Tf-RCNN platform is based on the following algorithm as shown in Figure 3.9

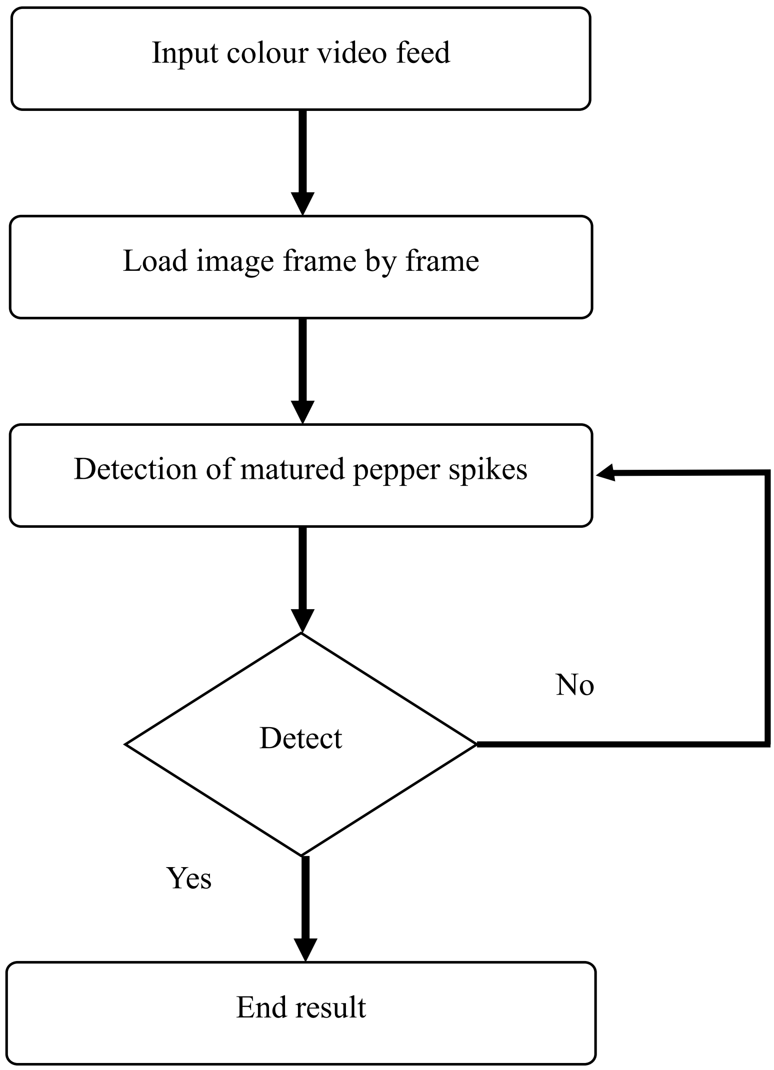

Figure 3.9 Algorithm for detection in Tf-RCNN platform

3.4 Performance Evaluation of the Machine Vision System

The machine vision system was evaluated for programmes in two different platforms viz OpenCV with Haar Cascade and TensorFlow with Faster-RCNN. The performance evaluation of the developed system was carried out using certain performance parameters and analysis. These parameters were needed to quantify the performance. The following parameters were considered for the performance evaluation of the developed machine vision system.

3.4.1 True Positive

True Positive is when the object is present in the frame and the model detects it correctly. In this study, true positive is correctly identifying the matured pepper spike as shown in Plate 3.7. The number of true positives in the platform's performance evaluation was counted and recorded manually.

3.4.2 False Positive

False Positive is when the model incorrectly detects the positive class. In this study, false positive is incorrectly identifying the matured pepper spike in its absence as shown in Plate 3.8. The number of false positives in the platform's performance evaluation was counted and recorded manually.

3.4.3 True Negative

True negative is when the model correctly identifies the negative class. In this study, true negative is identifying the negative objects in the frame as shown in Plate 3.9. The number of true negatives in the platform's performance evaluation was counted and recorded manually.

3.4.4 False Negative

False negative is when the model incorrectly identifies the negative class in the frame. In this study, false negative is showing no detection when, there are positive objects in the frame as shown in Plate 3.10. The number of false negatives in the platform's performance evaluation was counted and recorded manually.

Plate 3.7 True positive

Plate 3.8 False positive

Plate 3.9 True negative

Plate 3.10 False negative

3.4.5 Sensitivity

It is a statistical measure for the performance of a detection model. It measures the proportion of actual positives that are detected as such. It is also the rate of true positive. The maximum value for sensitivity for a good model is unity. It is calculated as following equation (2).

$$\text{Sensitivity} = \frac{\text{Total true positives}}{\text{Total true positives} + \text{Total false negatives}} \qquad \ldots. (2)$$

3.4.6 Specificity

It is another statistical measure for evaluating performance of a detection model. It measures the proportion of actual negatives that are correctly detected as such. It is also the true negative rate. The maximum value for specificity for a good model is unity. It is calculated as following equation (3).

$$\text{Specificity} = \frac{\text{Total true negatives}}{\text{Total true negatives + Total false positives}} \quad \text{.... (3)}$$

3.4.7 Accuracy

It is a statistical measure for consistency of the performance of developed model. It is the ratio of total number of true positives and true negatives to the total number of observations. It is calculated as following equation (4).

$$\text{Accuracy} = \frac{\text{True positives + True negatives}}{\text{True positives +True negatives +False positives +False negatives}} \quad \text{.... (4)}$$

3.4.8 Time taken for a detection

The time taken for detection for both the open source platforms was calculated using 'Time' module available in OpenCV library,. A code was written in the object detection programme using this module and it enabled us to calculate time for each detection.

3.5 Statistical Analysis

Two-way Analysis of Variance (ANOVA) was carried out for testing the effect of independent variable on dependent variables at 5 percent level of significance. Monhollen *et al.,* (2020) in a similar study, carried out their statistical analysis at 5% level of significance.

Treatment	**Parameters**
1. OpenCV-Haar Cascade 2. Tf-RCNN	1. Sensitivity 2. Specificity 3. Accuracy 4. Time

Result and Discussion

Chapter IV
RESULT AND DISCUSSION

The results obtained by conducting this study is explained here under the subtitles; physical properties of the matured black pepper spikes, computer programme for identifying matured black pepper spike, development of machine vision system and performance evaluation of the developed machine vision system.

4.1 Physical Properties of the Matured Black Pepper Spikes

Five different physical properties viz., colour, sphericity of berries, length of spike, diameter of spike, and diameter of berries were studied in two varieties of pepper viz *Karimunda* and *Panniyur 1*. The results obtained in the study is shown in the Table 4.1.

4.1.1 Colour

The colour of the matured black pepper was measured using the RGB value. The RGB value of colour image ranges from [0, 0, 0] to [255, 255, 255]. From the observations, it was found that the RGB values for *Karimunda* variety ranged from (20, 39, 3) and (255, 224, 111) and in *Panniyur 1* variety, value ranged from (35, 54, 10) - (255, 240, 100).

4.1.2 Sphericity of pepper berries

From the Table 4.1, the sphericity for *Karimunda* was obtained as 0.62 – 0.78 and in case of *Panniyur 1*, it was 0.32 – 0.55. The average sphericity of *Karimunda* was 0.58 and in Panniyur 1 it was 0.42. It was found that the sphericity of *Karimunda* variety was higher than *Panniyur 1*.

4.1.3 Length of black pepper spikes

The *Karimunda* variety was having a length about 6 -14.5 cm and *Panniyur 1* was having a length of 9 – 19.5 cm including the peduncle. For *Karimunda* variety, an average length (with peduncle) of 10.83 cm was obtained and in *Panniyur 1* variety, an average length (with peduncle) of 13.64 cm was obtained. The *Panniyur 1* variety was having more length than *Karimunda* variety.

4.1.4 Diameter of pepper spikes

From the Table 4.1, it is observed that, in the *Karimunda* variety, minimum diameter was 0.7 cm and maximum diameter was 1.3 cm. In case of *Panniyur 1* variety the minimum diameter found was 0.87 cm and maximum diameter as 1.7 cm. The average diameter of the *Karimunda* variety was obtained as 1.08 cm and *Panniyur 1* variety had 1.31 cm. From the results, it was found that in case of diameter of spikes *Panniyur 1* variety is larger than *Karimunda*.

4.1.5 Diameter of berries

From the data as shown in Table 4.1, average diameter of *Karimunda* was 0.42 cm and *Panniyur 1* was 0.59 cm. The diameter of berries of *Panniyur 1* variety was larger than *Karimunda*. For *Karimunda* variety, minimum was 0.3 cm and maximum diameter was 0.5 cm. In case of *Panniyur 1* variety, minimum was 0.45 cm and maximum was 0.7 cm.

All the replications of the physical properties and their calculations are depicted in the appendix.

Table 4.1 Physical properties of black pepper spike

<table>
<tr><th colspan="2">Property</th><th>Karimunda</th><th>Panniyur 1</th></tr>
<tr><td colspan="2">Colour</td><td>(20,39,3) - (255,254,111)</td><td>(35,54,10) - (255,240,100)</td></tr>
<tr><td colspan="2">Sphericity of berries</td><td>0.62 – 0.78</td><td>0.32 – 0.55</td></tr>
<tr><td rowspan="2">Length of spikes</td><td>With peduncle</td><td>6 -14.5</td><td>9 – 19.5</td></tr>
<tr><td>Without peduncle</td><td>4.5 - 13</td><td>7 - 18</td></tr>
<tr><td colspan="2">Diameter of spikes</td><td>0.77 – 1.27</td><td>0.87 - 1.7</td></tr>
<tr><td colspan="2">Diameter of berries</td><td>0.3 – 0.5</td><td>0.45 – 0.7</td></tr>
</table>

Plate 4.1 Pepper detected

On evaluating using a python module, it was found that RGB value of matured black pepper spikes is ranging from (20, 39, 3) to (255, 240, 100), berry sphericity from 0.32 to 0.78, berry diameter ranging from 0.3 to 0.7, length of spike ranging from 4.5 to 19.5, and diameter of spike ranging from 0.77 to 1.7 was detected as matured as shown in Plate 4.1.

4.2 Computer Programme for identifying Matured Pepper spike

The computer assisted programmes for identifying matured black pepper spikes in OpenCV- Haar cascade and Tf-RCNN platforms are explained here.

4.2.1 OpenCV – Haar Cascade Platform

The programme developed for OpenCV-Haar cascade u is shown below.

```
import argparse

import cv2
import os
def main(xml_path):
```

```
    pepper_cascade = cv2.CascadeClassifier(xml_path)

    cap = cv2.VideoCapture(0)

    while 1:
        ret, img = cap.read()
        # Resize image to 200 * 200
        img = cv2.resize(img, (200,200))
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        #Detect pepper
        peppers = pepper_cascade.detectMultiScale(gray,
  1.3, 3)

        # add this
        for (x,y,w,h) in peppers:

  cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)

        cv2.imshow('img',img)
        k = cv2.waitKey(30) & 0xff
        if k == 27:
            break

    cap.release()
    cv2.destroyAllWindows()

#parser = argparse.ArgumentParser()
#parser.add_argument('--xml_path',help="Path to
  XML",required=True)
#args = parser.parse_args()
#main(xml_path=args.xml_path)
```

```
main(xml_path='cascade.xml')
```

4.2.2 TensorFlow – RCNN Platform

The programme developed using TensorFlow as main library and faster-RCNN as classifier is shown below. The separate programmes for training and identification developed are narrated below.

4.2.2.1 Training Programme for TensorFlow-RCNN platform

The programme developed for training is shown below.

```
!pip install tensorflow==1.14
!apt-get install protobuf-compiler python-pil python-
  lxml python-tk
!pip install Cython
!pip install jupyter
!pip install matplotlib
!git clone https://github.com/vijendra1125/Tensorflow_O
  bject_detection_API-Custom_Faster_RCNN.git
!git clone https://github.com/tensorflow/models.git
!cp -r Tensorflow_Object_detection_API-
  Custom_Faster_RCNN/* .
%set_env PYTHONPATH=/content/models/research:/content/m
  odels/research/slim
# unzip images folder
!unzip ./images.zip
!python xml_to_csv.py
!python generate_tfrecord.py --
  csv_input=images/train_labels.csv --
  image_dir=images/train --output_path=train.record
!python generate_tfrecord.py --
  csv_input=images/test_labels.csv --
  image_dir=images/test --output_path=test.record
```

```
!wget-
  O /content/faster_rcnn_inception_v2_coco_2018_01_28.t
  ar.gz_ http://download.tensorflow.org/models/object_d
  etection/faster_rcnn_inception_v2_coco_2018_01_28.tar
  .gz
!tar xvzf /content/faster_rcnn_inception_v2_coco_2018_0
  1_28.tar.gz_ -C /content/CP
%cd /content/models/research

!protoc object_detection/protos/*.proto --python_out=.
!pip install --upgrade tf-slim
!python object_detection/legacy/train.py --
  train_dir=/content/CP/output --
  pipeline_config_path=/content/faster_rcnn_pipeline.co
  nfig

!python object_detection/legacy/train.py --
  train_dir=/content/CP/output --
  pipeline_config_path=/content/faster_rcnn_pipeline.co
  nfig
!pip uninstall tensorboard-plugin-wit
%load_ext tensorboard
%tensorboard --logdir /content/CP/output
!python object_detection/export_inference_graph.py \
--input_type=image_tensor \
--
  pipeline_config_path=/content/faster_rcnn_pipeline.co
  nfig \
--
  trained_checkpoint_prefix=/content/CP/output/model.ck
  pt-2000 \
--output_directory=/content/IG/tflite \
```

```
--add_postprocessing_op=true
! zip -r -
  X /content/tflite_saved_model.zip /content/IG/tflite/
  saved_model
!ls /content/CP
```

4.2.2.2 Detection programme of TensorFlow- RCNN platform

The programme developed for object detection is shown below.

```
import tensorflow as tf
tf.__version__
!cp /content/drive/My\ Drive/Pepper_Project/object_dect
  ion/v1/tflite_saved_model_2.zip /content/
!git clone https://github.com/tensorflow/models.git
%set_env PYTHONPATH=/content/models/research:/content/m
  odels/research/slim
%cd /content/models/research
!protoc object_detection/protos/*.proto --python_out=.
import numpy as np
import os
import six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import zipfile

from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image
from IPython.display import display
!pip install --upgrade tf-slim
from object_detection.utils import ops as utils_ops
```

```
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils
  as vis_util
import pathlib
# patch tf1 into `utils.ops`
utils_ops.tf = tf.compat.v1

# Patch the location of gfile
tf.gfile = tf.io.gfile
!unzip /content/tflite_saved_model_2.zip -
  d /content/tflite_saved_model
def load_model(model_name):
    # base_url = 'http://download.tensorflow.org/models
  /object_detection/'
    # model_file = model_name + '.tar.gz'
    # model_dir = tf.keras.utils.get_file(
    #     fname=model_name,
    #     origin=base_url + model_file,
    #     untar=True)
    model_dir = model_name
    model_dir = pathlib.Path(model_dir)/"saved_model"
    print(model_dir)
    model = tf.saved_model.load(export_dir=str(model_di
  r), tags=None)
    model = model.signatures['serving_default']

    return model
model = load_model('/content/tflite_saved_model/content
  /IG/tflite/')
print(model.inputs)
PATH_TO_LABELS = '/content/label_map.pbtxt'
```

```
category_index = label_map_util.create_category_index_f
  rom_labelmap(PATH_TO_LABELS, use_display_name=True)

def run_inference_for_single_image(model, image):
    image = np.asarray(image)
    # The input needs to be a tensor, convert it using
  `tf.convert_to_tensor`.
    input_tensor = tf.convert_to_tensor(image)
    # The model expects a batch of images, so add an ax
  is with `tf.newaxis`.
    input_tensor = input_tensor[tf.newaxis,...]

    # Run inference
    output_dict = model(input_tensor)

    # All outputs are batches tensors.
    # Convert to numpy arrays, and take index [0] to re
  move the batch dimension.
    # We're only interested in the first num_detections
  .
    num_detections = int(output_dict.pop('num_detection
  s'))
    output_dict = {key:value[0, :num_detections].numpy(
  )
                   for key,value in output_dict.items()
  }
    output_dict['num_detections'] = num_detections

    # detection_classes should be ints.
    output_dict['detection_classes'] = output_dict['det
  ection_classes'].astype(np.int64)
```

```
    # Handle models with masks:
    if 'detection_masks' in output_dict:
        # Reframe the the bbox mask to the image size.
        detection_masks_reframed = utils_ops.reframe_bo
  x_masks_to_image_masks(
                                        output_dict['detect
  ion_masks'], output_dict['detection_boxes'],
                                        image.shape[0], ima
  ge.shape[1])
        detection_masks_reframed = tf.cast(detection_ma
  sks_reframed > 0.5, tf.uint8)
        output_dict['detection_masks_reframed'] = detec
  tion_masks_reframed.numpy()

    return output_dict
from google.colab.patches import cv2_imshow
import numpy as np
import cv2

from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

def take_photo(filename='/content/h8.jpg', quality=0.8)
  :
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);
```

```
    const video = document.createElement('video');
    video.style.display = 'block';
    const stream = await navigator.mediaDevices.getUs
erMedia({video: true});

    document.body.appendChild(div);
    div.appendChild(video);
    video.srcObject = stream;
    await video.play();

    // Resize the output to fit the video element.
    google.colab.output.setIframeHeight(document.docu
mentElement.scrollHeight, true);
    // Wait for Capture to be clicked.
    await new Promise((resolve) => capture.onclick =
resolve);

    const canvas = document.createElement('canvas');
    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;
    canvas.getContext('2d').drawImage(video, 0, 0);
    stream.getVideoTracks()[0].stop();
    div.remove();
    return canvas.toDataURL('image/jpeg', quality);
  }
  ''')
display(js)
data = eval_js('takePhoto({})'.format(quality))
binary = b64decode(data.split(',')[1])
with open(filename, 'wb') as f:
  f.write(binary)
return filename
```

4.3 Development of Machine Vision System

Based on the data obtained for the physical parameters of matured pepper, a computer assisted programme and then a machine vision system for identifying matured black pepper spikes were developed. The whole setup comprised of a DSI display unit, Raspberry Pi 4 as a processor and a webcam as sensor as shown in Plate 4.2.

The system has feasibility to a 1-meter length to the surroundings. The monitor provides smooth and high-resolution display for inspection and execution of the programme. The system as a whole is well suited for a robotic pepper harvester or to a grading system.

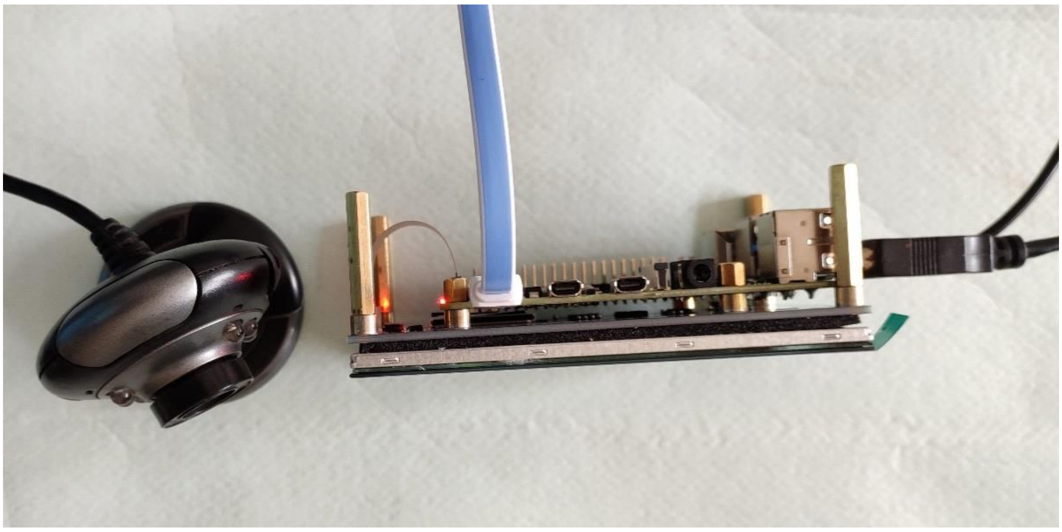

Plate 4.2 Machine vision system

4.4 Performance Evaluation of the Machine Vision System

The developed machine vision system was evaluated for their performance in two different platforms, are discussed below and the sample calculations are shown in appendix.

4.4.1 Sensitivity

The machine vision system developed for identifying matured black pepper spikes in OpenCV- Haar Cascade platform and Tf-RCNN platform were evaluated for their performance. Then sensitivity was calculated from the readings, for both the platforms. The values obtained for the platforms are shown in Figure 4.1, Figure 4.2 and Figure 4.3.

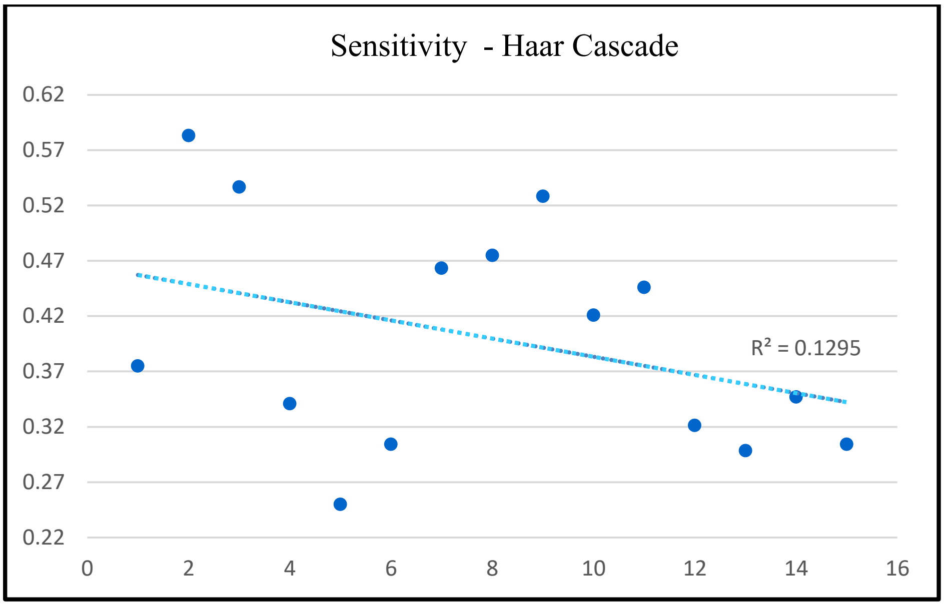

Figure 4.1 Sensitivity of Haar Cascade platform

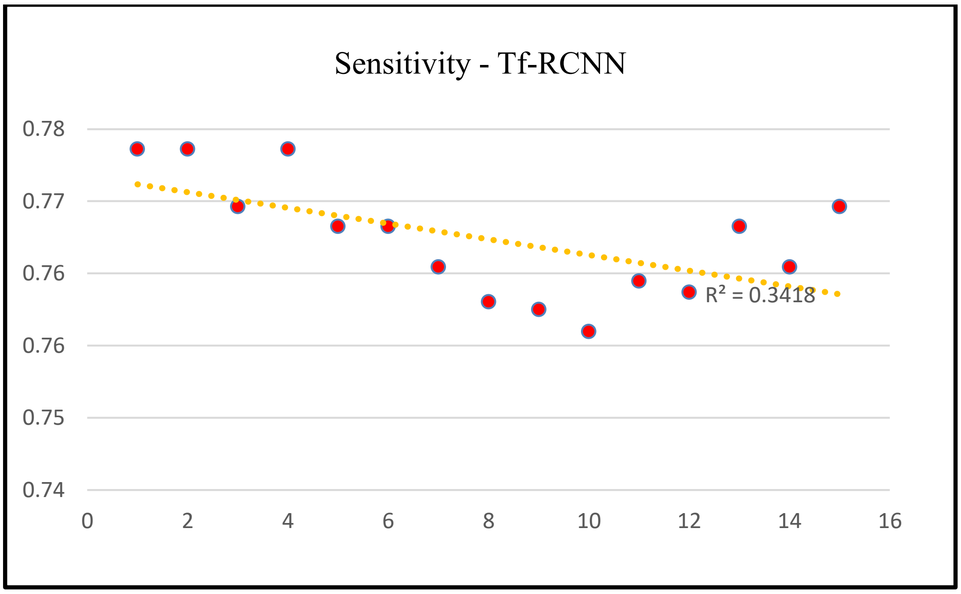

Figure 4.2 Sensitivity of Tf-RCNN platform

From the Figure 4.1 it is clear that the sensitivity of the Haar Cascade platform is less, ranging from 0.25 to 0.58. This low sensitivity affects the model, resulting in low accuracy and poor detection. It is revealed from the table that this platform have an average sensitivity of 0.39. In a similar study conducted by Loresco *et al.* (2018) the average sensitivity obtained in their evaluation is 0.24. This system has a higher sensitivity compared to a previous study. Also from the Figure 4.1, it can be seen that in the scatter plot, the set of data points do not fit to the best curve. The maximum value, 0.58 and minimum value 0.27 is outlying from the mean value. And the R^2 value reveals that only 12% of the data fit to the curve. This proves that only 12% of sensitivity values of this platform is consistent.

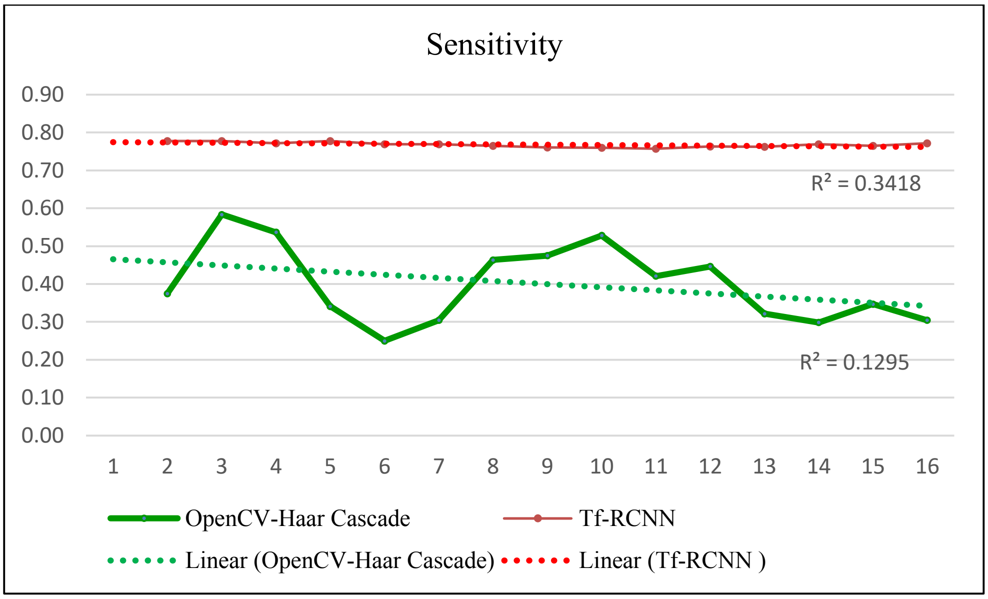

Figure 4.3 Comparison of platforms

Table 4.2 Analysis of variance for testing effect of different open source platforms on sensitivity

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Within treatment	0.073	14.000	0.005	1.019	0.486	2.484
Between treatment	1.059	1.000	1.059	205.973	0.000	4.600
Error	0.072	14.000	0.005			
Total	1.205	29.000				

In the Figure 4.2, it is clear that the Tf-RCNN platform has better sensitivity ranging from 0.77 to 0.78. As it shows higher sensitivity values, this platform have more accuracy in detection. The average sensitivity of this platform is found to be 0.78. In a similar study conducted by Tu *et al.* (2018) the average sensitivity value is obtained as 0.84. Also from the graphical distribution of the values, the R^2 value reveals that 34% of the data values fit to the curve and are consistent in nature.

In a study done by Gan *et al*. (2018), for a system for detecting green citrus fruits by a programme coded in python and using colour feature, they obtained a sensitivity of 78.1%.

From Figure 4.3, a comparative analysis of sensitivity of two platforms can be seen. It is found that sensitivity of OpenCV-Haar cascade platform is lesser than Tf-RCNN platform.

From the Table 4.2, from the P-value it is revealed that there is an effect on sensitivity in the two platforms and have significant difference between them.

4.4.2 Specificity

The performance of the developed OpenCV - Haar Cascade platform and Tf-RCNN platform for identifying matured black pepper spikes were evaluated for the parameter, specificity. The values obtained in the performance evaluation of the platforms is shown in the below Figure 4.4, Figure 4.5 and Figure 4.6.

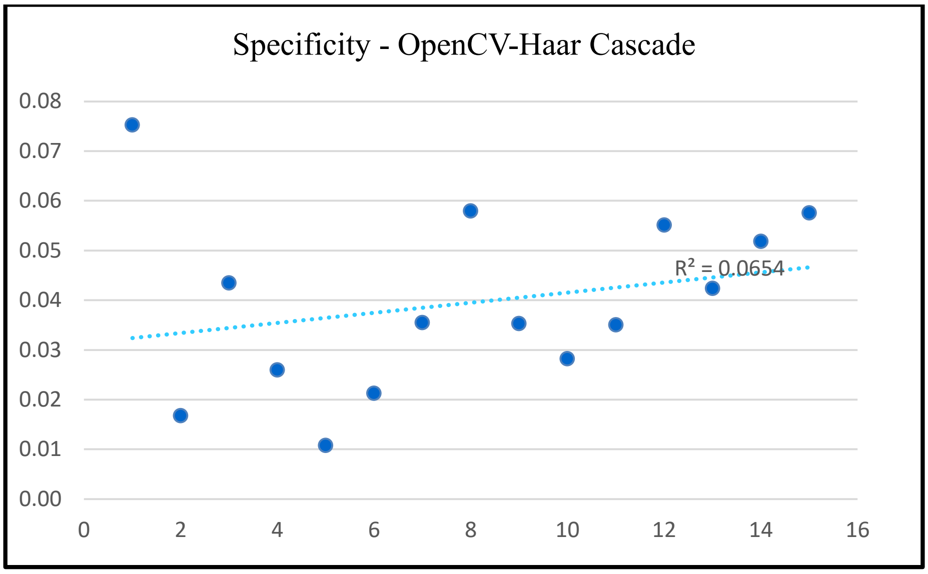

Figure 4.4 Specificity of OpenCV-Haar-cascade platform

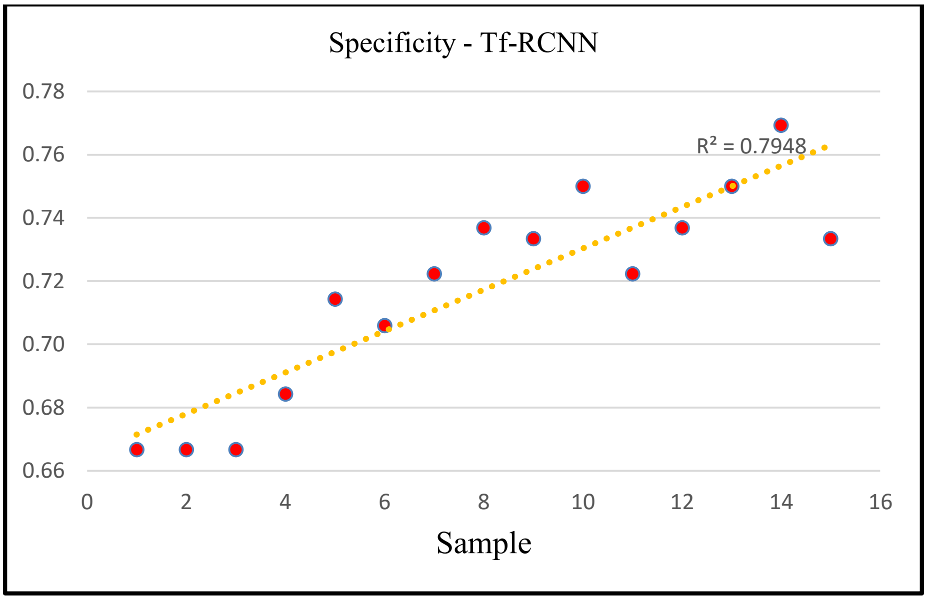

Figure 4.5 Specificity of Tf-RCNN platform

From the Figure 4.4, it is clear that the specificity values of the Haar-Cascade platform is very low, ranging from 0.01 to 0.08. This low specificity can affect the detection of the platform, causing poor results in harvest, by the robotic harvester based on this platform. The average specificity of this platform is found to be 0.04. From a similar study done by Loresco *et al.* (2018) the average specificity value was found as 0.07. So the developed system also have similar result. From the scatter plot of the values, the R^2 value of the plot shows that only 6% of the specificity values are consistent. Further, it can also be observed that the maximum and minimum values of specificity diverged a lot from the mean value.

From the Figure 4.5 it is seen that the Tf-RCNN platform has high values for specificity which ranged from 0.63 to 0.77. The robotic harvester based on this platform will be able to have less harvesting loss. The average specificity of this platform is 0.71. The specificity values also fit to the best curve in the graphical distribution, proving its consistency. From the scatter plot, the R^2 value shows that 79% of the data fit to the curve.

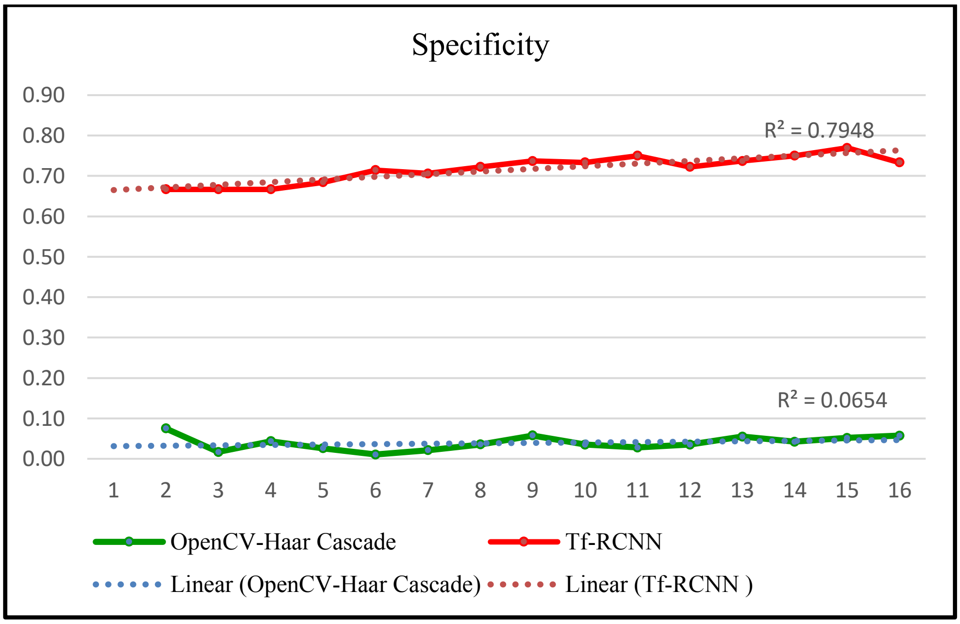

Figure 4.6 Comparison of platforms

Table 4.3 Analysis of variance for testing effect of different open source platforms on specificity

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Within treatment	0.018	14.000	0.001	1.302	0.314	2.484
Between treatment	3.377	1.000	3.377	3382.497	0.000	4.600

Error	0.014	14.000	0.001			
Total	3.409	29.000				

From Figure 4.6, a comparative analysis of specificity of two platforms can be seen. It is found that specificity of Tf-RCNN platform is higher than OpenCV-Haar cascade platform.

From the Table 4.3, from the P-value it is revealed that there is an effect on specificity in the two platforms and have significant difference between them.

4.4.3 Accuracy

The performances of the both platforms were evaluated for the parameter, accuracy. The different values obtained in the performance evaluation is shown below Figure 4.7 and Figure 4.8.

From the Figure 4.7 it is clear that the accuracy of the OpenCV-Haar cascade platform is low ranging from 0.05 to 0.17. It had an average value of 0.13. This platform had uncertain detections, hence has less accuracy. In the similar study done by Loresco *et al.* (2018), the average accuracy obtained is 0.14. This shows the developed system also has similar result. From the scatter plot, it can be seen that the readings do not fit to the best curve. And from the R^2 value of the plot, it is seen that only less than 5% of data are consistent in nature.

From the Figure 4.8, the accuracy of the Tf-RCNN platform ranged from 0.74 to 0.77. It has an average accuracy of 0.75. Zhang *et al.* (2020) did a similar study and obtained an average accuracy of 0.73. So the developed system has a better accuracy. And was accurate in identifying matured black pepper spikes. Also from the scatter plot by the R^2 value of the plot, it shows that 76% of data points are fitting the curve and are consistent.

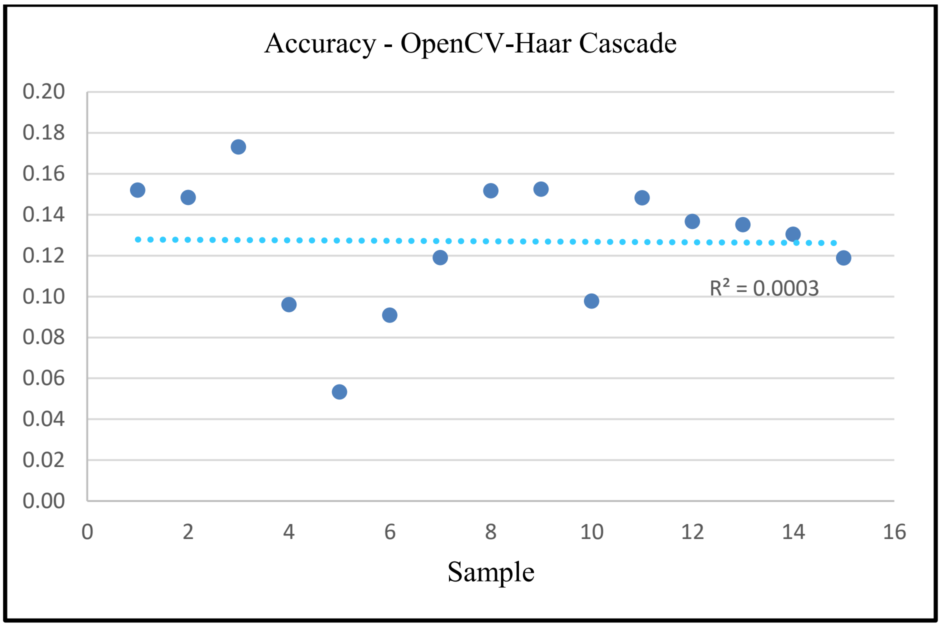

Figure 4.7 Accuracy of OpenCV-Haar Cascade platform

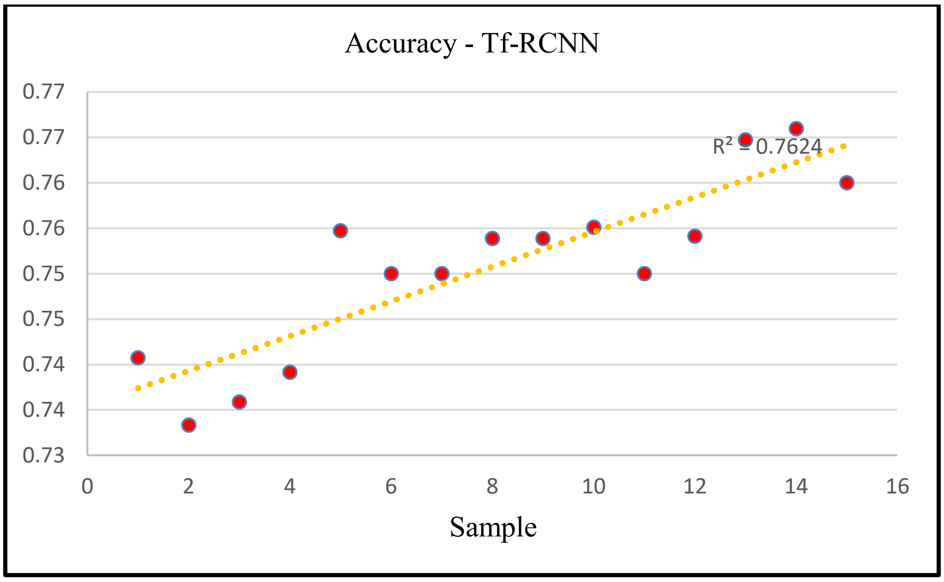

Figure 4.8 Accuracy of Tf-RCNN platform

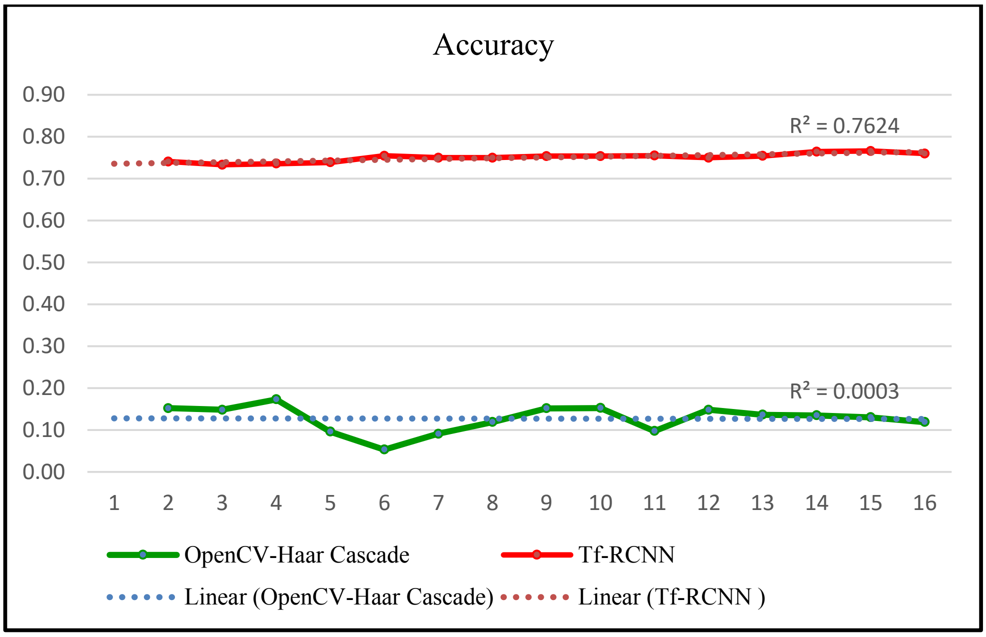

Figure 4.9 Comparison of platforms

Table 4.4 Analysis of variance for testing effect of different open source platforms on accuracy

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Within treatment	0.007	14.000	0.000	0.656	0.780	2.484
Between treatment	2.952	1.000	2.952	4011.410	0.000	4.600
Error	0.010	14.000	0.001			
	2.969	29.000				

From Figure 4.9, a comparative analysis of sensitivity of two platforms can be seen. It is found that accuracy of Tf-RCNN platform is higher than OpenCV-Haar cascade platform.

From the Table 4.4, from the P-value it is revealed that there is an effect on accuracy in the two platforms and have significant difference between them.

4.4.4 Time taken for detection

For the performance evaluation of the platforms, the time taken for detection was also measured and presented in Figure 4.10, Figure 4.11 and Figure 4.12.

From the Figure 4.10, it is seen that the time taken for detection in OpenCV-Haar-Cascade platform is less that ranged from 0.02 to 0.029 seconds. This platform had a smooth flow of video during the execution of programme, and no lag of time was experienced, clearly proving the readings in Figure 4.5. The average time for detection in this was found as 0.024 seconds. In a similar study conducted by Kumar *et al.* (2020) the average time taken for detection is 1.8 seconds. So this system took lesser time for detection. From the graphical scatter plot of values, from the R^2 value it is also revealed that only 7% of readings fit to the best curve and only they are consistent in nature in OpenCV-Haar-Cascade.

From the Figure 4.11, it is observed that the time for detection in Tf-RCNN is higher ranging from 0.41 to 0.43 seconds. This time duration was due to the video lag in the execution of programme. This was because of the heavy size of the platform. The average time for detection is found as 0.42 seconds. Zhang *et al.* (2020) in a similar study has obtained an average time of detection as 0.45 seconds. So the developed model has taken much lesser time for detection. From the graphical plot, the R^2 value reveals that only less than 5% of the data fit to the curve and are consistent.

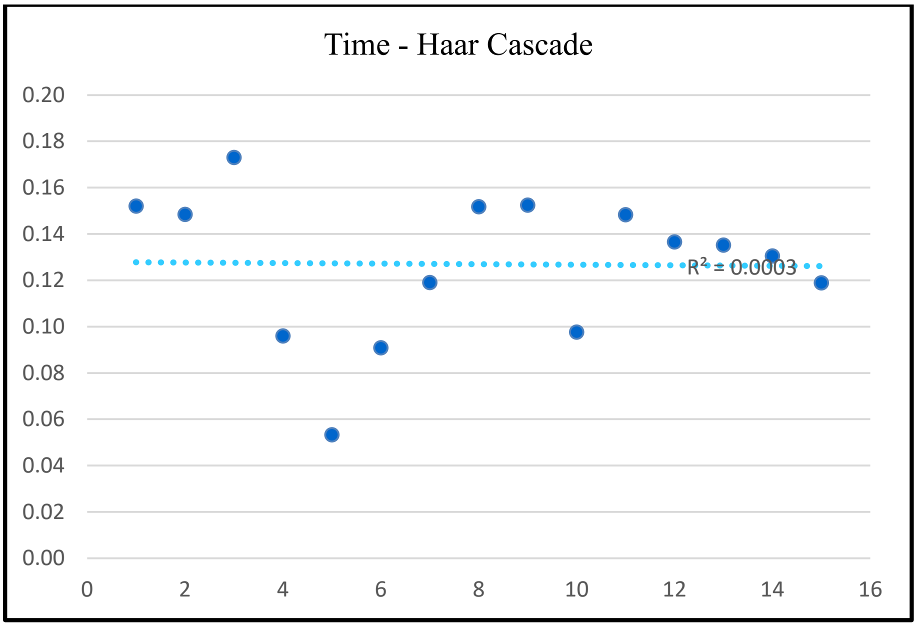

Figure 4.10 Time taken for detection in Haar-Cascade platform

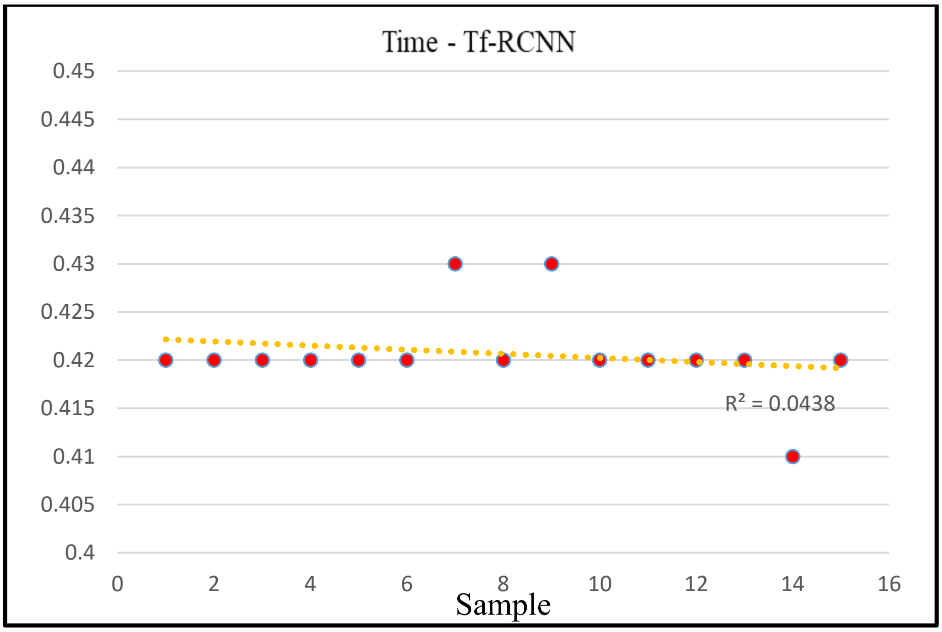

Figure 4.11 Time taken for detection in Tf-RCNN platform

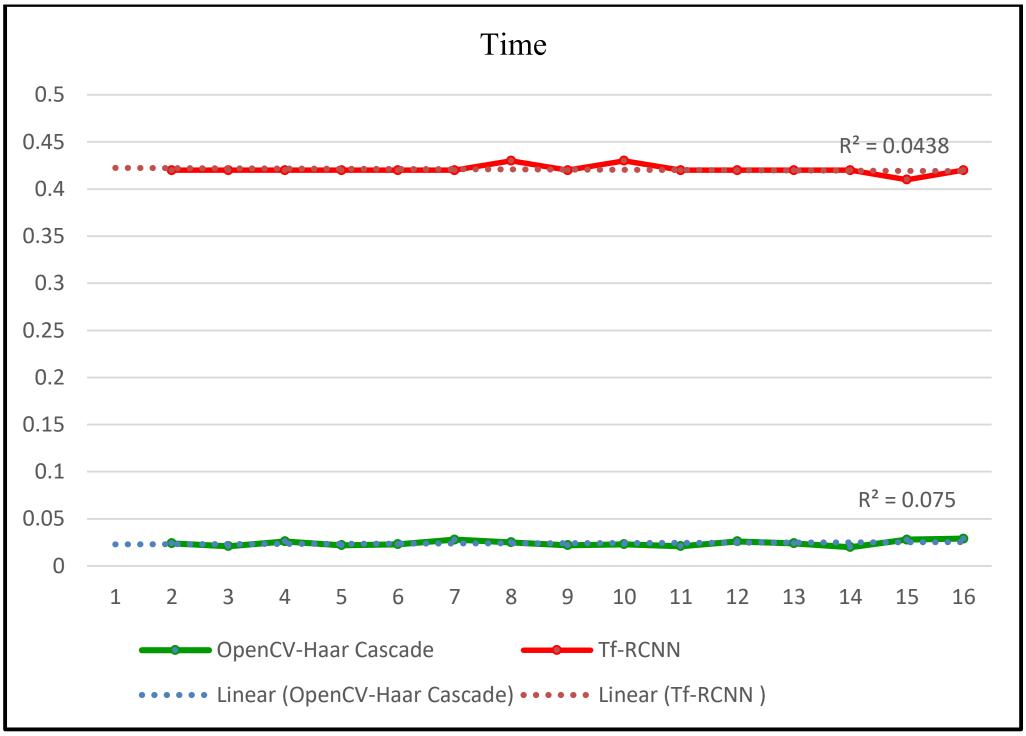

Figure 4.12 Comparison of platforms

Table 4.5 Analysis of variance for testing effect of different open source platforms on time for detection

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Within treatment	0.000	14.000	0.000	0.660	0.777	2.484
Between treatment	1.179	1.000	1.179	67979.942	0.000	4.600
Error	0.000	14.000	0.000			
Total	1.180	29.000				

From Figure 4.12, a comparative analysis of average time taken for detection of two platforms can be seen. It is found that time taken for OpenCV-Haar cascade platform is lesser than Tf-RCNN platform.

From the Table 4.5, from the P-value it is revealed that there is an effect on time in the two platforms and have a significant difference between them.

From the performance evaluation of the platforms, it is clear that the platform with TensorFlow library and Faster-RCNN as classifier, achieved the best performance on evaluation, despite the video lag. The Tf-RCNN platform was effective in detecting the matured black pepper spikes, which was inadequate in the OpenCV - Haar cascade platform. Also the Tf-RCNN platform was capable of rejecting the negative objects correctly than the other one. The two main objectives of an ideal machine vision system are; to correctly identify the positive object and to be efficient in rejecting the negative class. Both these were achieved by the developed system. The only flaw found in the Tf-RCNN was the time taken for detection and in order to overcome this, it is recommended to use high RAM computation systems (Zhang *et al.,* 2020). The images used for OpenCV-Haar cascade training were resized to very low resolution, because if high resolution images were used, the training process will be prolonged (Zhang *et al.,* 2020). This low resolution images would have affected the image feature extraction and so the less accuracy of this platform was obtained.

The Tf-RCNN platform for detection has turned to have the consistent performance in the machine vision system evaluation. So considering the whole machine vision system, it had a successful functioning and good performance to its intended use. Thus the machine vision system of the Tf-RCNN platform based computer assisted programme should be preferred for the future development of a robotic pepper harvester or any other black pepper grading systems and for further studies.

Summary and Conclusions

Chapter V

SUMMARY AND CONCLUSIONS

A machine vision system to identify matured black pepper spike was developed at KCAET, Tavanur. The development procedure and programming part of the research was completely carried out at KCAET, Tavanur. Two computer assisted programme were developed using OpenCV – Haar Cascade platform and TensorFlow – faster RCNN platform. The two programmes were subjected to laboratory evaluation using the developed machine vision system with 15 video clips having 40 images.

The computer assisted programme was coded in Python language as it was prominent in all the latest computer languages. In order to identify the matured black pepper spike, the main two features adopted were the shape and colour. For identifying matured black pepper spikes, physical properties like colour, sphericity, length of spikes, diameter of spikes and diameter of berries were studied. The samples for study was taken from KCAET, Tavanur, Randathani, Malappuram and Kattappana, Idukki. Shape feature would be able to distinguish the pepper spike and colour could decide the matured ones. Two different varieties of pepper were considered for the study; *Karimunda* and *Panniyur 1*.

Colour was measured in terms of RGB value. This observation contributed the range of colour value that should be there for matured pepper spike and can be used as a threshold for separating them. From the study, the colour values for *Karimunda* was obtained as (20, 39, 3) - (255, 254, 111) and in *Panniyur 1* value range was (35, 54, 10) - (255, 240, 100). Sphericity for *Karimunda* variety was obtained as 0.71 and for *Panniyur 1* was 0.44. In length of pepper spikes, average length of spike with peduncle in *Karimunda* was 10.83 cm and in *Panniyur 1* was 13.64 cm. Average diameter of spikes, for *Karimunda* was obtained as 1.08 cm and in *Panniyur 1* as 1.31 cm. And average diameter of berries for *Karimunda* was found as 0.42 cm and for *Panniyur 1* as 0.59 cm.

The first platform used is OpenCV - Haar Cascade platform. Images of matured pepper spike at different maturity stages and negative images were collected for dataset. The number of images for training has a pronounced impact on the detection accuracy of the platform. The programme was first created based on OpenCV as the main library and Haar-cascade as classifier. OpenCV is an image processing library and Haar-Cascade is an image processing algorithm widely used by programmers. The dataset comprised of 400 positive images and 400 negative images. The positive images undergone pre-processing steps and then trained for 20 stages. The training and testing was done using an OpenCV Cascade Trainer Tool.

In the OpenCV-Haar Cascade platform, more than, detecting the positive objects, it was falsely detecting negative objects. That is the false positive rate was higher than true positive rate. It had 0.41 of sensitivity, 0.04 of specificity and 0.13 accuracy. It took only 0.024 seconds of average time for detection. In the scatter plot of data points of the three parameters, it is seen that the data points are not consistent in nature. So it was comprehended that, the developed OpenCV- Haar Cascade platform could not achieve the required accuracy and precision in detection.

In the TensorFlow – Faster RCNN platform, the main programming library used is TensorFlow which is a latest machine learning library, generated by Google in 2015, along with a different classifier; Faster-RCNN. Faster RCNN is a new division from the RCNN which is a widely used Neural Network. Tf-RCNN platform was trained for fewer number of images for detection. The pre-processing of images was also easier in this platform. The images acquired were annotated for xml files and then used for training. The training was carried out in Google Colaboratory. Using the images data, the pre-trained model was subjected to fine tuning, so as to train it with matured pepper spike. The training was done for 2000 stages.

The Tf-RCNN platform was then tested and evaluated in Jupyter Notebook. This platform was also evaluated as same as that of the OpenCV-Haar cascade platform. This platform had a sensitivity of 0.77, specificity of 0.72 and accuracy of 0.75. But it took an average time of 0.42 seconds for detection. In the scatter plot of data points of the three parameters, it is seen that the data points are consistent in nature. From the results, it was distinct that this platform has the better detection accuracy. This platform was efficient in recognizing and detecting matured black pepper spikes as it was revealing in its resulted sensitivity and accuracy values. Also it has similar results in specificity; it was good at rejecting negative class.

In the comparative study of the two platforms, it is seen that the number of images required for Tf-RCNN platform is less than that used in OpenCV-Haar-Cascade, the pre-processing steps of Tf-RCNN is easier, it does not require editing and noise removal steps unlike Haar-Cascade platform, and in the detection also the features of the pepper spikes are clearly extracted and distinguished by Tf-RCNN platform more clearly than OpenCV-Haar Cascade platform. Tf-RCNN platform had higher sensitivity, specificity and accuracy values when compared with OpenCV – Haar Cascade. The time lag in the Tf-RCNN platform was experienced because of the heavy sized CNN architecture model, and it could be avoided by a different model.

It was proved that the OpenCV-Haar Cascade and Tf-RCNN platforms has a significant difference, statistically. The two-way ANOVA was carried out for 15 replications of specificity, sensitivity, accuracy and time for the platforms with a 5% level of significance. Both the F value and P value analysis confirmed that both the platforms have significant difference.

As a modification if the image acquisition camera was replaced using a 3-D camera, more additional information like spatial features and orientation from the images could also be extracted. This can increase the feature extraction more precisely and thus the detection accuracy.

Considering the fact that higher sensitivity, specificity and accuracy is for Tf-RCNN, it is concluded that TensorFlow – faster RCNN is better than OpenCV- Haar Cascade platform for identifying matured black pepper spikes.

Reference

REFERENCE

Ahmadabadi, H. N., Omid, M., Mohtasebi, S. S., and Firouz, M. S. 2017. Design, development and evaluation of an online grading system for peeled pistachios equipped with machine vision technology and support vector machine. *Inf. Processing Agric.* 4: 333-341.

[Anonymous]. 2019. Available:https://www.indiaagristat.com/[13 March 2019]

Arakeri, M. P. and Lakshmana. 2016. Computer Vision Based Fruit Grading System for Quality Evaluation of Tomato in Agriculture industry. *7th International Conference on Communication, Computing and Virtualization*, Procedia Computer Science, pp. 426-433.

Azarmdel, H., Jahanbakhsh, A., Mohtasebi, S. S., and Muñoz, A. R. 2020. Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM). *Postharvest Biol. Technol.* 166: 111201.

Barbedo, J. G. A. 2014. Using digital image processing for counting whiteflies on soybean leaves. *J. Asia-Pac. Entomol.* 17: 685-694.

Barnea, E., Mairon, R., and Shahar, O. B. 2016. Colour-agnostic shape-based 3D fruit detection for crop harvesting robots. *Biosystems Engg.* pp. 1-14.

Batchelo, B. G. and Braggins, D. W. 1992. Commercial Vision Systems. In: Computer Vision: Theory and Industrial Applications. Springer. 405p.

Beyaz, A., Gila, D. M. M., Ortega, J. G., and García, J. G. 2019. Olive fly sting detection based on computer vision. *Postharvest Biol. Technol.* 150: 129-136.

Blok, P. M., Barth, R., and Berg, W. V. D. 2016. Machine vision for a selective broccoli harvesting robot. *IFAC-PapersOnLine*, pp. 066-071.

Cavallo, D. P., Cefola, M., and Pace, B. 2019. Non-destructive and contactless quality evaluation of table grapes by a computer vision system. *Comput. Electr. Agric.* 156: 558-564.

Chaaro, L. and Anton , L.M. 2020. Crop and weed detection using image processing and deep learning techniques. Bachelor Degree Project in Production Engineering Thesis, University of Skovde.

Chen, S. Xiong, J., and Guo, W. 2019. Colored rice quality inspection system using machine vision. *J. Cereal Sci.* 88: 87-95.

Chung, C. L., Huang, K. J., Chen, S. Y. and Lai, M. H. 2016. Detecting Bakanae disease in rice seedlings by machine vision. *Comput. Electr. Agric.* 121: 404-411.

Clement, J., Novas, N., Gazquez, J. A., and Agugliaro, F. M. 2013. An active contour computer algorithm for the classification of cucumbers. *Comput. Electr. Agric.* 92: 75-81.

Dorj, U. O., Lee, M., and Yun, S. s. 2017. Yield estimation in citrus orchards via fruit detection and counting using image processing. *Comput. Electr. Agric.* 140: 103-112.

Dutta, M. K., Sengar,N., and Minhas, N. 2016. Image processing based classification of grapes after pesticide exposure. *LWT - Food Sci. Technol.* 72: 368-376.

Fashi, M., Naderloo, L., and Javadikia, H. 2019. The relationship between the appearance of pomegranate fruit and colour and size of arils based on image processing. *Postharvest Biol. Technol.* 154: 52-57.

Fu, L., Duan, J., and Zou, X. 2019. Banana detection based on colour and texture features in the natural environment. *Comput. Electr. Agric.* 167: 105057.

Gan, H., Lee, W.S. and Alchanatis, V. 2018. Immature green citrus fruit detection using colour and thermal images. *Comput. Electr. Agric.* 152: 117-125.

Gao , M. and Lu, T. F., 2006. Image Processing and Analysis forAutonomous Grapevine Pruning. *Proceedings of the 2006 IEEE.* IEEE, pp. 25-28.

George, M. 2015. Multiple Fruit and Vegetable Sorting System Using Machine Vision. *Int. J. Adv. Technol.* 6(1): 1000142.

Goh, G. 2017. Why momentum really works. [Online]. Distill. Available at: http://distill.pub/2017/momentum

Gongal, A., Karkee, M., and Amatya, S. 2018. Apple fruit size estimation using a 3D machine vision system. *Info. Processing Agric.* 5: 498-503.

Guanjun, B., shibo, C., and Liyong, Q. 2016. Multi-template matching algorithm for cucumber recognition in natural environment. *Comput. Electr. Agric.* 127: 754-762.

Habib, M. T., Majumdar, A., Jakaria, A. Z. M., and Akter, M. 2020. Machine vision based papaya disease recognition. *J. King Saud University – Comput. Inf. Sci.* 32: 300-309.

Healthline. 2019. Healthline [online]. Available: https://www.healthline.com/nutrition/black-pepper-benefits#2.-Has-anti-inflammatory-properties. [21 Mar. 2019]

Gao, H. 2017. Medium. [Online]. Available at: https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8

Ireri, D., Belal, E., okinda, C., and Makange, N. 2019. A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing. *Artif. Intell. Agric.* 2: 28-37.

Ji, W., Zhao, D., Cheng, F., and Xu, B. 2012. Automatic recognition vision system guided for apple harvesting robot. *Comput. Electr. Engg.* 38: 1186-1195.

Jun, Q., Sasao, A., Shibusawa, S., and Kondo, N. 2012. Extracting external features of sweet peppers using machine vision system on mobile fruits grading robot. *Int. J. Food Engg.* 8(3).

Kane, K. E. and Lee, W. S. 2007. Multispectral Imaging for In-field Green Citrus Identification. *2007 ASABE Annual International Meeting.* ASABE, pp.073025

Ke-ling, T., Lin-juan, L., and Li-ming, Y. 2018. Selection for high quality pepper seeds by machine vision and classifiers. *J. Integrative Agric.* 17(9): 1999-2006.

Khazaei, N. B. and Maleki, M. R. 2017. A robust algorithm based on colour features for grape cluster segmentation. *Comput. Electr. Agric.* 142: 41-49.

Kondo, N., Yamamoto, K., and Shimzu, H. 2009. A Machine Vision System for Tomato Cluster Harvesting Robot. *Engg. Agric. Environ. Food.* 2(2): 60-65.

Kumar, R. S., Cherukula, V., and Saxena, P. 2020. *Design of Pepper Harvesting Device,* s.l.: s.n.

Kurtulmus, F., Lee, W. S., and Vardar, A. 2011. Green citrus detection using ‘eigenfruit’, colour and circular Gabor texture features under natural outdoor conditions. *Comput. Electr. Agric.* 78: 140-149.

Li, H., Lee, W. S., and Wang, K. 2014. Identifying blueberry fruit of different growth stages using natural outdoor colour images. *Comput. Electr. Agric.* 106: 91-101.

Lin, G., Tang, Y., and Zou, X. 2019. In-field citrus detection and localisation based on RGB-D image analysis. *Biosystems Engg.* 186: 34-44.

Mahajan, S., Das, A., and Sardana, H. K. 2015. Image acquisition techniques for assessment of legume quality. *Trends Food Sci. Technol.* 42(2): 116-133.

Maldonado Jr., W. and Barbosa, J. C. 2016. Automatic green fruit counting in orange trees using digital images. *Comput. Electr. Agric.* 127: 572-581.

Malekabadi, A. J., Khojastehpour, M., Emadi, B., and Golzarian, M. R., 2017. Development of a machine vision system for determination of mechanical properties of onions. *Comput. Electr. Agric.* 141: 131-139.

Meixner, M. 2019. 11 Science-Backed Health Benefits of Black Pepper. Available: https://www.healthline.com/nutrition/black-pepper-benefits [21 March 2019]

Mim, F. S., Galib, S. M., Hasan, M. F., and Jerin, S. A. 2018. Automatic detection of mango ripening stages – An application of information technology to botany. *Scientia Horticulturae.* 237: 156-163.

Moallem, P., Serajoddin, A., and Pourghassem, H. 2017. Computer vision-based apple grading for golden delicious apples based on surface features. *Inf. Processing Agric*. 4: 33-40.

Mohammadi, V., Kheiralipour, K., and Varnamkhasti, M. G. 2015. Detecting maturity of persimmon fruit based on image processing technique. *Scientia Horticulturae.* 184: 123-128.

Momin, M. A., Yamamoto, K., and Miyamoto, M. 2017a. Machine vision based soybean quality evaluation. *Comput. Electr. Agric.* 140: 452-460.

Momin, M. A., Rahman, M. T., Sultana, M. S., and Igathinathane, C. 2017b. Geometry-based mass grading of mango fruits using image processing. *Inf. Processing Agric.* 4: 150-160.

Monhollen, N. S., Shinners, K. J., Friede, J. C., and M.C, Eduardo. 2020. In-field machine vision system for identifying corn kernel losses. *Comput. Electr. Agric.* 174: 105496.

Mordvintsev, A. and Abid, K. 2013. Face Detection using Haar Cascades. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#:~:text=Haar%2Dcascade%20Detection%20in%20OpenCV,-OpenCV%20comes%20with&text=If%20you%20want%20to%20train,use%20OpenCV%20to%20create%20one.&text=OpenCV%20already%20contains%20many%20pre,%2Fdata%2Fhaarcascades%2F%20folder.

Nasiri, A., Garavand, A. T., and Zhang, Y. D. 2019. Image-based deep learning automated sorting of date fruit. *Postharvest Biol. Technol.* 153: 133-141.

Nguyen, T. T., Vandevoorde, K., Wouters, N., and Kayacan, E. 2016. Detection of red and bicoloured apples on tree with an RGB-D camera. *Biosystems Engg.* 2016: 1-12.

Ohali, Y. A. 2011. Computer vision based date fruit grading system: Design and implementation. *J. King Saud University – Comput. Inf. Sci.* 23: 29-36.

Patel, H. N., Jain, R. K., and Joshi, M. V. 2011. Fruit Detection using Improved Multiple Features based Algorithm. *Int. J. Comput. Appl.* 13(2): 0975 – 8887.

Pereira, L. F. S., Jr., S. B., Valous, N. A., and Barbin, D. F. 2018. Predicting the ripening of papaya fruit with digital imaging and random forests. *Comput. Electr. Agric.* 145: 76-82.

Rajendra, P., Kondo, N., Ninomiya, K., and Kamata, J. 2009. Machine Vision Algorithm for Robots to Harvest Strawberries in Tabletop Culture Greenhouses. *Engg. Agric. Environ. Food.* 2(1): 24-30.

Rehman, T. U., Zaman, Q. U., and Chang, K. Y. 2019. Development and field evaluation of a machine vision based in-season weed detection system for wild blueberry. *Comput. Electr. Agric.* 162: 1-13.

Roshanianfard, A., Kamata, T., and Noguchi, N. 2018. Performance evaluation of harvesting robot for heavy-weight crops, *IFAC*, pp. 332-338.

Sandoval, J. R. M., Rosa, M. E. M., Ve, M. M., and Sandoval, E. M. 2018. Colour Analysis and Image Processing Applied in Agriculture. In: *Colorimetry Image Processing.* pp. 61-78.

Sharma, P. 2018. A Practical Implementation of the Faster R-CNN Algorithm for Object Detection (Part 2 – with Python codes). [Online] Available: https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/ [June 2020].

Sing, V. 2018. Custom Faster RCNN using TensorFlow Object Detection API [on-line]. Available: https://medium.com/@vijendra1125/tensorflow-api-custom-object-detection-55444de6562d [02 Oct 2018]

Tan, K., Lee, W. S., Gan, H., and Wang, S. 2018. Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. *Biosystems Engg.* 176: 59-72.

Taofik, A., Ismail, N., and Gerhana, Y. A. 2018. Design of Smart System to Detect Ripeness of Tomato and Chili with New Approach in Data Acquisition. *2nd Annual Applied Science and Engg. Conference.* IOP Publishing, pp. 1757-1780.

Thendral, R., Suhasini, A., and Senthil, N. 2014. A Comparative Analysis of Edge and Colour Based Segmentation for Orange Fruit Recognition. IEEE, pp. 463-467.

Thomas, L. and Rajeev, P. 2015. Black pepper. Indian Institute of Spices Research Kozhikode, 21p.

Tu, S., Xue, Y., Zheng, C., and Qi, Y. 2018. Detection of passion fruits and maturity classification using Red-Green-Blue Depth images. *Biosystems Engg.* 175: 156-167.

Vibhute, A. and Bodhe, S. 2012. Applications of Image Processing in Agriculture:A Survey. Int. J. of Comput. Appl.52.

Vithu, P. and Moses, J. A. 2016. Machine vision system for food grain quality evaluation: A Review. *Trends Food Sci. Technol.* 56: 13-20.

Wang, C., Tang, Y., and Zou, X. 2017. A robust fruit image segmentation algorithm against varying illumination for vision system of fruit harvesting robot. *Optik.* 131: 626-631.

Wang, X., Mao, H., Han, X., and Yin, J. 2011. Vision-based judgment of tomato maturity under growth conditions. *Afr. J. Biotechnol.* 10(18): 3616-3623.

Wan, P., Toudeshki, A., Tan, H., and Ehsani, R. 2018. A methodology for fresh tomato maturity detection using computer vision. *Comput. Electr. Agric.* 146: 43-50.

Williams, H. A., Jones, M.H., and Nejati, M. 2019. Robotic kiwifruit harvesting using machine vision, Convolutional neural networks, and robotic arms. *Biosystems Engg.* 181: 140-156.

Yegulalp, S. 2019. What is TensorFlow? The machine learning library explained. Available: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html

Yongsheng, S., Gang, L., and Juan, F. 2015. Location of apples in trees using stereoscopic vision. *Comput. Electr. Agric.* 112: 68-74.

Yun, C., Yalei, W., jian, Z., and Jun, C. 2018. Recognition of the position of Chinese wolfberry branches under the artificial background. *IFAC* PapersOnLine, pp. 321-325.

Yu, Y., Zhang, K., Yang, L., and Zhang, D. 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electr. Agric.* 163.

Zareiforoush, H., Minaei, S., Alizadeh, M. R., and Banakar, A. 2016. Design, development and performance evaluation of an automatic control system for rice whitening machine based on computer vision and fuzzy logic. *Comput. Electr. Agric.* 124: 14-22.

Zhao, Y., Gong, L., and Zhou, B. 2016. Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. *Biosystems Engg.* 148: 127-137.

Zhang, J., Karkee, M., Zhang, Q., Zhang, X., and Yaqoob, M. 2020. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Comput. Electr. Agric.* 0168-1699

Appendices

Appendix I

Cost Analysis

Software Development

Total working hours	=	75 hours
Cost per hour	=	Rs.500
Total Cost	=	500 × 75
	=	Rs. 37,500

Hardware Development

Webcam	=	Rs.700
Raspberry Pi	=	Rs.5500
DSI display	=	Rs. 3500
Accessory hardware	=	Rs. 800
Total cost	=	Rs. 10500

Total Cost of development = Rs. 48,000

Appendix II

Specification of the sensor

Particulars	Specification
Brand	QUANTAM
Manufacturer	QUANTAM, Aspire overseas pvt ltd
Colour	Black and silver
Resolution	500K pixels
Image quality	RGB24 or I420
Item Height	4 mm
Item Width	30 mm
Product Dimensions	0.4 x 3 x 0.4 cm; 200 g
Item model number	QHM495LM

Appendix III

Specifications of processor

Particulars	Specification
Brand	Raspberry Pi
Series	Pi 4, 4GB
Colour	Green
Pi Dimensions	9.6 x 7.3 x 3.2 cm; 80 Grams
RAM Size	4 GB
Maximum Memory Supported	4 GB
Connectivity Type	Wi-Fi, Bluetooth
Operating System	Chrome OS, Windows 10

Appendix IV

Specifications of display unit

Particulars	Specification
Brand	Raspberry Pi LCD Display Module 3
Manufacturer	WaveShare
Touchscreen	TFT
Resolution	320 x 240
Dimensions	7 x 5 x 1.5 cm
Hardware Interface	SPI
Backlight current	TBD
Backlight	LED

Appendix V

Measurement of RGB value

Sl. No	RGB Value (Range)	
	Karimunda	*Panniyur 1*
1	(24, 46, 19) - (251, 174, 172)	(93, 34, 9) – (218, 180, 111)
2	(26, 33, 2) - (255, 193, 174)	(93, 70, 13) - (255, 240, 100)
3	(46, 68, 19) - (255, 224, 196)	(94, 119, 29) - (251, 255, 110)
4	(79, 53, 26) - (255, 199, 169)	(94, 120, 38) - (252, 205, 110)
5	(47, 39, 3) - (255, 230, 202)	(94, 130, 60) - (255, 205, 101)
6	(52, 25, 0) - (255, 184, 192)	(94, 131, 63) - (255, 135, 101)
7	(69, 75, 40) - (255, 195, 186)	(94, 49, 40) - (255, 135, 125)
8	(105, 128, 79) -(255, 183, 175)	(94, 91, 21) - (255, 156, 110)
9	(40, 76, 32) - (255,225,254)	(95, 120, 19) - (255, 220, 110)
10	(28, 30, 0) - (255, 198, 185)	(95, 123, 28) - (255, 223, 130)
11	(37, 19, 6) - (255, 195, 186)	(97, 136, 39) - (253, 210, 130)
12	(66, 89, 5) - (255, 179, 187)	(98, 121, 19) - (254, 203, 130)
13	(65, 70, 16) - (255, 182, 173)	(35, 54, 1) - (255, 200, 150)
14	(82, 124, 59) - (253, 163, 138)	(44, 52, 7) - (255, 196, 150)
15	(26, 8, 2) - (255, 237, 187)	(45, 52, 15) - (255, 230, 150)
16	(20, 39, 3) - (255, 191, 185)	(52, 61, 7) - (255, 224, 130)
17	(44, 28, 6) - (255, 213, 206)	(57, 72, 15) - (255, 210, 130)
18	(27, 15, 15) - (255, 188, 230)	(64, 85, 23) - (255, 209, 150)
19	(75, 4, 1) - (255, 198, 111)	(66, 84, 11) - (255, 219, 160)
20	(35, 54, 1) - (255, 241, 199)	(93, 34, 9) - (218, 180, 111)

Appendix VI

Sphericity of pepper berries

Sl No.	Sphericity	
	Karimunda	*Panniyur 1*
1	0.73	0.36
2	0.69	0.42
3	0.71	0.52
4	0.62	0.47
5	0.71	0.35
6	0.62	0.34
7	0.67	0.41
8	0.73	0.45
9	0.78	0.51
10	0.75	0.32
11	0.78	0.55
12	0.75	0.42
13	0.71	0.53
14	0.74	0.51
15	0.75	0.42
16	0.69	0.38
17	0.69	0.49
18	0.67	0.37
19	0.68	0.46
20	0.70	0.55

Appendix VII

Length of pepper spikes

Sl no.	Length of spikes (cm)			
	With peduncle	Without peduncle	With peduncle	Without peduncle
	Karimunda		*Panniyur 1*	
1	9	8.5	18.5	17
2	9.5	8	12.5	11.5
3	9.5	8.5	17.5	16
4	6	4.5	16	15
5	11	10	18	16.5
6	12	10	9	7
7	11	10	19.5	18
8	12	10.5	17	16
9	9.5	8	16.5	15
10	9	8	16	15
11	11	9	9	8
12	12.5	11	9.5	8.5
13	8	6.5	9.5	8.5
14	13.5	12.5	10.5	9
15	12.5	12	14	12
16	11.5	10.5	9.3	8.3
17	14.5	13	12	10.5
18	13.5	12.5	13	12
19	8.5	8	12.5	11.5
20	12.5	11	13	12

Appendix VIII

Diameter of pepper spikes

Sl no.	Diameter of spike (cm)	
	Karimunda	*Panniyur 1*
1	0.77	1.70
2	1.17	1.70
3	0.80	1.47
4	1.17	0.87
5	1.27	1.37
6	1.10	1.33
7	1.23	1.50
8	0.83	1.27
9	1.20	1.27
10	1.10	1.43
11	1.20	1.33
12	1.10	1.10
13	1.20	1.10
14	1.03	1.20
15	1.00	1.10
16	1.03	1.20
17	0.87	1.30
18	1.27	1.27
19	1.10	1.45
20	1.20	1.35

Appendix IX

Diameter of berries

Sl No.	Diameter of berry (cm)	
	Karimunda	*Panniyur 1*
1	0.45	0.5
2	0.55	0.65
3	0.6	0.55
4	0.55	0.6
5	0.65	0.7
6	0.7	0.65
7	0.7	0.75
8	0.7	0.7
9	0.65	0.7
10	0.7	0.65
11	0.5	0.75
12	0.6	0.5
13	0.6	0.6
14	0.7	0.6
15	0.6	0.7
16	0.65	0.6
17	0.55	0.65
18	0.45	0.55
19	0.5	0.65
20	0.6	0.65

Appendix X

Sample Calculation

1. Sphericity

Diameter of inscribing circle = 11 mm

Diameter of circumscribing circle = 15 mm

$$\text{Sphericity} = \frac{\text{Diameter of inscribing circle}}{\text{Diameter of circumscribing circle}}$$

$$= 11 / 15$$

$$= 0.73$$

2. Sensitivity

Total true positives = 12

Total false negatives = 20

$$\text{Sensitivity} = \frac{\text{Total true positives}}{\text{Total true positives + Total false negatives}}$$

$$= 12 / (12 + 20)$$

$$= 0.375$$

3. Specificity

Total true negatives = 7

Total false positives = 56

$$\text{Specificity} = \frac{\text{Total true negatives}}{\text{Total true negatives + Total false positives}}$$

$$= 7 / (7 + 56)$$

$$= 0.111$$

4. Accuracy

Total True positive = 12

Total true negative = 7

Total TP, TN, FP, FN = 125

$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{\text{True positives} + \text{True negatives} + \text{False positives} + \text{False negatives}}$$

$$\text{Accuracy} = (12 + 7) / 125$$

$$= 0.15$$

Appendix XI
Performance Evaluation of OpenCV - Haar Cascade

Sl No.	OpenCV - Haar Cascade				Sensitivity	Specificity	Accuracy	Time (s)
	TP	TN	FP	FN				
1	12	7	86	20	0.38	0.08	0.15	0.024
2	21	2	117	15	0.58	0.02	0.15	0.021
3	22	5	110	19	0.54	0.04	0.17	0.026
4	15	4	150	29	0.34	0.03	0.10	0.022
5	10	2	183	30	0.25	0.01	0.05	0.023
6	14	3	138	32	0.30	0.02	0.09	0.028
7	19	6	163	22	0.46	0.04	0.12	0.025
8	19	8	130	21	0.48	0.06	0.15	0.022
9	28	6	164	25	0.53	0.04	0.15	0.023
10	16	5	172	22	0.42	0.03	0.10	0.021
11	29	6	165	36	0.45	0.04	0.15	0.026
12	18	7	120	38	0.32	0.06	0.14	0.024
13	20	5	113	47	0.30	0.04	0.14	0.02
14	17	7	128	32	0.35	0.05	0.13	0.028
15	14	8	131	32	0.30	0.06	0.12	0.029

Performance Evaluation of Tf – RCNN

Sl No.	Tf - RCNN				Sensitivity	Specificity	Accuracy	Time (s)
	TP	TN	FP	FN				
1	28	12	6	8	0.78	0.67	0.74	0.42
2	21	12	6	6	0.78	0.67	0.73	0.42
3	27	12	6	8	0.77	0.67	0.74	0.42
4	21	13	6	6	0.78	0.68	0.74	0.42
5	30	10	4	9	0.77	0.71	0.75	0.42
6	30	12	5	9	0.77	0.71	0.75	0.42
7	26	13	5	8	0.76	0.72	0.75	0.43
8	35	14	5	11	0.76	0.74	0.75	0.42
9	38	11	4	12	0.76	0.73	0.75	0.43
10	25	12	4	8	0.76	0.75	0.76	0.42
11	29	13	5	9	0.76	0.72	0.75	0.42
12	32	14	5	10	0.76	0.74	0.75	0.42
13	40	12	4	12	0.77	0.75	0.76	0.42
14	26	10	3	8	0.76	0.77	0.77	0.41
15	27	11	4	8	0.77	0.73	0.76	0.42

Abstract

DEVELOPMENT OF A MACHINE VISION SYSTEM TO IDENTIFY MATURED PEPPER SPIKES

by

MEERA T

(2018-18-014)

ABSTRACT OF THESIS

Submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY

IN

AGRICULTURAL ENGINEERING

(Farm Power and Machinery)

Faculty of Agricultural Engineering and Technology

Kerala Agricultural University

Department of Farm Machinery and Power Engineering

Kelappaji College of Agricultural Engineering and Technology

Tavanur - 679 573, Kerala

2020

ABSTRACT

Black pepper is a perennial crop and one of the most economically significant spices in India. It has a high commercial value in the market all around the world. Its fruit is harvested, dried and powdered for many cuisines and processed for many value added products. Black pepper is a flowering vine growing up to 4 m in height. The berries turns from green to red on maturity and are harvested when it starts to turn red. For achieving good quality and good sized pepper, it should be harvested at its proper matured state. Farmers for their time saving and due to heavy work intensity, harvest almost all the fruits which are in a range of maturity along with the real matured ones. This eventually affects the crop yield and quality. Hence employing an automated identification system in this case would be effective. An application programme interface was developed for this, using the fruit features like the shape, colour and size. By using the machine learning techniques and computer vision technology, two programmes were developed in python language, one using OpenCV library and Haar Cascade classifier, and other platform with TensorFlow as library and faster-RCNN as classifier. Studies were also carried out to analyse the physical properties of black pepper. Using image acquisition, a dataset was created and was used for training and preparation of both the models. The hardware part of the system comprised of a webcam as sensor, Raspberry Pi processor, a RPI display unit and some accessory parts. The hardware and software parts were installed and assembled, and subjected to performance evaluation. It was revealed that the Tf-RCNN platform had better performance and efficiency. The performance evaluation parameters viz., sensitivity, specificity and accuracy values were 78%, 71% and 75% respectively for the second model. It was statistically verified that there is a significant difference between the two platforms and the second model had better consistency.