Journal Article

Detection, Quantification and Classification of Ripened Tomatoes: A Comparative Analysis of Image Processing and Machine Learning

Alam Siddiquee, K N E., Islam, Md. S., Dowla, M Y U., Rezaul, K M. and Grout, V

This article is *accepted for publication* and is published by Institution of Engineering and Technology (IET). The article is available at: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2019.0738

Detection, Quantification and Classification of Ripened Tomatoes: A Comparative Analysis of Image Processing and Machine Learning

Kazy Noor E Alam Siddiquee[1], Md. Shabiul Islam[2], Mohammad Yasin Ud Dowla[3], Karim Mohammed Rezaul[4], Vic Grout[4]

[1]PhD Student, Faculty of Engineering, Multimedia University, Selangor, Malaysia and Assistant Professor, Department of Computer Science and Engineering, University of Science and Technology Chittagong, Chittagong, Bangladesh,

[2]Faculty of Engineering, Multimedia University, Selangor, Malaysia

[3]Department of Computer Science and Engineering, University of Science and Technology Chittagong, Chittagong, Bangladesh

[4]Faculty of Arts, Science and Technology, Wrexham Glyndŵr University, UK

Abstract*—In this paper, specifically for detection of ripe/unripe tomatoes with/without defects in the crop field, two distinct methods are described and compared. One is a machine learning approach, known as 'Cascaded Object Detector' and the other is a composition of traditional customized methods, individually known as 'Colour Transformation', 'Colour Segmentation' and 'Circular Hough Transformation'. The (Viola Jones) Cascaded Object Detector generates 'histogram of oriented gradient' (HOG) features to detect tomatoes. For ripeness checking, the RGB mean is calculated with a set of rules. However, for traditional methods, color thresholding is applied to detect tomatoes either from a natural or solid background and RGB colour is adjusted to identify ripened tomatoes. In this work, Colour Segmentation is applied in the detection of tomatoes with defects, which has not previously been applied under machine learning techniques. The function modules of this algorithm are fed formatted images, captured by a camera mounted on a mobile robot. This robot was designed, built and operated in a tomato field to identify and quantify both green and ripened tomatoes as well as to detect damaged/blemished ones. This algorithm is shown to be optimally feasible for any micro-controller based miniature electronic devices in terms of its run time complexity of $O(n^3)$ for traditional method in best and average cases. Comparisons show that the accuracy of the machine learning method is 95%, better than that of the Colour Segmentation Method using MATLAB. This result is potentially significant for farmers in crop fields to identify the condition of tomatoes quickly.

Keywords: Tomato detection, Quantification of tomatoes, Detection of tomatoes with defects, Circular Hough Transformation, Colour Segmentation

I. INTRODUCTION

The main focus of this paper is on detection of ripe/unripe tomatoes with/without defects in the crop field and in this paper two distinct methods are described and compared. One is the 'Cascaded Object Detector' and the other is a composition of some traditional customized image processing methods.

Image processing methods are gradually improving in terms of their accuracy and faster processing speeds. For a small-scale system, traditional methods are ideal [1]. For measureable shapes of objects, such as circles, the Circular Hough Transformation (CHT) is suitable [2] [3] [4] [5] [6]. If the image requires greater component analysis, then thresholding and Colour Segmentation may provide better image information [7]. However, some additional methods such as classifiers [8] [9] [10] [11], filters [12], support vector machines [13] or neural networks [13], are also available, often providing better outcomes in the detection of fruit or crops within fields.

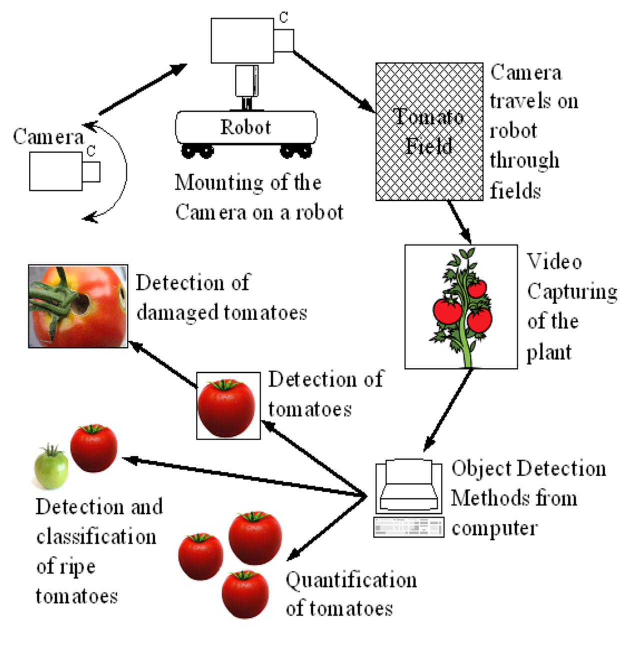


Fig. 1. The research framework for identification and quantification

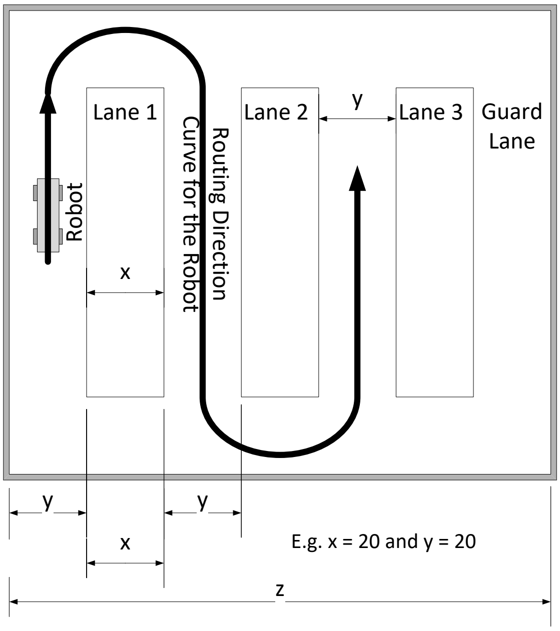


Fig. 2. Structure of the land and routing of robot

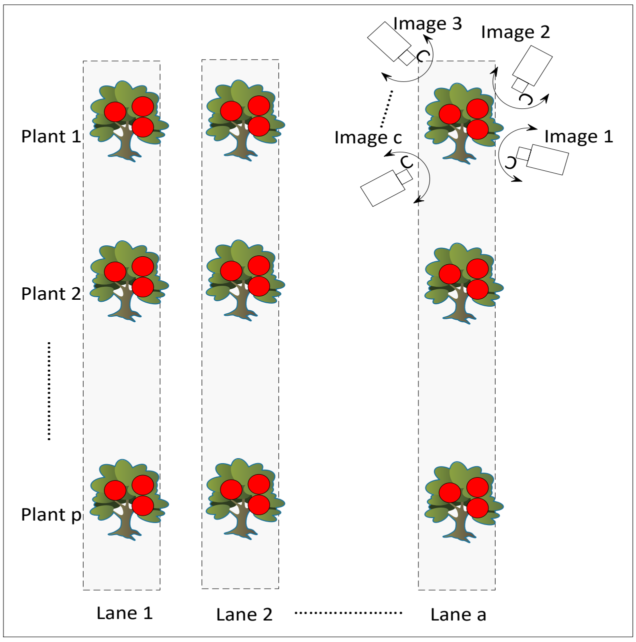


Fig. 3. Distribution of tomato plants in each lane and capturing of images from plants

A. The problem and its justification

The essential problem considered in this paper is to help farmers or growers in processes following cultivation. Traditionally farmers look for ripened tomatoes to sell, the extent to which tomatoes are ripened depending on colour in most cases. If tomatoes can be classified as ripened, semi-ripened or green, with this information being generated automatically, it will assist farmers in saving considerable manual time. This classification process can be applied either before harvesting (in the field) or after harvesting (in storage) or both. Additionally, automatic quantification of tomatoes, classified in the field, will help to determine economic values more efficiently than traditional methods. Additionally, this research helps detect damaged or blemished tomatoes using techniques that can also be applied in both field and storage.

An agricultural field in Chittagong[1] in Bangladesh was selected as the sample crop field for this research. Out of various crops, those which have the most clearly-measurable generic geometrical shapes (i.e. circles), such as tomatoes, were chosen as the primary object for detection. Images of tomatoes were captured by a single camera as shown in the research model in Fig. 1.

A robot is used for the mobility of the camera in routing to different locations over the land through guard lanes as depicted in Fig. 2 and Fig. 3. Each lane is taken as 20 inches (approx. 51cm) and the gap between lanes (guard lanes) are similarly 20 inches.

The circular Hough transformation was used for detection and quantification of tomatoes.

Using thresholding and segmentation, tomatoes with defects have been identified.

Detection and quantification of green, ripened and tomatoes with defects will clearly reduce farmers' workloads and make an important contribution to maintaining market food values even during packing of tomatoes.

This research focuses on the following objectives:

i. Detection and quantification of tomatoes using both the colour transformation and machine learning methods

ii. Checking the ripeness of tomatoes and classifying ripened, semi-ripened and green tomatoes using colour thresholding and cascaded classifier methods (using HOG features)

iii. Detection of tomatoes with defects (using the Colour Segmentation Method)

iv. With a mathematical model for asymptotic analysis of state-of-art algorithms, providing a comparative analysis of two methods used in this paper including their results

II. RELATED WORK

Texture mapping was one of the first object detection methods researched and developed (c1981) [14], followed by detection of objects, such as vehicles, and tracking in surveillance systems. Object detection in real time environments was first introduced through the Viola Jones Algorithm [15] with Haar cascaded classifiers as the basic

[1] Courtesy of Bangladesh Agricultural Research Institute, Chittagong, Bangladesh and acknowledgements to both Ms. Kashfia Ashrafi and Ms. Sheena Laila Sheraj of Department of Computer Science and Engineering, University of Science and Technology Chittagong

methodology. Facial recognition problems were effectively solved and subsequently these Haar features had a focus in the training of sequential classifier approaches [15]. Haar-SURF features were then added with the AdaBoost classifier for object detection, giving better performance [10], overcoming weaknesses in the processing of foreground-based Haar features. Cascaded classifiers with Haar features were added to improve this feature [11] [8]. The traditional AdaBoost method was again enhanced with the added feature of multi-scan detection techniques with soft cascaded classifiers [16]. VeDAS for vehicle detection was added later to improve Haar features for object detection. However, this was only capable of detection of objects from partial rear views [16]. The success rate and accuracy was limited to 87%.

However, accuracy remained a significant challenge in the detection of objects and filters on image processing were introduced. The Kalman filter was initiated in CCTV videos [9] and its accuracy reached almost 96% although, weather conditions could reduce its accuracy to 94%. Additional hardware such as a MMW radar and/or monocular camera showed an accuracy of 92.36% [17] but could not improve further. A Gabor filter was applied for optimal results in vehicle detection [12] and significant results were obtained in the challenges of vehicle heterogeneity [12]. Ultimately, this filter was constrained by frequency responses.

For food and crop detection, several research projects are noteworthy. A.R. Jim´enez et al. [18] conducted a survey as part of a review of locating fruits on trees using computer vision tools. In this, the applicability of the CCD sensor is depicted as 85% for detection of fruit in the tree. In [19] fruit was classified using a fitness-scaled chaotic artificial bee colony (FSCABC) algorithm and feedforward neural network (FNN). This established FSCABC–FNN as having more than 87% accuracy. A review [20] was published in 2014 focusing on the latest developments and applications of computer vision in terms of external quality factors for fruit and vegetables. Colour transformation and machine learning appeared subsequently to enrich detection and computation of these processes for objects (foods and fruits).

The Hough Transformation (HT) is considered best for detection of objects characterised by parametric curves in single images [2] [13] [4] with the Circular Hough Transformation (CHT) a modified variant of HT. CHT achieved wide application in detection of objects with shapes similar to circles [4] [6] [1]. Despite its processing overheads and near exponential run time complexity, CHT performed best in the detection of objects from natural background images taken from real time fields or applications. Although, in some cases, colour transformation and colour segmentation method are needed, accuracy is still optimal compared to others [1] [7]. It remains true that humans can count objects in an image if necessary but, although a challenge with early methods, CHT eventually made it easier to reduce human errors [7]. Subsequently, many applications related to object counting have been noted [21] [3]. The particular challenge of detecting overlapping objects in a single image also proved that CHT is more powerful than other approaches [5]. Therefore, in this paper, CHT is the major research focus for detection and quantification of tomatoes.

Machine learning and deep learning have played a vital role in this area and several contributions by many researchers are notable, particularly in fruit or crop detection. Inkyu Sa et al. [22] present an approach to fruit detection using deep convolutional neural networks. Using imagery from RGB colour and Near InfraRed, Faster R-CNN has been developed for the detection of fruits with performance increased from 80% to 83%. Another approach by Horea and Mihai is noted [23], in which a neural network is trained by a high quality data set for detection of fruits. Zheng et al. [24], using the deep learning method, trained a deep learning classifier with a larger data set, giving a strong benchmark to support the deep learning classification and detection of fruits. The accuracy was over 99%. HOG features along with LBP, Gabor LBP, global colour histogram and global shape features are used in [25] for detection of fruits. This approach offers lower false rates. In [26] HOG features are used to detect and count mangoes in trees. Similarly, Wang [27] has developed an approach to estimate the size of on-tree mangoes. Along with an RGB camera and Laser rangefinder, this uses HOG features of cascaded classifiers and the Otsu method, followed by colour thresholding. Clearly machine learning is gaining popularity in detection of fruits.

Consequently, this research also focuses on the machine learning method using cascaded classifiers with HOG features applied.

A comparative table of the characteristics of related research have been added in Table 1. A hierarchy of different object detection and quantification methods have been highlighted at a glance. Detection was initiated using texture mapping which was later on developed using classifiers and features followed by further improvements in accuracy. Using of filters and different improvements was next of the hierarchy whereas computer vision, neural networks, machine learning and other related methods at the next of that specify how the methodological approach differed with certain level of accuracy.

Table 1 . A comparative table of the characteristics of the related work

Sl. No.	**Name of authors and years**	**Comparative characteristics of related work**
1	F. Crow, 1984	Object detection methods using Texture mapping
2	P. Viola and M. Jones, 2011	Object detection using cascaded classifiers
3	S. Shujuan, X. Zhize, W. Xingang, H. Guan, W. Wenqi and X. De, 2015	Haar-SURF features were added with the AdaBoost classifier for object detection
4	A. Broggi, E. Cardarelli, S. Cattani, P. Medici and M. Sabbatelli, 2014 and X. Zhuang, W. Kang and Q. Wu, 2016	Cascaded classifiers with Haar features were used to improve the weakness

5	R. K. Satzoda and M. M. Trivedi, 2016	AdaBoost method was enhanced adding VeDAS to improve Haar features
6	Z. Chen, T. Ellis and S. A. Velastin, 2012	Kalman filter was introduced
7	X. Wang, L. Xu, H. Sun, J. Xin and N. Zheng, 2016	Additional hardware interfaced to improve performances
8	H. David and A. T.A, 2014	Using of Gabor Filter
9	R. Jiménez, C. A. and J. R. L. Pons, 2000	Survey on Fruit and crop detection using computer vision
10	Y. Zhang, S. Wang, G. Ji and P. Phillips, 2014	Fruit classification using Artificial intelligence and Neural Network
11	Z. BaoHua, H. WenQian, L. JiangBo, Z. ChunJiang, F. ShuXiang, W. JiTao and L. ChengLiang, 2014	Latest developments and applications of computer vision in terms of external quality factors for fruit and vegetables using Colour transformation and Machine learning
12	T. Atherton and D. Kerbyson, 1999 and E. Davies, 2005	Hough transformation for object detection
13	H. Yuen, Princen, J. Illingworth and J. Kittler, 1990 and J. G. A. Barbedo, 2012 and P. Mukhopadhyay and B. B. Chaudhuri, 2015	Circular Hough transformation for object detection
14	T. Liebig, 2015	Colour Transformation and Colour Segmentation
15	T. D'Orazio, C. Guaragnella, M. Leo and A. Distante, 2004 and M. Rizon, H. Yazid and P. Saad, 2007	Object Counting methods
16	J. Ni, Z. Khan, S. Wang, K. Wang and S. K. Haider, 2016	Circular Hough Transformation for detection of single object from overlapped objects
17	I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez and C. McCool, 2016	fruit detection using deep convolutional neural networks
18	Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, T.-L. Su and M. Zuo, 2019	using the deep learning method, trained a deep learning classifier with a larger data set
19	H. Kuang, C. Liu, L. L. H. Chan and H. Yan, 2018	HOG features along with LBP, Gabor LBP, global colour histogram and global shape features are used for detection of fruits
20	N. M. Ali, M. S. Karis, N. Maisarah, M. Sobran, M. B. Bahar, O. K. Ken and M. M. Ibrahim, 2017	HOG features are used to detect and count mangoes in trees
21	Z. Wang, K. B. Walsh and B. Verma, 2017	An approach to estimate the size of on-tree mangoes using an RGB camera and Laser rangefinder. This uses HOD features of cascaded classifiers and the Otsu method, followed by colour thresholding

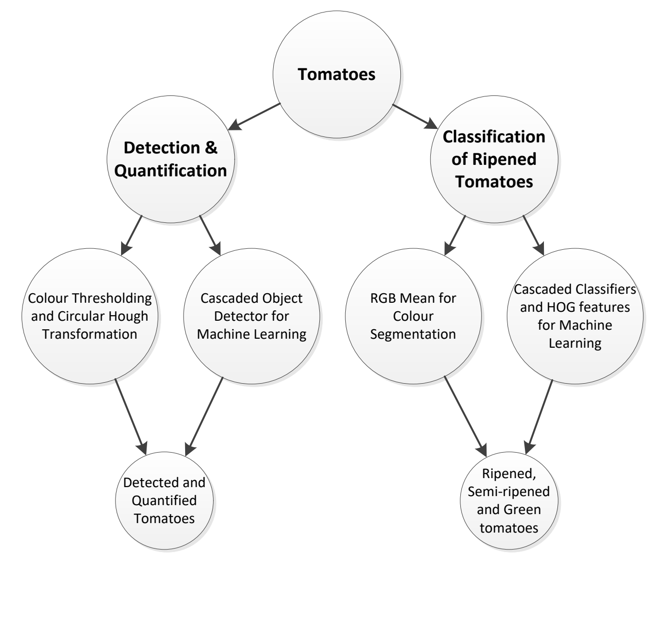
III. Materials and Methodology

A. Materials used

The essential kit in this work is a robot[2] with a camera mounted at its head (Fig. 33 in the appendix). Tomato plants were growing conventionally in the field.

A smaller data set is used for the Convolutional Neural Network (CNN) (Machine learning method) with HOG features. The whole data set is divided into three parts. The first is the training set containing 41 % (of the whole set, which is 247), the second is the test data set which is 75 containing 30% and the remaining 29% is for validations (cross validation) which is 72. From the related research, it noted that the Circular Hough Transformation is one of the best algorithms from all traditional methods and cascaded object detection using HOG features is one of the optimal algorithms for detection of overall shape of tomatoes. Both methodologies are at a glance highlighted in Figure 4.

B. Measurement of lanes and land guard lanes



Assume that the number of lanes and guard lanes are a and b respectively for a total length X for all lanes and a total length Y for all guard lanes (Fig. 2 and Fig. 3).

If the width of each lane, guard lane and the whole land are x, y and z respectively, then the relationships among them are:

$$ax + by = z \text{ ....} \quad (1)$$

$$b = a + 1 \text{ ....} \quad (2)$$

Here, x, y and z are constants and, from these relationships, the values of a and b can be calculated.

---

[2] Specification: A4Tech model PK-760E
Still Image Capture Resolution: Up to 5 Megapixel, 2560x2048 (Software Enhanced), Image Sensor: 1/6"CMOS, 640×480 pixels, Frame Rate: 30 fps at VGA Mode, Lens: F=2.4, f=3.0 mm, View Angle: 66°, Focus Range: Automatic Focus, 10 cm to infinity, Exposure Control: Automatic, White Balance: Automatic, Microphone: Built-in, Computer interface: USB 2.0, System Requirements: Windows XP / Vista / 7 / 8 / 8.1 / 10, https://www.a4tech.com/product.aspx?id=147

$X = ax$................................ (3)

$Y = by$ ....(4)

So,

$Y = (a+1) * y = ay + y$ .. (5)

Also assume that the quantity of plants in each lane is p, a common value for each lane from the value of a. Then there are a total of P plants with

$P = pa$ .... (6)

Assume furthermore that a total of C images have been captured for detection of a total of D tomatoes in this crop field (Fig. 7).

$C = \sum_{i=1}^{P} c_i$ .... (7)

Here, c^i is a series, with $c^i > 0$

Each c^i is an ad hoc number of captured images for a single plant.

If, c^i is equal for all plants, then, c will be a constant for each plant (Fig. 3). Therefore,

$C = cP = cpa$ .... (8)

For a total D of tomatoes from all images in the crop field and a total of C images,

$D = C * \sum_{j=1}^{k} d_j = \sum_{i=1}^{P} C_i * \sum_{j=1}^{k} d_j$ ... (9)

Here, d is the number of tomatoes detected from each image and $k = c^i$ which implies the number of images captured for each plant. If c is constant for all plants then,

$D = C * \sum_{j=1}^{k} d_j = cpa * \sum_{j=1}^{k} d_j$ ... .(10)

C. Detection and quantification of tomatoes

For detection of tomatoes, their quantification, checking of ripeness and defect identification, the following two methodologies have been applied.

- Colour thresholding
- Cascaded object Detector

i. Using the Colour Thresholding Method

a. Steps of Detection with Solid Background:

1. Taking input of images
2. Applying Circular Hough transformation for detection and counting

b. Steps of detection with Natural Background

1. Taking inputs of images
2. Colour adjustment
3. Adjustment of internal threshold
4. Adjustment of sensitivity
5. Adjustment of object polarity
6. Applying of the Circular Hough Transformation for detection and counting

For the Circular Hough Transformation (CHT), we assume that those circles to be detected are lying in the function:

$(x-k)^2 + (y-p)^2 = s^2$ ....(11)

In polar coordinates, these functions are:

$k = x - s\cos\Theta$ ....(12)

$p = x - s\sin\Theta$ ....(13)

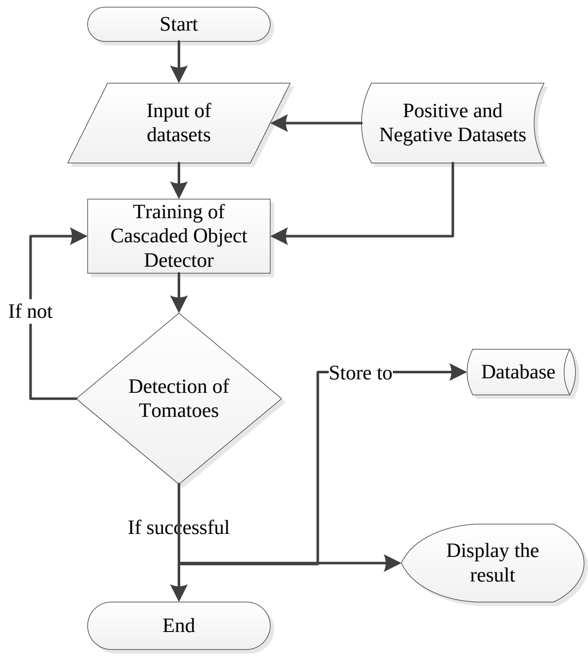


Fig. 5. Flow chart for the cascaded object detector method

Where, (k, p) is the centre of the circle and s is the radius

Assume an input image ‘IMG’, a minimum radius, MIN_R and a maximum, MAX_R. The internal threshold is I_T and sensitivity = 0.85. Object polarity, OBJ_P = ‘dark’ as objects are darker than the background. The edge threshold is E_T. Centres and radii of detected circles are c(k,p) and R respectively. For counting/incrementing, the COUNT variable is used and it denotes labels of images. Initially, COUNT = 0

To increase detection sensitivity, assume that all circles < I_T, sensitivity is set to 0.94

Pseudocode:

```
Sub Function
    INPUT IMG
    SET MIN_R = MIN(R)
    SET MAX_R = MAX(R)
    Start Loop
        SET sensitivity = 0.94
        Call function IFINDCIRCLES
        Lower value of E_T
        Return values of c(k, p) and R
        IF IFINDCIRCLES is TRUE
            Insert circles from c(k, p) and R for detected
    tomatoes in yellow colour,
            ADD boundary circles
            SET COUNT = COUNT + 1
            GOTO LABEL
        End IF
    End Loop
    PRINT COUNT
    LABEL:
    Step 1: Call function METHOD_OF_CIRCLE
END Sub
```

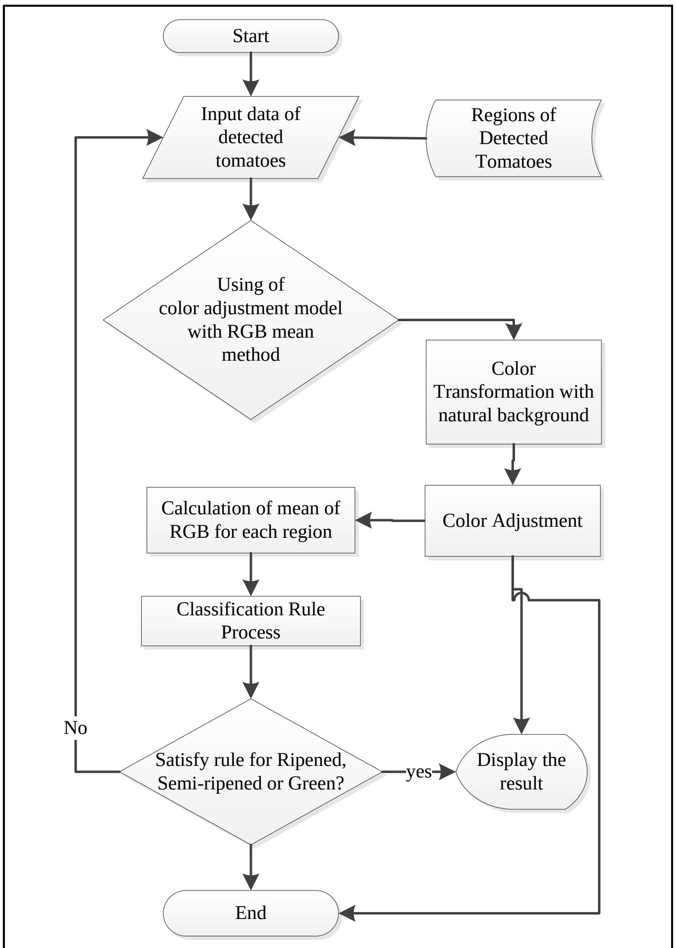


Fig. 6. Flow chart for the detection and classification of ripened tomatoes using traditional approach

Here are two conditions CON1 and CON2.

```
METHOD_OF_CIRCLE
Step 1: If CON1 then TWO_STAGE
        Else if CON2 then PHASE_CODING
END Function
```

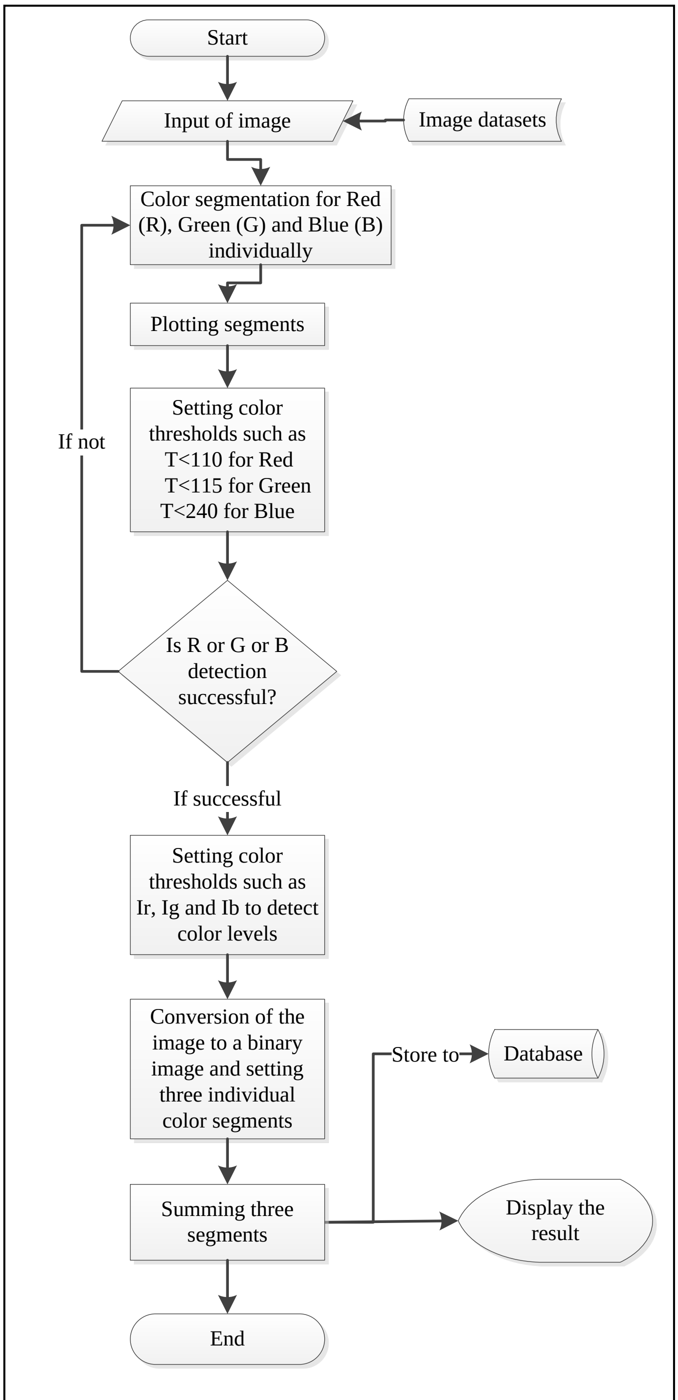


Fig. 7. Flow chart for detection of damaged tomatoes

Here CON1 denotes the condition for lower sensitivity (e.g. sensitivity < 0.92) and used for using of two-stage method whereas CON2 denotes the condition of higher sensitivity (e.g.

sensitivity >= 0.92 or 0.95) which is used for using of phase-coding method as it is faster and robust to noise[3].

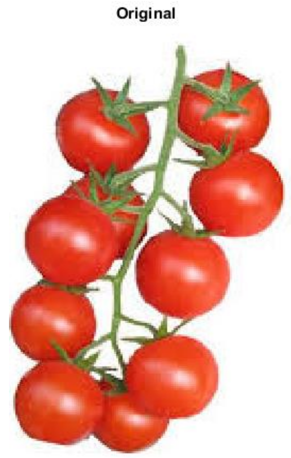


Fig. 8. Input image for detection and quantification of tomatoes

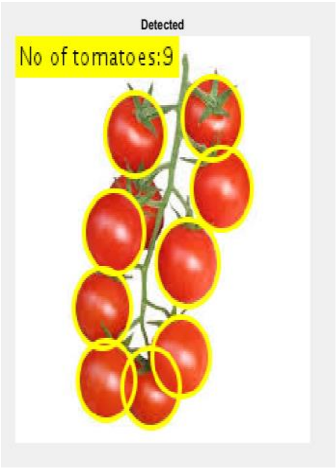


Fig. 9. Detected and quantified tomatoes

Fig. 10. Input image for detection and quantification of tomatoes in different colors with solid background

Fig. 11. Detected and quantified tomatoes in different colors

Fig. 12. Input image for detection of tomatoes with natural background (under shadow)

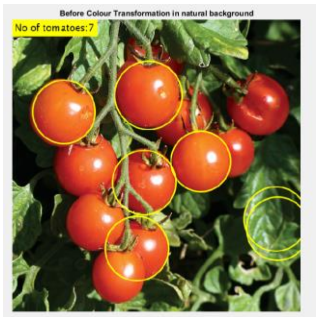


Fig. 13. Detected and counted with false positives in natural background (under shadow)

The edge threshold value is recursively lowered as the higher value (closer to 1) will compute stronger edges and lower values (closer to 0) will compute weaker edges. As the circles (tomatoes) are mostly darker compared to the background, most edge pixels have to be detected through computation.

If the total operation reading for a single image is f(t) then the number of operations can be computed thus:

Step 1 requires 1 operation, step 2 requires 2 operations, step 3 requires 1 operation, step 4 requires n operations, step 5 requires 1 operation, step 6 requires n operations, step 7 requires n for each of n circles and n for the COUNT variable, therefore n*n + n operations, step 8 requires 1 operation.

Therefore,

$$f(t) = 1 + 2 + 1 + n + 1 + n + (n * n + n) + 1 = O(n^2) \ldots \ldots\ldots\ldots (14)$$

Each image process has quadratic run time complexity.

Therefore, for $C = \sum_{i=1}^{P} C_i$ images, the run-time complexity is exponential whereas for

$$C = cP = cpa \text{ (from equation 8)}$$

There will be consistently be $O(n^2)$ (quadratic) run time complexity for each of C as $C = cpa$ is a linear function with $O(n)$. Therefore, the total process will have $O(n)$* $O(n^2)$ or $O(n^3)$ run time complexity (cubic time).

---

[3] Mathworks - https://www.mathworks.com/help/images/detect-and-measure-circular-objects-in-an-image.html

Fig. 14. Accuracy of detection increased after colour transformation

Fig. 15. Green tomatoes on a tomato plant in daylight with a natural background (under sunny lighting)

ii. Using the Cascaded Object Detector Method

Steps of this method:

(1) Extracting the HOG features of the training samples

(2) Training classifier using the extracted features and corresponding labels

(3) Extracting the Region-of-Interest (ROI) on the test image

(4) Detection of Tomatoes

Classifiers of the Cascaded Object Detector contain stages in which a group of weak learners is trained using boosting to allow for training computing weighted average of decisions from weak learners (Fig. 5).

This detector needs both positive and negative datasets. Images of positive datasets contain tomatoes in it that will be detected after training, whereas images of negative datasets do not. The detector is trained for HOG features using those tomatoes that are detected from positive datasets.

Total Data: 247 images
Training: 100 images
Validation: 72 images
Test Dataset: 75 images
Samples: 207
Background: 621
Image pixels: 64 x 64

The above data relate to sample patches of images used for training and 64 x 64 is the patch size. Background is either foliage or 'solid' and there are 621 images with such backgrounds.

Out of all, some of them were taken under sunny light and some of them from shadows. Detection sets were comprised of separated, overlapped and occlusions. Here, at the training phase, training datasets are loaded and at the detection phase, detection are done from test datasets.

The algorithm is given below –

Step 1: Load positive dataset

Step 2: Load negative dataset

Step 3: Train the cascaded object detector for HOG features

Step 4: Read an image from either positive or negative datasets

Step 5: Detect tomatoes from images of Test datasets

Results are depicted in Fig. 26 and Fig. 27.

Assume there are p positive datasets, n negative datasets and a database of bootstrapping negative datasets d. If there are g learning goals then there will be a cascade of the following series of binary nodes of classifiers.

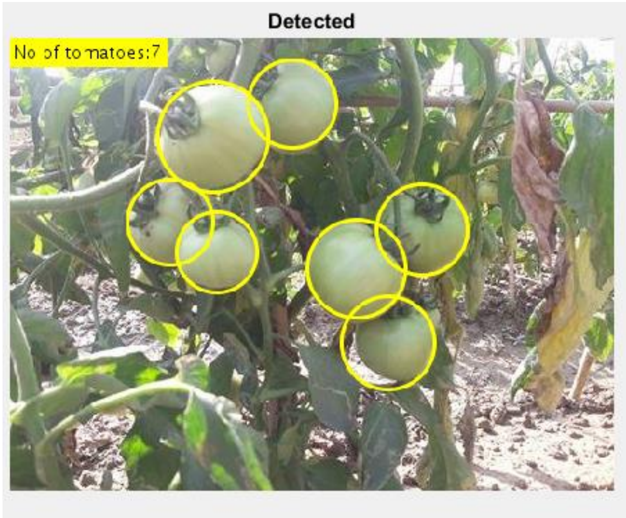


Fig. 16. Detected and quantified tomatoes on a plant in daylight with a natural background (Under sunny lighting)

H1, H2, H3, ……………, Hn … assuming that there are n stages.

Here, n >> p

At each learning stage H, learning will be,

number of positive samples = floor (p/(1+(n – 1)*(1 – TPR)))

TPR = True Positive Rate

FPR = False Positive Rate

FNR = False Negative Rate

At each stage, the number of positive samples will be reduced. At some point, the positive samples will run out. Hence, this is an O(n) problem. There is a trade-off between fewer stages with a lower FPR at each stage and more stages with a higher FPR at each stage. Since FPR reduces exponentially at each stage, a larger number of simple stages are preferred. If the number of stages increases, then the number of training data sets also needs to increase. As a result, the probability of FNR also increases, which results in mistakenly rejecting positive samples.

Consequently, HOG features have been adopted here as the research has a limited number of data sets. HOG features perform better on overall shapes of objects (tomatoes) with smaller data sets [31].

Under certain conditions, the Cascaded Object Detector Method using HOG features is better, when compared to the traditional colour transformation method, as this machine learning method demonstrably performs better for run time processes.

D. Detection and Classification of Ripened Tomatoes

i. Using colour thresholding method

The ripened and green tomatoes are detected using the colour thresholding method (Fig. 23.a, Fig. 23.b and Fig. 24) although the red and green tomatoes could not both have been identified in a single image. The two results of detection of ripened and green tomatoes have been added and the result found in Fig. 25.b, Fig. 25.c and Fig. 25.d using the colour thresholding method where the input image for this method has been used is shown in Fig. 25.a.

Steps (according to Figure 6):

1. take input of images
2. Regions of Interest (ROI) detected
3. colour transformation is used
4. colour adjustment followed by Hough transformation are done
5. Find the mean of ***R, G, B*** values for each region
6. Classify tomatoes according to following using the rules of intensity value considering Fig. 28
   a) Ripened
   b) Semi-ripened
   c) Green
7. Ripen tomatoes are detected using Hough Transformation and displayed

For finding the mean of RGB following equations have been used -

$$R_{avg} = \sum_{i=x_{min},y_{min}}^{x_{min}+w,y_{min}+h} (color\ values, from\ R)$$

…….. (16)

$$G_{avg} = \sum_{i=x_{min},y_{min}}^{x_{min}+w,y_{min}+h} (color\ values, from\ G)$$

…….. (17)

$$B_{avg} = \sum_{i=x_{min},y_{min}}^{x_{min}+w,y_{min}+h} (color\ values, from\ B)$$

…….. (18)

Above equations 16, 17 and 18 are used to compute mean values to build rules defined in the decision table of Table 2.

Tomatoes are classified as Ripened, Semi-ripened or green using rules established in Table 2, according to Fig. 29 and Fig. 30. Using conventional colour analysis with Red, Green and Blue, rules for Ripened, Semi-ripened and Green have been set [32].

ii. Using Machine Learning method

Total Data: 247 images
Training: 117 images
Ripped – 45
Green – 46
Semi ripped – 26
Validation: 72 images
Test Dataset: 75 images

The algorithm is given below –
Step 1: Load positive dataset
Step 2: Load negative dataset
Step 3: Train the cascaded object detector for HOG features
Step 4: Read an image from either positive or negative datasets
Step 5: Detect tomatoes from images of test datasets
Step 6: Check Ripe tomatoes

E. Algorithm and process for detection of damaged tomatoes

The image IMG is processed as an input file. A variable for Thresholding T is set. Another variable SE declared. Ir, Ig and Ib are threshold variables. B_IMG is further image variable for storing a binary image. Three segment variables are set as I1, I2 and I3. ISUM is a variable for summing of segment variables. According to Figure 7 following steps have been applied:

1. taking input image from the image datasets

2. Colour segmentation for R, G and B
3. Plotting segments for R, G and B
4. Setting colour thresholds for detection of R, G and B
5. If detection is TRUE or successful
    i. then setting these thresholds to detect colour levels
    ii. conversion of image to a binary image
    iii. summering three segments and store to database
    iv. displaying the result
Else
    Return to step 2 for colour segmentation

The pseudocode is given below -

```
Step 1: Read image IMG as Input from Image datasets
Repeat Step 2 for I = 1 to 3
Step 2: Call Function Col_Seg (R, G, B)
    if I = 1 then Red colour segment
        elseif I = 2 then Green colour segment
        elseif I = 3 then Blue colour segment
    else BREAK
Step 3: plot segments Red, Green and Blue
Step 4: Set Threshold to detect colour according to followings
T<110 for Red
T<115 for Green
T<240 for Blue
Step 5: Call function IMFILL(T, S as STRING), STREL(D as STRING, A as INT) and IMDILATE(T, SE)
    T = IMFILL(T, S as STRING)
    SE = STREL(D as STRING, A as INT)
    T = IMDILATE(T, SE)
Step 6: IMG is processed and call Function
    SUBPLOT(1, 2, 1)
    IMSHOW(T)
Step 7: Set Threshold colour levels as given below
Ir = 0.7, Ig = 0.7, Ib = 0.8
Step 8: Convert IMG to a binary image using threshold values
    Set level = in range of [0, 1]
B_IMG = IM2BW(IMG, level)
I1 = IM2BW(Red, Ir)
I2 = IM2BW(Green, Ig)
I3 = IM2BW(Blue, Ib)
Step 9: Sum three segment values with ISUM and displaying the binary image
    ISUM = I1 & I2 & I3
    Call function SUBPLOT(1,2,2)
    Call function IMSHOW(ISUM)
END
```

Using the function IM2BW(IMG, level), the B_IMG file replaces all pixels in the input image IMG with luminance greater than level value 1 (white) and replaces all other pixels with luminance level value 0 (black).

The total operations processed in this algorithmic process are then as follows:

Step 1 has 1 operation, step 2 has 3 operations, step 4 has 3 operations, step 5 has n+n+n = 3n operations, step 6 has 2 operations, step 7 has 3 operations, step 8 has, for both level >=0 and 1, n*n operations and 3 operations and step 9 has 3 operations.

Therefore, the total process has a total of

$1 + 3 + 3 + 3n + 2 + 3 + (n^2 + 3) + 3$ operations, which represents $O(n^2)$ or quadratic run time complexity for processing of a single image.

Thus, for $C = \sum_{i=1}^{P} C_i$ images, there will be exponential run time complexity. However, for $C = cP = cpa$, $O(n^2)$ or quadratic run time complexity for each of C and as $C = cpa$ will result instead: a linear function with $O(n)$. Thus, the total process will be of $O(n)$* $O(n^2)$ or $O(n^3)$ run time complexity (cubic time).

IV. Result and Discussion:

The system is using a simple camera robot (figure 37 in Appendix section) and images have been collected (figure 39 in Appendix section). Input images, such as figure 38, figure 39, figure 40 and figure 41 have been taken as input to methodologies described in the previous section. The result is shown in terms of performances of detection, identification of damaged tomatoes and identification of ripened tomatoes.

A. Performance in detection of tomatoes

Images with solid backgrounds (Fig. 8 and Fig. 10) have better detection and quantification results (Fig. 9 and Fig. 11) than those (Fig. 13, Fig. 14 and Fig. 16) with natural backgrounds (Fig.12 and Fig. 15). Since the background has a similar brightness across much of it, circles are more identifiable than when darker than the background.

For images of natural backgrounds, there is an uneven brightness over the image. Hence, some false positives appear in Fig. 13 for the input image of Fig. 12. Therefore, a colour transformation was essential to increase the accuracy of detection (Fig. 14).

This additional transformation will add an extra overhead to the total process.

Different parameters in Table 3 provide sensitivity rates for both the Two-stage and phase-code processes with a common object polarity and threshold. The object polarity has been set to dark. Detection performance has also been compared for solid and natural backgrounds in Fig. 31.b and Fig. 31.a respectively. The number of false positives is less although tomatoes against a natural background have still been considered: this is the primary challenge for the real time problem in this research as has been already identified in Fig. 15 and Fig. 16. The accuracy of the Colour Segmentation Method is approximately 84% (under sunny light and less in shadows) whereas that of machine learning is approximately 92%.

B. Identification of damaged Tomatoes

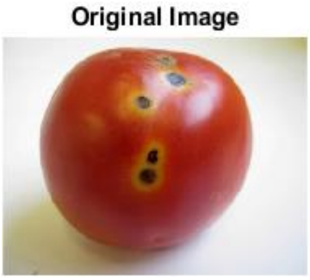


Fig. 17. Input image for detection of damaged tomato

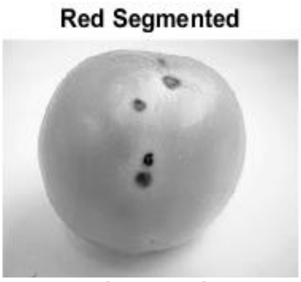


Fig. 18. Red segmented

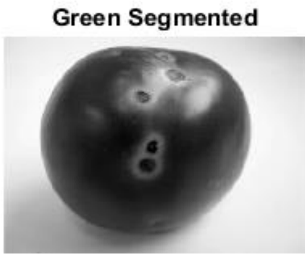


Fig. 19. Green segmented

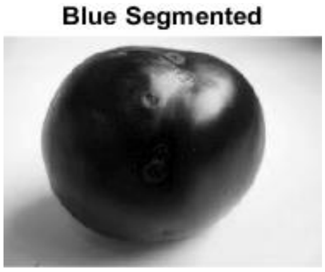


Fig. 20. Blue segmented

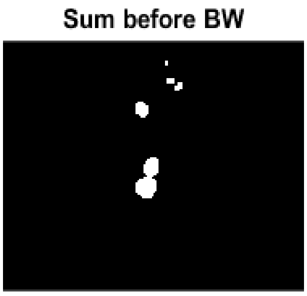


Fig. 21. Summed before

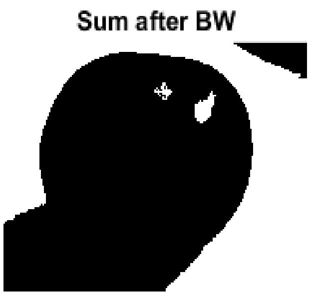


Fig. 22. Summed after

The result of the blemished tomato is shown in Fig. 21 and Fig. 22 with the input image in Fig. 17 processed in Fig. 18, Fig. 19 and Fig. 20.

Colour Segmentation has been performed using values of Red (Fig. 18), Green (Fig. 19) and Blue (Fig. 20).

The values obtained from segmentation are shown before binary conversion in Fig. 21 and after in Fig. 22.

C. Identification of Ripened Tomatoes

Fig. 23.a. Input image for detection of Ripened Tomatoes after RGB adjustment

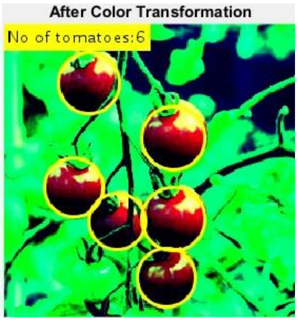


Fig. 23.b. Detection of ripened tomatoes after colour thresholding

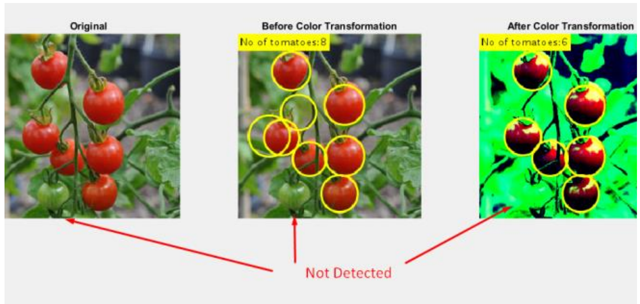


Fig. 24. Undetected green tomatoes in colour thresholding

Fig. 25.a. Input image for Detection and quantification of ripened and green tomatoes

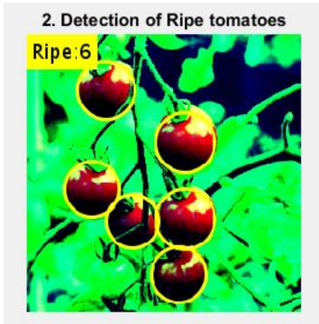


Fig. 25.b. Detected and quantified ripe tomatoes before color transformation

Fig. 25.c. Detected and quantified green tomatoes after Colour Transformation

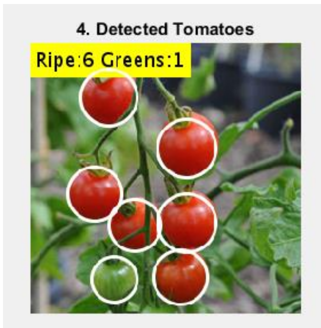


Fig. 25.d. Detected and quantified ripe and green tomatoes after Colour Transformation

Using the colour thresholding method, with a natural background, the process of classification of tomatoes gets nested with an additional adjustment of colours and the addition of two results in a single image so that both ripened and green tomatoes with quantification are identified (in Fig. 23.a, Fig. 23.b and Fig. 24). For the input image of Fig. 25.a. an RGB colour adjustment has been performed and ripe tomatoes have been detected (in Fig. 23.b. and Fig. 25.b). Still, the green one could not be identified (in Fig. 23.b. , Fig. 24. and Fig. 25.b.). Hence, there was an additional colour adjustment for green. The result is now visible in Fig. 25.c. For the quantification of both ripe and green tomatoes, two results (Fig. 25.b. and Fig. 25.c.) are added in a single image (Fig. 25.d.).

Using a cascaded classifier, tomatoes are classified into three categories: ripened, semi-ripened and green (Fig. 30). Input images (figure 26) has been collected from sunny lighting and after detection (figure 27), they have been again processed for checking of ripeness. Figure 28 denotes the calculation of RGB mean stated in the algorithm of detection of tomatoes using colour thresholding methods. Figure 30 represents the decision rule using which the ripeness of tomatoes are calculated from RGB images. A rule has been established for classification based on the calculation of a weighted average of Red, Green and Blue using equation 16. The rule is formulated in Table 4 with its respective colour parameters.

A. *Comparison of methods and analysis*

The Decision table (Table 2) is an accurate reflection of classification of ripened tomatoes based on rules using different colour weighted values. There are three rules stated.

The first rule is for the ripened decision and the second and third for semi-ripened and green decisions respectively.

Fig. 26. Input image for cascaded classifier (in sunny lighting)

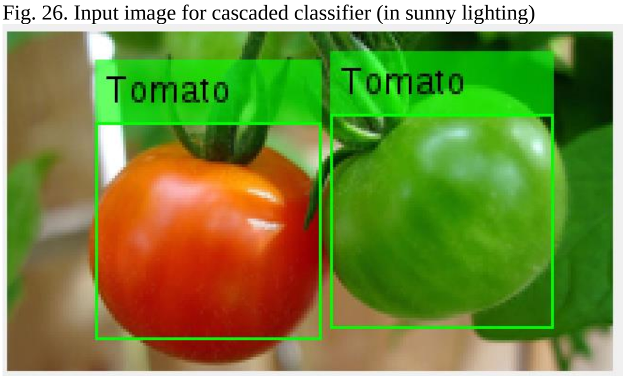


Fig. 27. Detected tomatoes by Cascaded Classifier (in sunny lighting)

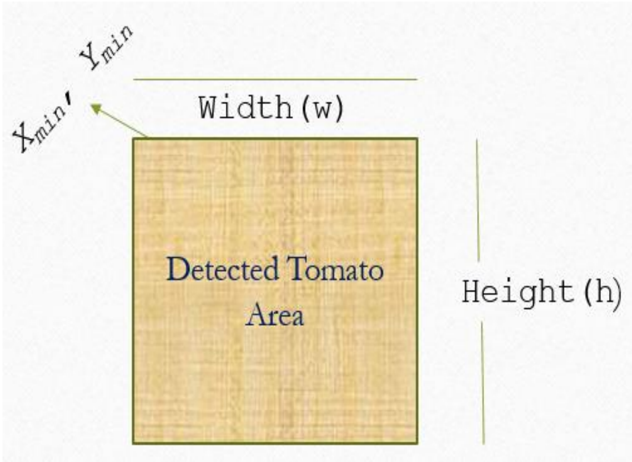


Fig. 28. Calculation of RGB Mean

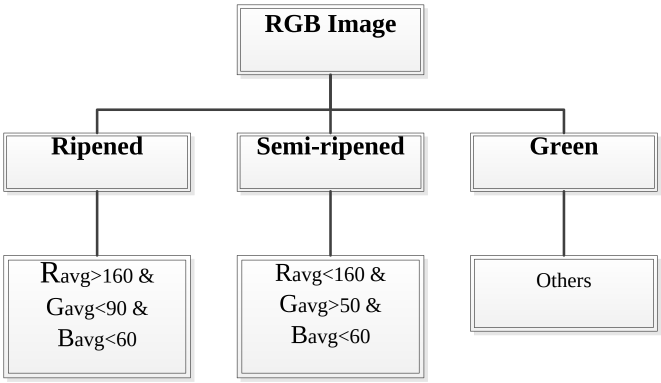


Fig. 29. Decision of Ripeness based on RGB Mean

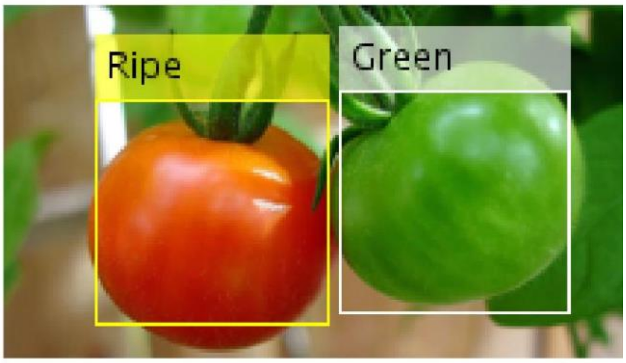


Fig. 30. Classification on Ripeness based on RGB Mean

Table 2: Decision Table

Rule No.	$\mathbf{R_{avg}}$	$\mathbf{G_{avg}}$	$\mathbf{B_{avg}}$	**Decision**
1	>160	<90	<60	Ripened
2	<160	>50	<60	Semi-ripened
3	<50	>50	<50	Green

Table 3. Different parameters for optimal detection

Topic	**Regular**	**Revised**

Circle finding Method	Two-stage	Phase code
Sensitivity	0.90	>=0.92
Object Polarity	Dark	Dark
Edge Threshold	0.11	0.11

Table 4: Comparison between regular and revised parameters

Topic	**Regular parameters**	**Revised parameters**
Total Tomatoes in image	75	75
Detected in shadow lighting	50	59
Detected in Sunny lighting	58	63
False positive	5	5
Undetected	1	0

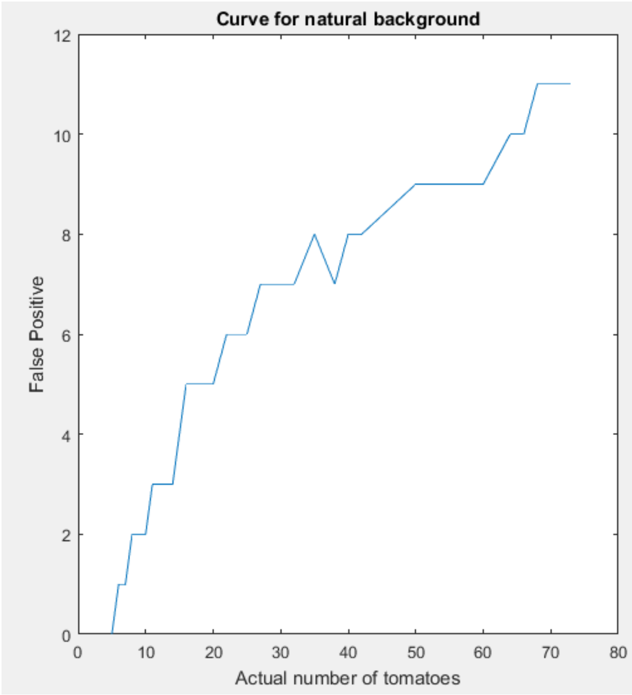


Fig. 31.a. No. of False positive vs. No. of Detected tomatoes in Natural background

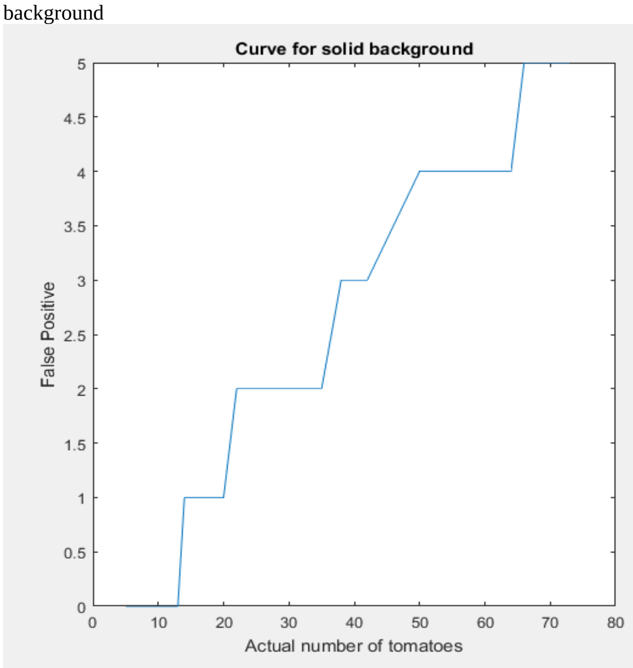


Fig. 31.b. No. of False positive vs. No. of Detected tomatoes in Solid background

From Table 3, for optimal detection, sensitivity is set to 0.92 (compared with 0.90 previously). This increase is due to the detection of tomatoes (circles) being lower in colour transformation methods. Object polarity was set to Dark as the detector can better detect objects brighter than the background.

The circle finding method was revised to Phase code as this was faster and more robust to noise than the Two-stage method.

The result has also been updated from Table 4. False positives have been removed but undetected tomatoes increased to 2.

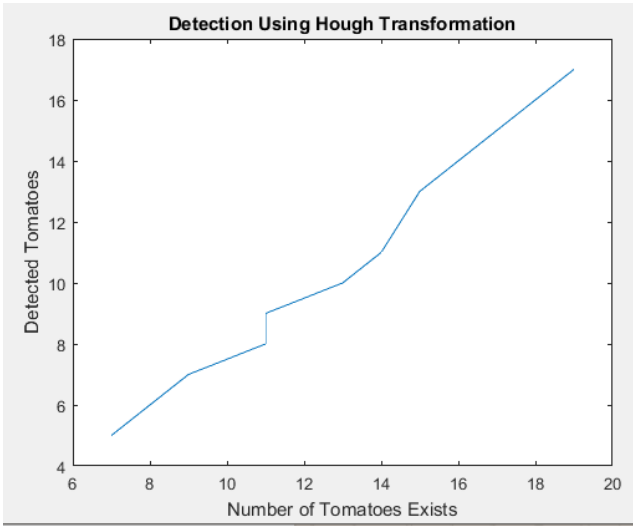


Fig. 32. Detected tomatoes from actual tomatoes using HT method

Table 5. Comparison of Cascade Object Detector (COD) over Colour Segmentation Method (CSM) in ripeness detection

Topic	CSM	COD
Accuracy (in terms of false Detection)	84%	92%
Time Complexity	O(n)	$O(n^2)$

For detection of tomatoes, the machine learning method demonstrated better results in accuracy and run time complexity. However, checking ripeness by COD preserves $O(n^2)$ time complexity, whereas CSM retains O(n) time complexity (Table 5).

In Fig. 32 it has been depicted that the detection ratio is increased in terms of the actual number of tomatoes. In Table 6 and Table 7 the comparison between traditional method and the machine learning method are given in terms of true positives, false positives and false negatives.

Table 6. True positive vs False negative

Lighting	No. of tomatoes exists (Out of 150)	True Positives	%	False negative	%
Sunny (Tr)	75	63	84	12	16
Sunny (ML)	75	69	92	06	08
Shadow (Tr)	75	59	78	16	21.3
Shadow (ML)	75	65	86	10	13

Tr = Traditional method and

ML = Machine Learning Method

Table 7. No. of False positives

Lighting	No. of tomatoes exists (Out of 150)	False positive	%
Sunny (Tr)	75	05	06
Sunny (ML)	75	04	05.3
Shadow (Tr)	75	05	06
Shadow (ML)	75	05	06

The Cascaded Classifier Object Detector has been found to be the better of the two methods in terms of detection. However, it has certain limitations regarding its runtime complexity: $O(n^2)$ compared to O(n) for the Colour segmented method with lower accuracy. False positives and false negatives appeared in various results shown in Fig.10, Fig. 11, Fig. 13 and Fig. 21.

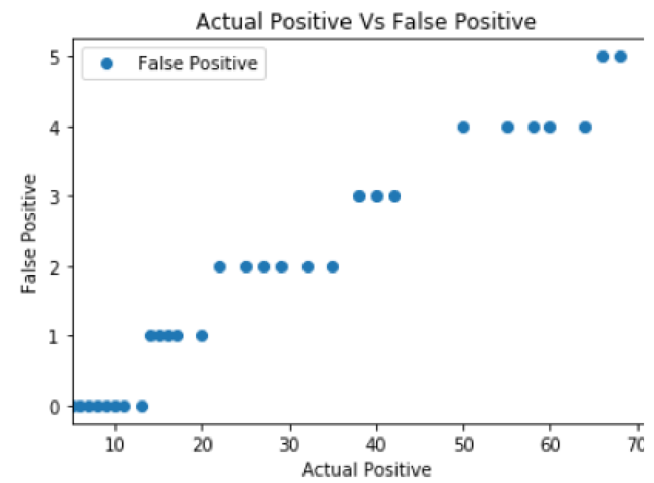


Figure 33 Dataset plot for prediction

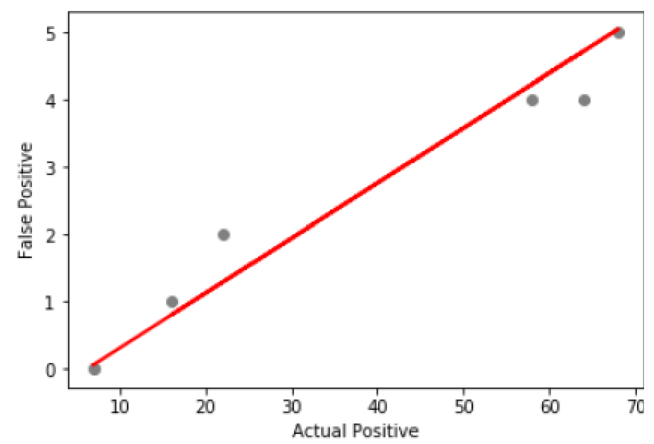


Figure 34 Scatter plot of the regression line for the model of prediction

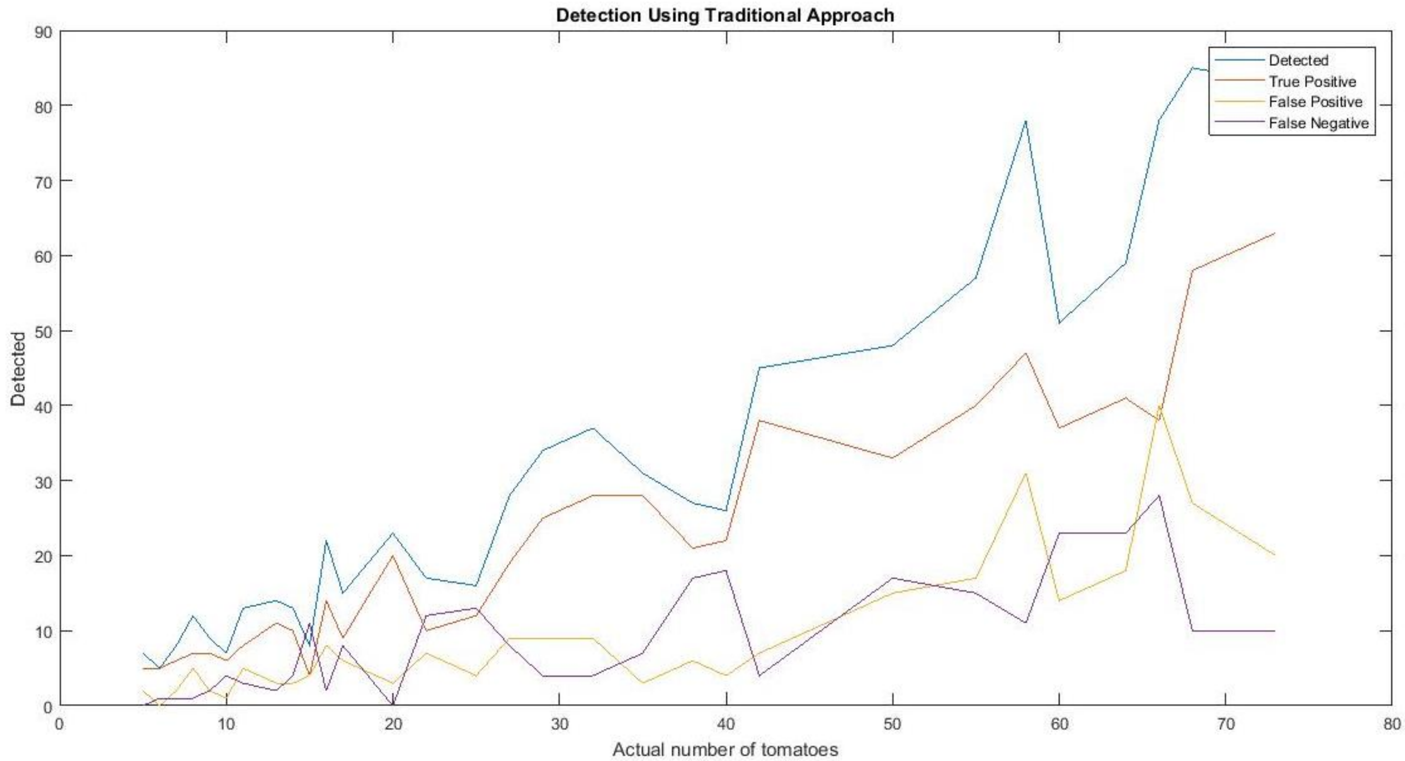


Fig. 35 Performancee of Traditional Method

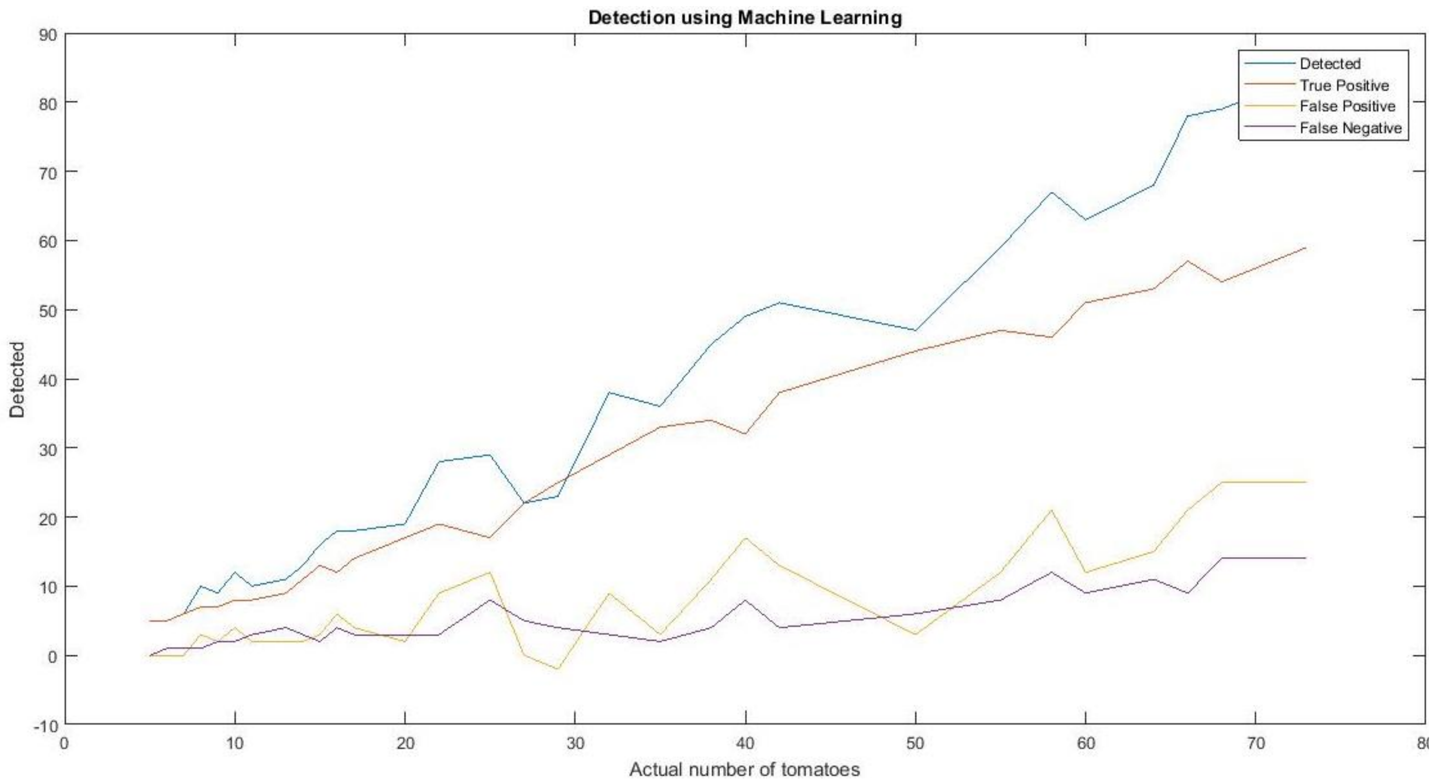


Fig. 36 Performance of Machine Learning

In Figure 35 and Figure 36, performances of Traditional method and Machine Learning methods have been depicted respectively. Clearly, it shows that Machine Learning method out performs the Traditional one.

Using python and Sqllite, two datasets of actual positives and false positives have been used. For exploratory data analysis, these datasets were trained under a prediction model which was later on depicted with a regression line as a fitted curve. The plotting of this dataset is shown in figure 33 whereas the prediction model was depending on the correlation of this dataset (shown in figure 34). According to the scoring of this model, it was an optimally fitting curve whose accuracy in r^2 was 0.9216. Hence, the accuracy of the model was 92% for which no further optimization is required.

V. Conclusion

For detection of tomatoes, the primary objective was to evaluate against a natural background. However, to compare its effectiveness and accuracy, detection against solid backgrounds (the original background removed manually) has also been performed. Performance under a natural background is still lagging behind that of the solid background; but the result is closer. We expect improved accuracy in our future research.

In colour transformation methods, some of its parameters were revised to obtain better results. Sensitivity, circle detection methods and polarity were changed.

For checking ripeness using the machine learning method, we have used the HOG feature. Future research will be performed under Haar features using larger data sets.

Damaged tomatoes have been identified using a traditional Colour Segmentation Method. Not too many tomatoes have been taken as samples (10 only). In this research, tomatoes having some black spots or dots on their outer skin are considered as damaged tomatoes. Here, the machine learning approach could not be applied due to time limitation.

In our research the quantification of tomatoes has been focused on quantification from individual images, which is a limitation of this research. However, in the broader sense, the objective was the cumulative process, which is quantification of tomatoes from the entire field.

For detection, no processes exceeded $O(n^2)$ time. In case of detection of tomatoes, colour transformation used $O(n^2)$ or cubic ($O(n^3)$) time complexity. However, checking ripeness by COD preserves $O(n^2)$ time complexity, whereas CSM retains $O(n)$ time complexity (Table 4).

This research has proposed several algorithms using both image processing and machine learning methods for detection and quantification of tomatoes, which will significantly benefit farmers in terms of food values. Quantified ripened tomatoes will reduce time and provide a better price for farmers for their clients. Identification of tomatoes with defects in the crop field will reduce the time to compute market values, which will ease the pricing, selling and consuming process in the market.

In spite of being an exponentially complex process of Circular Hough Transformation, we ensure the customized process to be limited to quadratic complexity at run time for single image processes and cubic complexity at run time overall processes. Therefore, images are scalable for a constant finite quantity for all plants. However, the accuracy of the colour transformation and Colour Segmentation Method is limited to 84% (Table 5) but with different regular and revised parameters (Table 4).

By comparison, machine learning increased the accuracy of results, highlighted in Table 4.

VI. References

[1] J. G. A. Barbedo, “A review on methods for automatic counting of objects in digital images,” *IEEE Latin America Transactions,* vol. 10, no. 5, p. 2112 – 2124, 2012.

[2] T. Atherton and D. Kerbyson, “Size invariant circle detection,” *Image and Vision Computing,* vol. 17, no. 11, pp. 795-803, 1999.

[3] T. D’Orazio, C. Guaragnella, M. Leo and A. Distante, “A new algorithm for ball recognition using circle hough transform and neural classifier,” *Pattern Recognition,* vol. 37, no. 3, p. 393 – 408, 2004.

[4] P. Mukhopadhyay and B. B. Chaudhuri, “A survey of hough transform,” *Pattern Recognition,* vol. 48, no. 3, p. 993 – 1010, 2015.

[5] J. Ni, Z. Khan, S. Wang, K. Wang and S. K. Haider, “Automatic Detection and Counting of Circular Shaped Overlapped Objects Using Circular Hough Transform and Contour Detection,” in *12th World Congress on Intelligent Control and Automation*, 2016 .

[6] H. Yuen, .. Princen, J. Illingworth and J. Kittler, “Comparative study of Hough transform methods for circle finding,” *Image and Vision Computing,* vol. 8, no. 1, p. 71–77, 1990.

[7] T. Liebig, “Privacy preserving centralized counting of moving objects,” *Lecture Notes in Geo information and Cartography, Lisbon, Portugal,* vol. 217, p. 91 – 103, 2015.

[8] A. Broggi, E. Cardarelli, S. Cattani, P. Medici and M. Sabbatelli, “Vehicle detection for autonomous parking using a Soft-Cascade AdaBoost classifier,” in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014.

[9] Z. Chen, T. Ellis and S. A. Velastin, “Vehicle detection, tracking and classification in urban traffic,” in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2012.

[10] S. Shujuan, X. Zhize, W. Xingang, H. Guan, W. Wenqi and X. De, "Real-time vehicle detection using Haar-SURF mixed features and gentle AdaBoost classifier," in *27th Chinese Control and Decision Conference (CCDC)*, 2015.

[11] X. Zhuang, W. Kang and Q. Wu, "Real-time vehicle detection with foreground-based cascade classifier," *IET Image Processing,* vol. 10, no. 4, 2016.

[12] H. David and A. T.A., "Improving the Performance of Vehicle Detection and Verification by Log Gabor Filter Optimization," in *Fourth International Conference on Advances in Computing and Communications (ICACC)*, 2014.

[13] E. Davies, Machine Vision: Theory, Algorithms, Practicalities. Chapter 10, 3rd ed., Morgan Kauffman Publishers, 2005.

[14] F. Crow, "Summed-area tables for texture mapping," *Proceedings of SIGGRAPH,* vol. 18, no. 3, p. 207–212, 1984.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition,* 2011.

[16] R. K. Satzoda and M. M. Trivedi, "Multi-part Vehicle Detection using Symmetry Derived Analysis and Active Learning," *IEEE Transactions on Intelligent Transportation Systems,* 2016.

[17] X. Wang, L. Xu, H. Sun, J. Xin and N. Zheng, "On-Road Vehicle Detection and Tracking Using MMW Radar and Monovision Fusion," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, no. 7, pp. 2075 - 2084, 2016.

[18] R. Jiménez, C. A. and J. R. L. Pons, "A survey of computer vision methods for locating fruit on trees, https://doi.org/10.13031/2013.3096," *Trans. ASAE,* vol. 43, no. 6, pp. 911-920, 2000.

[19] Y. Zhang, S. Wang, G. Ji and P. Phillips, "Fruit classification using computer vision and feedforward neural network," *Journal of Food Engineering, https://doi.org/10.1016/j.jfoodeng.2014.07.001,* vol. 143, pp. 167 - 177, 2014.

[20] Z. BaoHua, H. WenQian, L. JiangBo, Z. ChunJiang, F. ShuXiang, W. JiTao and L. ChengLiang, "Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review, https://doi.org/10.1016/j.foodres.2014.03.0," *Food Research International,* vol. 62, pp. 326 - 343, 2014.

[21] M. Rizon, H. Yazid and P. Saad, "A comparison of circular object detection using hough transform and chord intersection," *Geometric Modelling and Imaging, GMAI 2007, Zurich, Switzerland,* p. 115 – 120, 2007.

[22] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez and C. McCool, "DeepFruits: A Fruit Detection System Using Deep Neural Networks," *Sensors MDPI; doi: 10.3390/s16081222,* vol. 16(8): 1222, 2016.

[23] H. Mureșan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica; DOI: 10.2478/ausi-2018-0002,* Vols. 10(1):26-42, 2018.

[24] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, T.-L. Su and M. Zuo, "CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture," *Sensor MDPI; doi: 10.3390/s19051058,* vol. 19(5): 1058, 2019.

[25] H. Kuang, C. Liu, L. L. H. Chan and H. Yan, "Multi-class Fruit Detection Based on Image Region Selection and Improved Object Proposals," *Neurocomputing 283; DOI: 10.1016/j.neucom.2017.12.057,* 2018.

[26] N. M. Ali, M. S. Karis, N. Maisarah, M. Sobran, M. B. Bahar, O. K. Ken and M. M. Ibrahim, "Detection of multiple mangoes using histogram of oriented gradient techniquein in aerial monitoring," *ARPN Journal of Engineering and Applied Sciences,* vol. 12, no. 8, 2017.

[27] Z. Wang, K. B. Walsh and B. Verma, "On-Tree Mango Fruit Size Estimation Using RGB-D Images," *Sensors: https://doi.org/10.3390/s17122738*, Vols. 17(12), 2738, 2017.

[28] JianxinWu and J. M. Rehg, "Chapter 8: Object Detection," in *Ensemble Machine Learning: Methods and Applications*, DOI 10.1007/978-1-4419-9326-7 8, © Springer Science+Business Media, LLC 2012, 2012, p. 225.

[29] P. B. Pathare, U. L. Opara and F. A.-J. Al-Said, "Colour Measurement and Analysis in Fresh and Processed Foods: A Review," *Food and Bioprocess Technology; DOI: 10.1007/s11947-012-0867-9,* 2013.

[30] H.-H. Nguyen, P.-T. Tran-Huu, H. M. Tran and S. V.-U. Ha, "Improved Optical Flow Estimation In Traffic Monitoring System," *Information and Communication Technologies (WICT),* pp. 165 - 169, December 2013.

[31] P. K. Bhaskar, S.-P. Yong and L. T. Jung, "Enhanced and Effective Parallel Optical Flow Method for Vehicle Detection and Tracking," in *2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC).*

[32] D. J. Fleet and Y. Weiss, "Optical Flow Estimation," *Handbook of Mathematical Models in Computer Vision. Springer. ISBN 0-387-26371-3,* January, 2006.

VII. Appendix

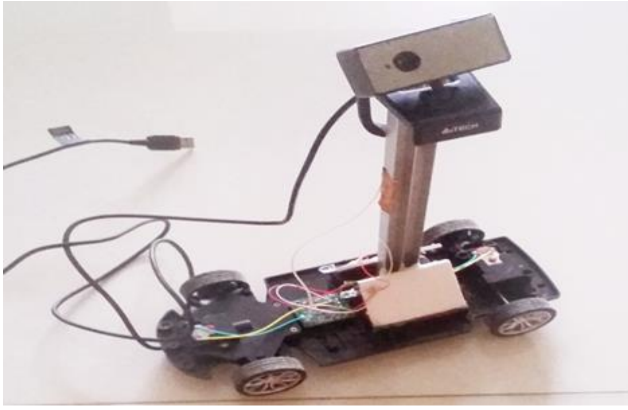


Fig. 37 The mobile robot used for this research

Different views of a single plant:

Fig. 38 Images of tomatoes collected from the field (front side of the plant)

Fig. 39 Mobile robot being used in the field

Fig. 40 Images collected from the tomato field (rear view of the plant)

Fig. 41 Images collected from the field (left hand view of the plant)