Smart Fruit Identification and Counting using Machine Vision Approach

Madhura R. Shankarpure[1], Dipti D. Patil[2]
Department of Computer Engineering, STES's Smt. Kashibai Navale College of Engineering, India[1]
Department of Information Technology, MKSSS's Cummins College of Engineering for Women, India[2]

***Abstract*—Estimating fruit yield holds significant importance for farmers as it enables them to make precise resource management decisions for fruit harvesting. The adoption of automated image processing technology not only reduces the human labor required but also enhances the accuracy of ripe fruit estimates. This research delves into the performance of an image processing algorithm designed to count and identify oranges. The study employed a multi-phase approach, starting with the creation of a mask to isolate orange content, followed by the detection of circular shapes within the mask. Lastly, the algorithm filters and counts the identified circles. The outcome of this study revealed that the algorithm demonstrated an impressive success rate of approximately 72.4% in correctly identifying oranges with standard deviation of +/- 12.20.**

***Keywords*—Image processing; fruit; multiphase approach; counting**

I. Introduction

Crop yield estimation is important to farmers so that they can accurately predict and manage the resources that will be required to harvest the crop, and to tell when to best pick the crop. The technique of estimation is laborious, occasionally assisted by the use of hand counters. Even though only a few trees are typically examined per block, many man hours are still needed in the hot, humid area [1]. Although multiple load assessments throughout crop growth are ideal, labor constraints make this impossible. Additionally, the sample process has a limit on how accurately the process can estimate the yield of the entire orchard [2]. To meet the requirements of the expanding population while effectively utilizing the resources at hand, sustainable agriculture is important [3]. Precision agriculture, which is aided by cutting-edge sensing and image processing systems [4], artificial intelligence, and other technologies, can produce it. Early in the 1980s, Precision agriculture was created [5]. With the use of Deep Learning architectures and contemporary machine vision, Precision agriculture has a revolutionary impact on a number of agricultural applications, including crop monitoring, disease diagnosis, and intelligent yield calculation. Among these, intelligent fruit yield assessment is crucial in determining the ultimate choices for fruit management and harvesting [6]. An automated image processing technique will allow farmers to reduce the amount of manual labor that is required, and increase the accuracy of the estimated count of ripe fruits. Hannan et al. [7] presents a machine learning system for recognizing orange fruit that consists of segmentation, region labelling, size filtering, perimeter extraction, and perimeter-based detection. Fruits could be clearly differentiated from the background thanks to the segmentation technique, which is based on a Bayesian discriminating analysis. Segmentation, region labelling and perimeter-based detection are all parts of the fruit recognition method. In order to provide adaptive segmentation under changing outside lighting for the segmentation of the orange fruit, the red color space factor was used for augmentation in [8]. Laplacian pyramid transform and fuzzy logic are two image fusion techniques that are used to increase the effectiveness of fruit detection in comparison to Leemans et al.'s [9] use of solely thermal images. In the example in [10], color cues in a picture can be effectively used to segment flaws in "Jonagold" apples. It has been discovered that texture characteristics provide helpful information for evaluating the quality of fruits for example, classifying the grade of apples after dehydration with an accuracy of 95% [11]. The dual color/shape analysis algorithm and laser range-finder model were used to create the spherical fruit recognition system in [12]. The models for illumination and surface reflectance for use in outdoor color vision are explored in [13], with a focus on how to forecast the color of surfaces in outdoor settings. The fruit detection algorithm was created by the authors of [14] using a combination of several observing approaches for a tree canopy. For color recognition, texture and color features are used. Fruit identification uses an effective blend of color and texture. By using a minimal distance classifier and statistical and co-occurrence data obtained from wavelet. processed sub-bands, recognition is accomplished [15]. To distinguish apples with red and green color, color and texture traits are used [16]. According to Femandez et al. [17], a tomato-harvesting robot's stereoscopic vision system has a detection accuracy of about 85%. The author in [18] uses a local or shape-based analysis to quickly identify the fruit and was able to identify it at particular stages of maturation. In order to characterize the characteristics, a key point extraction technique [19] combined corner points and edge points, but it was unable to represent surfaces or an object as a whole. Lokhande et.al. [20] utilize a machine learning technique to evaluate the raw data and determine the trust characteristic. To identify the object a fuzzy control method can be used [21].

This research aims to modify and test a method developed by Payne, Walsh, Subedi and Jarvis to identify ripe Mangos, to identify oranges [22]. To do this however, the traits that make up an orange to human vision must be understood. Oranges have a few distinct features, the first being its bright orange hue, and the second being a rough texture on the skin. Fortunately, oranges are mostly spherical in shape, which proves to be easier when distinguishing between oranges and other objects on an orange tree.

The structure of this paper is as follows. Section II gives the detail explanation of algorithm to identify and count the orange fruit. After that Section III shows the results of developed algorithm with respect to each step. Then Section IV is about discussion, limitation of developed algorithm and finally in Section V, Conclusion is presented.

II. Material and Methods

The process of orange identification and counting is divided into a two-phase approach. In the first phase, the algorithm's objective is to identify and isolate the portions of the image that pertain to oranges. This phase typically involves segmentation and the creation of a mask or region of interest that encompasses the oranges in the image.

The second phase of the process is designed for the actual counting of oranges. This typically involves analyzing the mask or segmented regions generated in the first phase to determine individual orange objects or "blobs." Each blob represents a distinct orange in the image. The algorithm counts the number of these blobs to provide an estimate of the total number of oranges.

A. Orange Isolation

The first phase of the process, as illustrated in Fig. 1, draws inspiration from the work of A. Payne et al. [22] and involves the adaptation of an algorithm originally proposed by Payne for mango crop detection to be applied to oranges. This initial phase is comprised of five distinct steps, each aimed at the application of masks to the image for the purpose of identifying oranges.

In these steps, the image undergoes a comprehensive analysis within both the RGB and YCbCr color spaces, specifically seeking out regions that exhibit the characteristic coloration of oranges. These analyses result in the generation of color masks that highlight the areas matching the orange color profile in both color spaces. Subsequently, these individual color masks are combined to form a single comprehensive orange level mask, which serves as a vital component for the subsequent phase of the process.

This orange level mask is seamlessly transferred to the second phase, where it is employed to facilitate the counting of circular regions within the image. The counting operation essentially determines the total count of oranges present within the image. This two-phase approach offers a systematic and structured method for robust orange identification and counting within images.

The algorithm for orange identification consists of four distinct steps to create masks, which are then combined to form a final mask aimed at isolating orange components within the image.

Step 1: The initial step involves calculating a Normalized Difference Index (NDI) between the red and green color channels for each pixel. The NDI is computed as (g - r) / (g + r) [26]. Subsequently, a threshold is applied to identify regions with NDI values greater than 0. This step serves to accentuate areas with higher redness levels, while minimizing the influence of foliage.

Step 2: While Payne's algorithm originally employs a 3x3 variance filter for each color channel, the approach is adapted for orange identification. Instead, a 3x3 adaptive mean threshold filter is used. This new filter effectively removes excessive foliage from the image and eliminates irrelevant areas, such as the sky, which are unrelated to the oranges.

Step 3: The image is then transformed into the YCbCr color space, and the Cr and Cb channels are extracted for thresholding. The Cr channel measures the red difference within the image, and a threshold condition of pixel value >= 140 is applied, based on the observed average minimum Cr value across various regions in images containing oranges.

Step 4: The Cb channel is employed to evaluate the blue difference within the image. For oranges, a range of lower to medium values is acceptable, as they combine with the Cr channel to produce an orange color. Therefore, a threshold condition of 30 <= pixel value <= 120 is implemented.

After these four stages, the masks generated in each step are combined using Equation 1 to form a final mask. This final mask selectively highlights the orange components within the image:

$$pixel_{finale} = pixel_{NDI} \ \wedge \ pixel_{mean} \wedge pixel_{cr} \ \wedge \ pixel_{cb} \quad (1)$$

By applying these steps and the combination of masks, the algorithm effectively isolates the orange elements in the image, facilitating the subsequent phase of counting and identification.

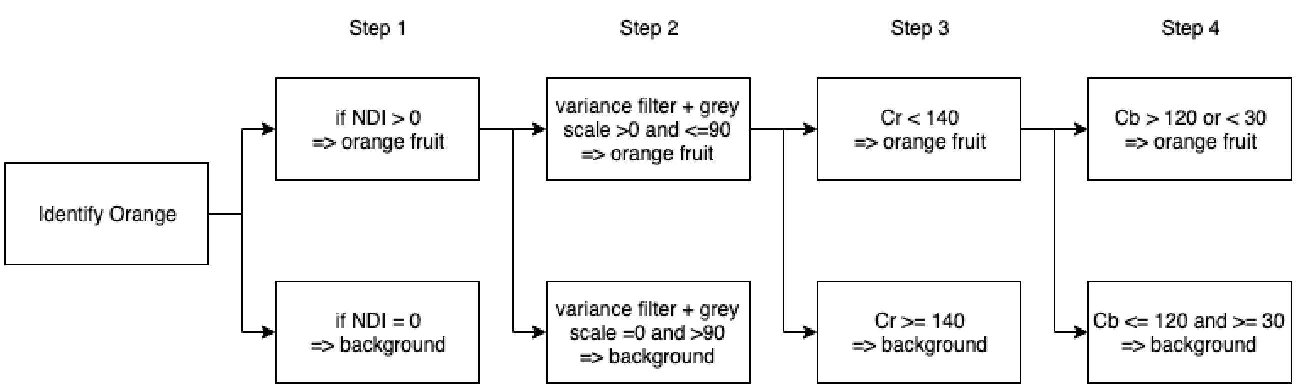


Fig. 1. Outline of the process of the stages of the algorithm.

B. Orange Counting

In the second phase, the mask that was generated previously is used to count the number of circles, each circle representing an orange fruit.

The process comprises multiple steps: First, a Gaussian blur is applied to the mask to soften its edges, preparing it for subsequent Canny edge detection. In the second step, a Hough transformation is executed to identify circular regions in the image, with carefully chosen parameter values such as dp, minDist, thresholds, minimum radius, and maximum radius, based on extensive experimentation [23]. In the third step, filtering mechanisms are implemented to refine the detected circles. The filtering involves a rolling average of circle radii, a check for the percentage of pixels within each circle matching the orange criteria established in phase 1, verification that the circle is within the image frame, and ensuring that the circle is not occluded by other detected oranges. These measures are taken to accurately identify oranges among the generated circles.

Algorithm 1: Orange Counting Algorithm

This algorithm is designed to count oranges within a binarized image. It proceeds as follows:

1. Start by applying a median blur to the image to reduce noise.
2. Employ a Hough circle detection technique to identify circular regions within the image. Fine-tune the parameters for best results.
3. Create an empty mask called "found" to keep track of detected circles and initialize the "averageR" variable.
4. For each detected circle:
   - Evaluate the ratio of orange pixels within the circle.
   - Check for overlap with previously detected circles in the "found" mask.
   - Perform certain checks based on circle size and orange pixel ratio to determine confidence in it being an orange.
   - If the circle is highly confident, mark it as a green circle.
   - If moderately confident, mark it as an orange, purple, or red circle based on its size.
   - If not filled enough but the size is close, mark it as maroon.
5. Update the "averageR" based on the newly confirmed oranges.
6. Continue this process for all detected circles.

The result is an image with marked circles in Fig. 2f , with green representing highly confident orange detections and other colors indicating varying levels of confidence. The algorithm filters out overlapping circles to ensure accurate orange counting.

Algorithm 2: Detect Overlapping Circles

This algorithm's purpose is to determine if a circle should be ignored due to excessive overlap with other circles. It operates as follows:

1. Receive input parameters: foundMask, a mask indicating taken (1) and untaken (0) regions, the circle under consideration, the binarized orange image img, and the percentage of orange pixels within the circle (orangePixels).
2. If the percentage of orange pixels within the circle is less than or equal to 50%, return True, indicating that the circle should be ignored.
3. Create a circleMask to represent the area covered by the current circle.
4. Iterate through the image:
   - Check if a pixel falls within the current circle and hasn't been taken (foundMask is False) and the circle mask (circleMask) is True.
   - Keep track of the areas that haven't been taken (areaNotTaken), the area newly covered by the current circle (areaNewOrangeInFrame), and the area that is both not taken and covered by the circle (areaFilledNotTaken).
5. Perform checks to determine if the orange in the circle is distinct and that the non-overlapping section is partly filled. If these conditions are not met, return False.
6. If all conditions are met, return true, indicating that the circle should not be ignored.

This algorithm evaluates circles for potential overlap, ensuring that they are distinct and meet specific criteria before considering them in the orange counting process

Algorithm 3: Count Orange Pixels within a Circle

This algorithm is designed to count the number of pixels within a circular region that have been marked as orange within a binarized image. The steps are as follows:

1. Receive the circle under consideration (`circle`) and the binarized orange image (`img`) as input, and aim to calculate the percentage of pixels that are marked as orange.
2. Create a `circleMask` to represent the area covered by the current circle.
3. Calculate the area of the circle using the formula `π * circle.radius^2`.
4. Initialize a counter, `numOutOfFrame`, to the area of the circle (i.e., `circleArea`).
5. Set the initial count of orange pixels, `numOrange`, to 0.
6. Iterate through the height and width of the image.
7. Decrease the value of `numOutOfFrame` by 1 for each iteration.
8. Check if the current pixel in the image is both marked as orange (`img[i, j]`) and falls within the circle

(`circleMask` is True).

9. If both conditions are met, increment the `numOrange` count by 1.
10. After processing all pixels, check that a sufficient portion of the circle is within the frame. If the percentage of the circle outside the frame exceeds 40%, increase `numOrange` by the number of pixels outside the frame (`numOutOfFrame`).
11. Calculate the percentage of the circle filled with orange pixels within the frame (`percentFilledInFrame`) as a ratio of `numOrange` to the total number of pixels in the image.
12. Calculate the overall percentage of the circle filled with orange pixels, considering both inside and outside the frame (`percentFilled`) as a ratio of `numOrange` to the total number of pixels minus those outside the frame (`numTotal - numOutOfFrame`).
13. Return the minimum of `percentFilledInFrame` and `percentFilled`.

This algorithm assesses the number of orange pixels within a circular region and accounts for pixels both within and outside the frame. The result represents the percentage of orange-filled area within the circle, ensuring a comprehensive measure of orange presence.

III. RESULTS

The algorithm was subjected to a rigorous testing process using a diverse set of online free available orange images that varied in brightness and foliage levels. Algorithm is experimented on set of 10 orange images from [25] which are freely available. Across the entire 10 images, the algorithm exhibited a noteworthy level of success, achieving an approximate 72.4% accuracy in confidently detecting the presence of oranges with standard deviation of +/- 12.20. However, it's important to note that the results were not solely binary, as the algorithm also provided insights into its level of confidence for each image. Fig. 2 shows the correlation of decrease in accuracy of algorithm against actual as fruit count increase. Both trend lines have strong R2 and steady different curves that diverge as fruit count increases.

For images where the algorithm expressed only moderate confidence in its results, typically numbering around one to two images, a specific visual representation was employed to convey this information. In these cases, oranges that the algorithm was highly confident about were distinctly highlighted with green circles. Oranges for which the algorithm expressed moderate confidence were encircled in orange, while those instances where the algorithm had low confidence were marked with either purple or red circles.

Fig. 3 shows step by step results of orange ident and counting also. Fig. 3(a) is original image taken from [24]. Fig. 3(b) is result of the first step in which algorithm is to produce a normalized index between the red and green channels. Fig. 3(c) is step 2 in which an adaptive mean threshold filter replaces the variance filter to improve foliage and sky removal. Fig. 3(d) and Fig. 3(e) shows step 3 and 4 inch which the Cr channel uses a threshold of pixel value >= 140 for medium to high red levels, and the Cb channel employs a threshold condition of 30 <= pixel value <= 120 to detect oranges based on blue differences respectively. In Fig. 3(f) show only orange components of image by applying a mask. Fig. 3(g) and Fig. 3(h) shows the result of orange counting section in which identify the circular regions and filter the circles.

This approach not only provides a quantitative measure of algorithm performance but also offers visual cues to users, making it easier to interpret and potentially refine the results, which can be particularly valuable in scenarios where the algorithm's output may be used for decision-making or further analysis. Fig. 4 shows count of circle for color green, orange, and red, purple in Fig. 3(f). It shows high confidence by detecting green circles as per Algorithm 1.

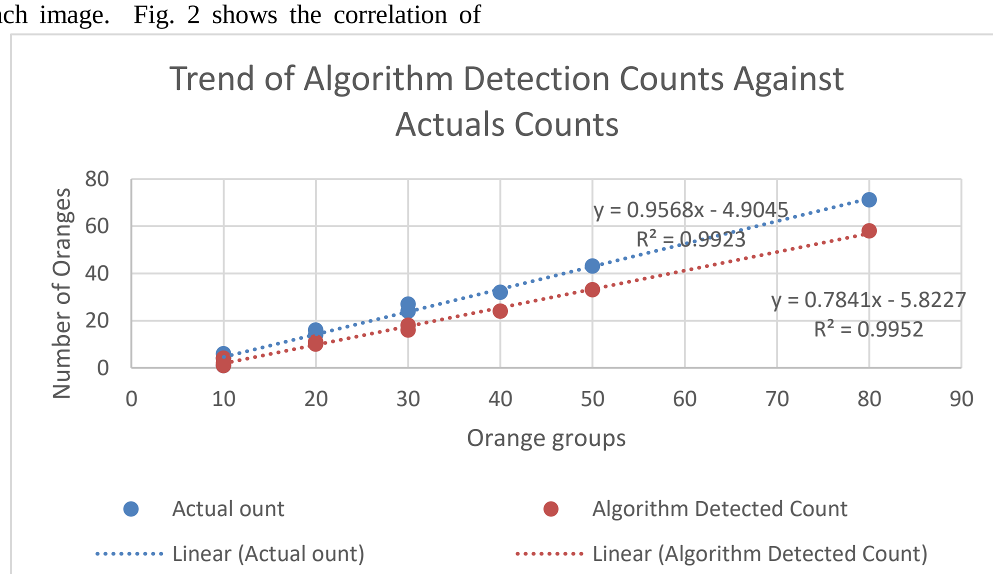
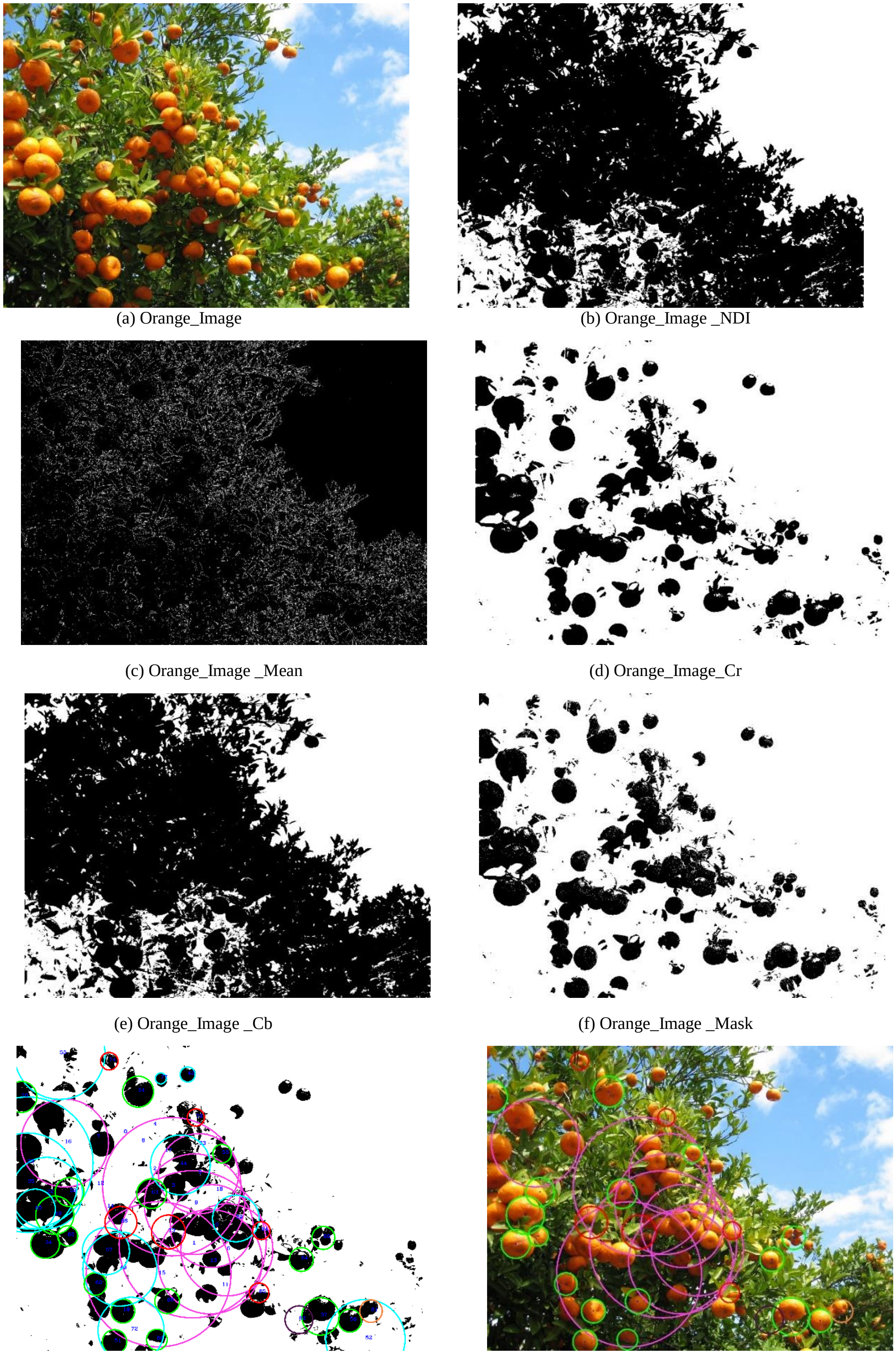


Fig. 2. Trend of algorithm detection counts against actuals counts.



(a) Orange_Image (b) Orange_Image _NDI

(c) Orange_Image _Mean (d) Orange_Image_Cr

(e) Orange_Image _Cb (f) Orange_Image _Mask

(g) Orange_Image _Circle (h) Finale_Result_Image

Fig. 3. An example image going through each step to detect the oranges within the image.

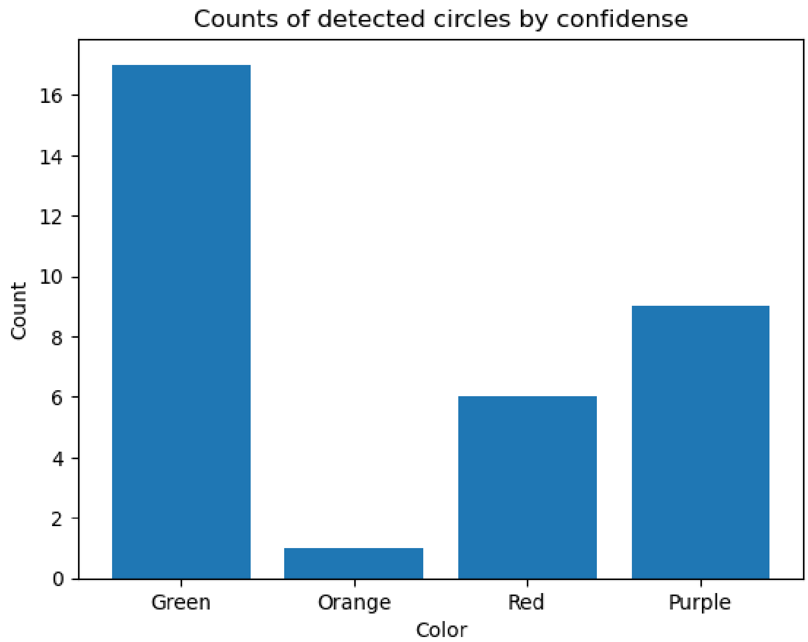


Fig. 4. Count of detected circles by confidence.

IV. Discussion

During the testing of the proposed algorithm, it became evident that it performs significantly better on images with a resolution larger than 1300x1300 pixels. However, it has limitations—it fails to work on images where the diameter of the oranges exceeds half of the screen size, or when the oranges are very small (less than 150 pixels). This limitation is attributed to the choice of the minimum separation value between circles in.

It was observed that these limitations can be effectively addressed by using higher image resolutions. In practical applications where image quality and consistent zoom levels are known and controlled, these shortcomings can be overcome. Additionally, analyzing luminance levels could be explored as a means to normalize images, particularly in cases where highly illuminated images pose challenges for orange detection.

The algorithm exhibits varying levels of performance due to inefficiencies within the circle detection process, making it computationally expensive and particularly evident in handling large and cluttered images. Further optimization is essential to enhance its efficiency and adaptability across different image scenarios.

V. Conclusion

The results derived from this algorithm demonstrate achieving an accuracy rate of approximately 72.4%. Result summarizes that purpose algorithm works well on lower fruit count. However, there remains a clear imperative for further enhancements. The algorithm's current limitations are evident in its inability to account for varying levels of ripeness in oranges. Moreover, the section responsible for orange detection lacks computational efficiency, thereby affecting program runtime.

Future research endeavors should prioritize the development of a more efficient algorithm for shape detection within images. Additionally, it is essential to optimize the color segmentation component to handle a broader range of brightness levels and mitigate issues related to clutter. These improvements will contribute to the advancement of automated fruit counting processes in agriculture and other related domains.

References

[1] Shankarpure, M. R.,& Patil, D. D. A Comprehensive Survey on Methods and Techniques for Automated Fruit Plucking. International Journal of Intelligent Systems and Applications in Engineering, 11(1), 156-168. 2023.

[2] Anuja Bhargava; Atul Bansal(2021). Fruits and vegetables quality evaluation using computer vision: A review, Journal of King Saud University - Computer and Information Sciences, Volume 33, Issue 3, Pages 243-257.

[3] Erdenee, B.; Ryutaro, T; Tana, G.(2010), Particular Agricultural Land Cover Classification Case Study Of Tsagaannuur, Mongolia. In: IEEE International Geoscience & Remote Sensing Symposium, 3194-3197.

[4] V.K. Tewari; A.K. Arudra; S.P. Kumar; V. Pandey; N.S. Chande (2013),Estimation of plant nitrogen content using digital image processing Int. Commission Agricu. Biosyst. Eng., pp. 78-86.

[5] M. Krishna; G. Jabert (2013) Pest control in agriculture plantation using image processing, IOSR J. Electron. Commun. Eng. (IOSR-JECE), pp. 68-74.

[6] J.K. Patil; R. Kumar (2011)Advances in image processing for detection of plant diseases, J. Adv. Bioinf. Appl. Res. ISSN, pp. 135-141.

[7] Hannan M.W.; Burks T.F.; Bulanon D.M.(2009), A Machine Vision Algorithm for Orange Fruit Detection, Agricultural Engineering International: the CIGREjournal, Vol-XI.

[8] Bulanon D.M.; Burks T.F.;Alchanatis V.(2009), Image Fusion of visible and thermal images for fruit detection, Biosystems Engineering, Vol-103, Issue-1, pages:12-22.

[9] Hannan M.W.; Burks T.F.; Bulanon D.M.(2009), A MachineVision Algorithm for Orange Fruit Detection, Agricultural Engineering International: the CIGRE journal, vol-XI,Pages:1-7.

[10] Leemans, V. and Destain, M.-F(2004), A real-time grading method of apple based on features extracted from defects, Journal of Food Engineering, vol.61, pp.83-89.

[11] Hayashi Shigehiko; Ota Tomohiko; Kubota Kotaro; Ganno Katsunobu and Kondo Naoshi (2005), Robotic Harvesting Technology for Fruit Vegetables in Protected Horticultural Production, Information and Technology for Sustainable Fruit and Vegetable Production FRUTIC , France.

[12] Bulanon D.M., Burks T.F. and Alchanatis V.(2008), Study of temporal variation in citrus canopy using thermal imaging for citrus fruit detection, Biosystems Engineering, Vol-101, Issue 2, Pages 161-171.

[13] Shasi Buluswar (2002), Models for Outdoor Color Vision, Doctoral dissertation, University of Massachusetts, Amherst.

[14] Bulanon D.M. ; Burks T.F.; Alcahnatis V.(2009) , Improving Fruit detection for robotic fruit harvesting", ISHS Acta Horticulturae 824: Internation Symosium on Application of Precision Agriculture for Fruits and Vegetables.

[15] Woo Chaw Seng and Seyed Hadi Mirisaee(2009), A New Method for Fruits Recognition System, MNCC Transactions on ICT, Vol. 1, No. 1.

[16] Blasco J.;Aleixos N.; Molto E.(2003), Machine Vision System for Automatic Quality Grading of Fruit, Biosystems Engineering , Vol-85, Issue 4, Pages-415-423.

[17] Fernández, L., Castillero, C. and Aguilera, J. M.(2005), An application of image analysis to dehydration of apple discs Journal of Food Engineering, vol.67, pp.185-193.

[18] Jimenez A.R., Ceres R., Pons J.L.(2000),A Survey of Computer Vision Methods for Locating Fruit on Trees, Transaction of the ASAE, Vol. 43(6), pages: 1911-1920.

[19] Borse, J.; Patil, D (2021). Tracking Keypoints from Consecutive Video Frames Using CNN Features for Space Applications. Tehnički glasnik, 15 (1), 11-17.

[20] Lokhande, Meghana P. and Dipti Durgesh Patil(2021),Trust Computation Model for IoT Devices Using Machine Learning Techniques, Proceeding of First Doctoral Symposium on Natural Computing Research. Lecture Notes in Networks and Systems, vol 169. Springer.

[21] Lokhande, Meghana P. and Dipti Durgesh Patil(2022),Object Identification in Remotely-Assisted Robotic Surgery Using Fuzzy Inference System, Demystifying Federated Learning for Blockchain and Industrial Internet.of Things, , pp. 58-73.

[22] A. Payne; K. Walsh; P. Subedi; and D. Jarvis(2013), Estimation of mango crop yield using image analysis – Segmentation method, Computers and Electronics in Agriculture, vol. 91, pp. 57–64.

[23] V. K. Yadav;S. Batham, A. K. Acharya, and R. Paul(2014), Approach to accurate circle detection: Circular Hough Transform and Local Maxima concept, in 2014 International Conference on Electronics and Communication Systems (ICECS), (Coimbatore), pp. 1–5, IEEE.

[24] Citrus Farming, https://www.usesfordiatomaceousearth.com/citrus-farming/ , 14 /10/2023.

[25] https://www.pexels.com/photo/orange-fruit-on-tree-3804878/, 23/10/2023.

[26] Stajnko, D., Lakota, M., Hocevar, M., 2004. Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. Computers and Electronics in Agriculture 42, 31–34.