International Journal of Scientific Research and Management (IJSRM)
||Volume||5||Issue||07||Pages||6524-6530||2017||
Website: www.ijsrm.in ISSN (e): 2321-3418
Index Copernicus value (2015): 57.47 DOI: 10.18535/ijsrm/v5i7.92

An Analysis of Fruit Detection and Yield Estimation Using on Tree Image

A. Saranya[1], Dr.N. Sujatha[2]

[1]PG & Research Department of Computer Science, Raja Doraisingham Govt. Arts College, Sivagangai.
Email: saradolly17@gmail.com
[2]Assistant Professor, PG & Research Department of Computer Science, Raja Doraisingham Govt. Arts College, Sivagangai.

Abstract:
In this paper, we present a technique to evaluate citrus fruit yield from the tree images. Manually counting the fruit for yield estimation for marketing and other managerial tasks is time consuming and requires more human resources, which do not always come cheap. Different methods have been used for estimation, yet separation of fruit from its background poses complicated and challenges, and render the exercise inaccurate. In this paper, we use Bicubic Interpolation technique to detect and count the number of fruits from its background, which segments the image accurately thus enabling more accurate yield estimation

Keywords—Precision agriculture, yield estimation, bicubic Interpolation, leaf occlusion, illumination, morphology

I Introduction

In citrus field, crop estimation is classically takes a few weeks formerly to fruitage to estimate the resource requirement. The manual process of crop valuation is done by hand pawns. First, yield would be estimated at numerous periods during crop development but it requires enormous labor cost and time. Particularly, low-cost yield estimation is important for growers, specifically if it can be done timely in the growing season. As orange juice to be processed within 48 hours of harvesting, orange juice producers need suppliers to provide accurate yield estimates to guarantee that their juice plants can run at full capacity given the time constraints. Also, more accurate yield estimates will allow farmers to plant more precisely for their harvesting labor and other logistical needs. Image processing can help in improving the decision making process for irrigation, fruit organizing and yield estimation. Detection of fruit is important as the following fruit counting depends on accurate detection. Citrus fruit have different properties that can be used for detection purposes. The most natural property to be used for such purpose is the color. In the past, color has been extensively used. Color on its own may not provide enough information as it may change depending upon how ripe the fruit is. For instance, unripe oranges may be greenish, while over-ripe oranges maybe brownish. That poses a challenge when detection is based on filtering of orange color only. Lighting is an additional challenge as the oranges may look differently, under varying lighting conditions, such as bright sunlight, cloudy, and evening. Under cloudy situations, the images are more consistent in terms of brightness and intensity changes. While in the latter case, things may look different if they are exposed to direct sunlight as opposed to when they are under a shadow. Another problem is that of occlusion. Occlusion may be affected either by leaves or by the neighboring oranges. In the first case, a single orange maybe counted as more than one, as leaves may affect the shape of the orange. In the later case, fruits may appear as one larger fruit with irregular shape. In the first case, a need rises for a way to count two or more smaller citrus blobs as one, while in the later case we need to break down

a larger blob into smaller units, where each unit is counted as a separate fruit.

For fruit detection and mass estimation using a closed imaging system, several machine vision applications have been studied. Developed a machine vision system is used to identify citrus fruit and its size on a canopy shake and catch harvester during harvesting. The citrus detection and size identification algorithm included a Bayesian classifier using color information and watershed segmentation. The total area of citrus fruit was measured, yielding between citrus fruit count and actual mass.

A highly saturated area recovering algorithm (HSAR) and watershed algorithm using *h*-minima transform were used to identify the number of fruit and fruit sizes. Fruit recognition using outdoor images in open areas, however, is challenging because object colors in images vary greatly under different illumination conditions. Many studies of fruit detection in outdoor images have shown reduced performance developed a citrus yield mapping system using machine vision with outdoor imaging. The acquired images were converted to the Hue, Saturation, and Value (HSV) color space, and then fruit objects were segmented using erosion and dilation. A connected component was considered a fruit. However, the segmentation using erosion, and dilation caused underestimation because some groups of fruit were not segmented properly and were considered as one fruit.

The overall goals of this study were to develop a machine vision system using outdoor RGB images to correctly count citrus fruit and estimate their mass, which can be used in creating a geo-referenced fruit drop map. Specific objectives of the research were:

1. To build a machine vision system suitable for field conditions in citrus groves.
2. To develop an image enhancement algorithm to address problems from variable illumination.
3. To develop an image processing algorithm that can estimate fruit count and mass.

II Related Works And Existing System

Manually counting the fruit for yield estimation for marketing and other executive tasks is time consuming and requires human resources, which do not always come cheap. Various approaches have been used for the said purpose, nevertheless separation of fruit from its background is complicated, and renders the exercise inaccurate. Here, they use k-means segmentation for recognition of fruit, which segments the image accurately thus enabling more accurate yield estimation.

The k-means segmentation algorithm is used on orange tree images. First, some preprocessing steps are performed including noise reduction and image enhancement.

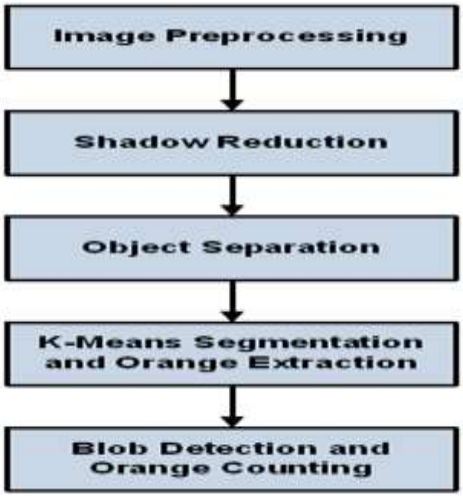


Fig: Overview of the technique

Then, extract oranges using blob detection and size calculation which is then followed by the final yield estimation. The details of these steps are given in the following sub-sections.

Image Preprocessing: In general, images contain a lot of redundant information not needed for the application in hand. The image may contain noise which makes the edge detection and the segmentation tasks prone to errors. Hence, it isoften necessary to carry out certain type of noise reduction and image enhancement before processing of the images. It smooths the image without effecting significant features of the image

such as lines, edges or other important details that are necessary for analysis and interpretation of the images.

Shadow Reduction

Lighting conditions play a vital role in the performance of many computer vision algorithms. Shadows in an image can make problems in recognition, segmentation and tracking algorithms to produce desired outcomes. Different objects can be combined through shadows and they can obscure object recognition systems. When dealing with outdoor images, shadow minimization is an important task. So, converts the RGB image to L*a*b image, where L is the luminosity layer, while a and b represents color-opponent dimensions. Next, increase the luminosity of the image which results in reduced shadow effect in the image. Finally, the image is converted to RGB color space after replacing the luminosity layer with the processed data.

Object Separation

One of the major challenges in orange counting is that of overlapped oranges. Due to overlapping, multiple fruits maybe counted as a single fruit which negatively impacts the fruit counts and the yield estimates. To overcome this challenge and to separate the overlapping fruit, convolve the image with a variance mask. After the convolution, each pixel in the output RGB image contains the neighbor variance of the R, G and B channels, respectively. The image is then transformed to gray scale by taking the mean of the three color channels. Finally, a threshold is applied on the image. These steps not only separate overlapping fruit, they also help in reducing undesired edges such as those within the foliage or the grass.

K-Means Segmentation and Orange Extraction

Image Segmentation is the very huge part of the whole process of yield estimation. Here, k-means clustering algorithm is used for orange segmentation. K-means clustering is an unsupervised classification technique which finds a pattern in a collection of unlabeled data. The k-means algorithm aims at reducing a squared error function by iteratively reorganizing the clusters. The iterations continue until the cluster means do not move further than a certain cut-off value. K-means algorithm is popular due to its simplicity and relatively low computational complexity. It is suitable for the number of clusters (K) is easy to choose. An image of an orange tree generally consists of regions representing the oranges, leaves, branches and sky.

III PROPOSED SYSTEM

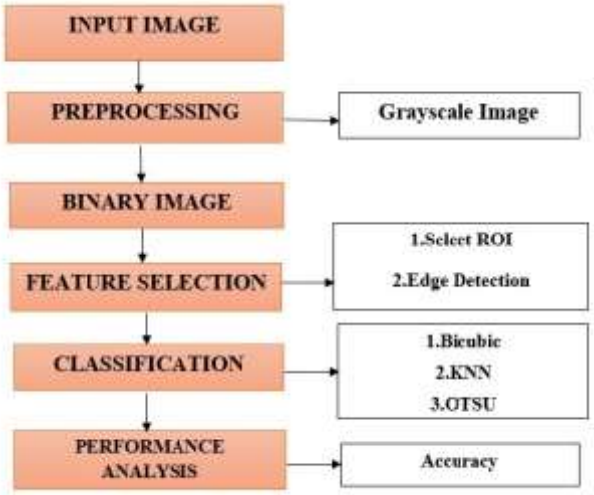


FIG. PROPOSED BLOCK

Bicubic Interpolation

In Bicubic interpolation, it uses the weighted average of four translated pixel values for each output pixel value.

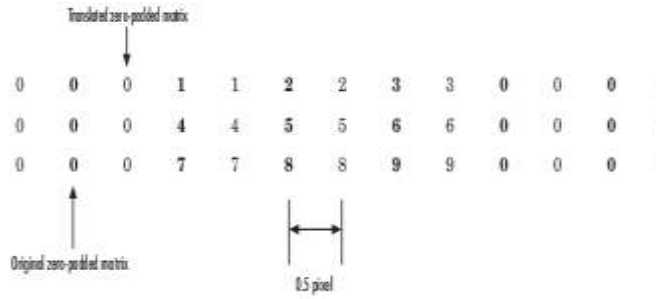
For example, suppose this matrix,

```
1 2 3
4 5 6
7 8 9
```

represents your input image. To translate this image 0.5 pixels in the positive horizontal direction using bicubic interpolation. The Translate block's bicubic interpolation algorithm is illustrated by the following steps:

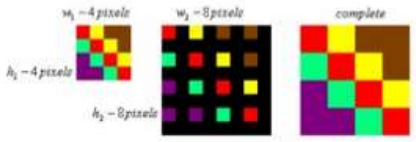
1. Zero pad the input matrix and convert it by 0.5 pixel to the right.

DOI: 10.18535/ijsrm/v5i7.92



2. Create the output matrix by replacing each input pixel value with the weighted average of the two translated values on either side. The outcome is the following matrix where the output matrix has one more column than the input matrix:

0.375	1.5	3	1.625
1.875	4.875	6.375	3.125
3.375	8.25	9.75	4.625



The black pixels represent empty spaces where interpolation is needed, an complete picture is the result of nearest neighbor interpolation.

Scaling algorithm is to find appropriate spot to put the empty spaces inside original image, and to fill all those spaces with livelier colors.

Fig. example for interpolation

HSV Color Space

HSV stands for Hue, Saturation, and Value. The value represents intensity of a color, which is decoupled from the color information in the represented image. The hue and saturation components are closely related to the way human eye perceives color resulting in image processing algorithms with physiological basis. As hue varies from 0 to 1.0, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red, so that there are actually red values both at 0 and 1.0. As saturation varies from 0 to 1.0, the related colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness, varies from 0 to 1.0, the corresponding colors become increasingly brighter.

A. Hue Calculation

The hue value can be calculated using the following formula:

$$H = \begin{cases} 60° \times \left(\frac{G' - B'}{\Delta} \bmod 6\right), C_{max} = R' \\ 60° \times \left(\frac{B' - R'}{\Delta} + 2\right), C_{max} = G' \\ 60° \times \left(\frac{R' - G'}{\Delta} + 4\right), C_{max} = B' \end{cases}$$

B. Saturation Calculation

The saturation value can be calculated using the following formula:

$$S = \begin{cases} 0, & \Delta = 0 \\ \frac{\Delta}{C_{max}}, & \Delta <> 0 \end{cases}$$

C. Value calculation

The value is calculated using the following formula.

$$V = C_{max}$$

Region of Interest

Generally, the image is divided into regions which represent the relevant objects in best method in the scene. Many region properties like area, shape, statistical parameters and texture can be extracted and used for additional analysis of the data. [11]

KNN for Classification

Let's see how to use KNN for classification. In this case, we are given some data points for training and also a new unlabeled data for testing. Our goal is to find the class label for the new point. The algorithm has different behavior based on k.

Case 1: k = 1 or Nearest Neighbor Rule

This is the simplest scenario. Let x be the point to be labeled. Find the point closest to x. Let it be y. Now nearest neighbor rule asks to assign the label of y to x. This seems too simplistic and sometimes even counter intuitive. If you feel that this procedure will result a huge error, you are right – but there is a catch. This reasoning holds only when the number of data points is not extended.

Assume all points are in a D dimensional plane. The number of points is reasonably large. This means that the density of the plane at any point is fairly high. In other words, within any subspace there is adequate number of points. Consider a point x in the subspace which also has a lot of neighbors. Now let y be the nearest neighbor. If x and y are sufficiently close, then we can assume that probability that x and y belong to same class is fairly same – Then by decision theory, x and y have the same class.

The striking result of Nearest Neighbor rule is to obtain a fairly tight error bound. The bound is

$$P^* \leq P \leq P^*\left(2 - \frac{c}{c-1}P^*\right)$$

Where P^* is the Bayes error rate, c is the number of classes and P is the error rate of Nearest Neighbor. The result is certainly very striking because it says that if the number of points is fairly large then the error rate of Nearest Neighbor is less than twice the Bayes error rate.

Case 2: k = K or k-Nearest Neighbor Rule

This is a straightforward extension of KNN. Basically what we do is that we try to find the k nearest neighbor and do a majority voting. In general, k is odd when the number of classes is 2. Let's say k = 5 and there are 3 instances of C1 and 2 instances of C2. In this case, KNN says that new point has to label as C1 as it forms the majority. We follow a similar argument when there are multiple classes.

One of the direct extensions is not to give 1 vote to all the neighbors. A very common thing to do is *weighted KNN* where each point has a weight which is typically calculated using its distance.

It is rather obvious that the accuracy might rise when you increase k but the computation cost also increases.

Some Basic Observations

1. If we assume that the points are d-dimensional, then the straight forward implementation of finding k Nearest Neighbor takes O(dn) time.
2. KNN in two ways – One way is that KNN tries to estimate the following probability of the point to be labeled (and apply Bayesian decision theory based on the posterior probability). An alternate way is that KNN calculates the decision surface (either implicitly or explicitly) and then uses it to decide on the class of the new points.
3. Various possible ways to apply weights for KNN – One popular example is the Shephard's method.
4. Even though the naive method takes O(dn) time, it is very hard to do better unless we make some other assumptions. There are some well-organized data structures like **KD-Tree** which can reduce the time complexity but they do it at the cost of increased training time and complexity.
5. In KNN, k is usually an odd number if the number of classes is 2.
6. Choice of k is very critical – A small value of k means that noise will have a higher influence on the result. A huge value makes it computationally expensive and kind of defeat the basic philosophy behind KNN (that points that are near might have similar densities or classes).A simple approach to select k is set $k = \sqrt{n}$
7. The interesting facts about data structures and algorithms are when you apply KNN on graphs – **Euclidean minimum spanning tree** and **Nearest neighbor graph** .

8. There are also some nice techniques like condensing, search tree and partial distance that try to reduce the time taken to find the k nearest neighbor.

DOI: 10.18535/ijsrm/v5i7.92

Otsu's Method

Among all the segmentation methods, Otsu method is one of the most successful methods for image thresholding because of its simple calculation. Otsu's method is an automatic threshold selection based regional segmentation method.

In this method [12], it states that the objective function of Otsu method is equivalent to that of K means method in multilevel thresholding. It based on the same criteria that minimize the class variance. Also it works on global thresholding and it requires computing a gray level histogram before running. This method produces good segmentation result. At the same it takes comparatively more time and increases the complexity of the algorithm.

IV Results and discussion

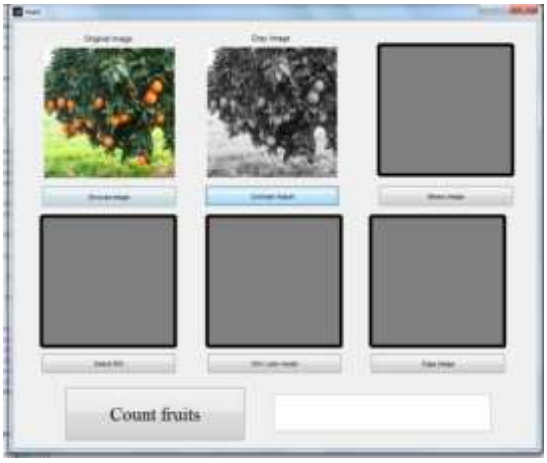


Fig. Input images

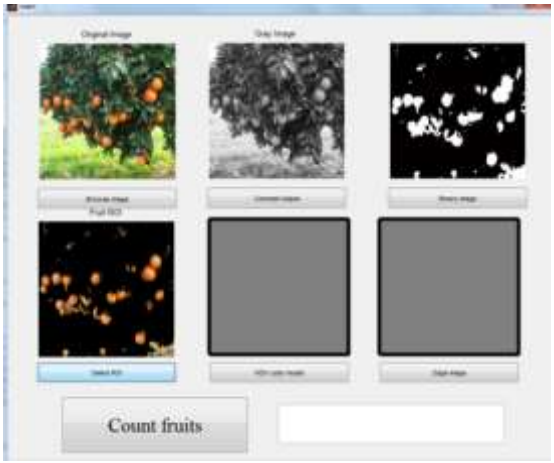


Fig. Binary and ROI using interpolation results

Shape values:

Area - 5

Centroid - (1.800000, 308.600000)

Perimeter - 5.414214e+00

Roundness - 2.143429e+00

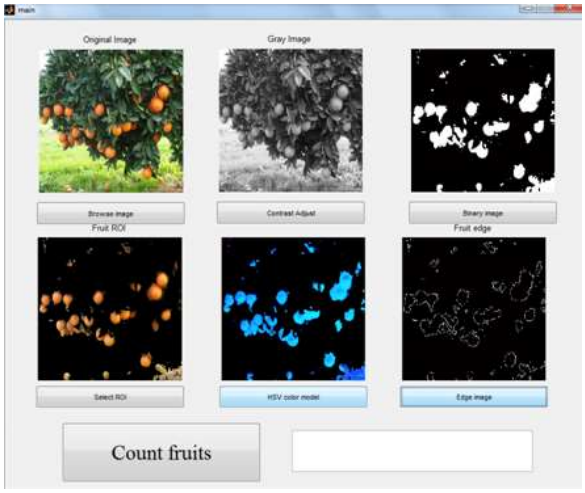


Fig. Results of HSV and edge using classifiers

Mean red - 22.023365

Mean green - (14.480576)

Mean Blue - 7.563736

Mean Hue - 0.012170

Mean Saturation - (0.089391)

Mean Value - 22.023365

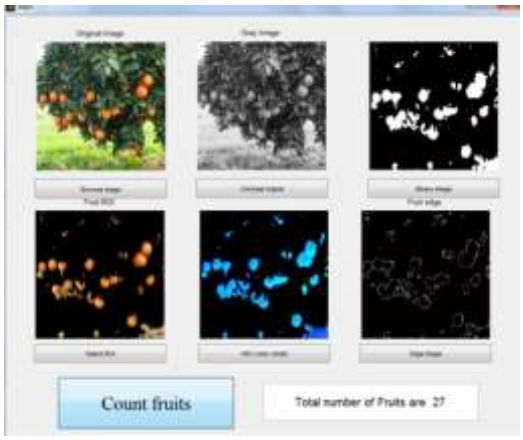


Fig. fruit counts
Total number of Fruits: 27

V Performance Analysis

The performance results for the proposed schemes is shown below

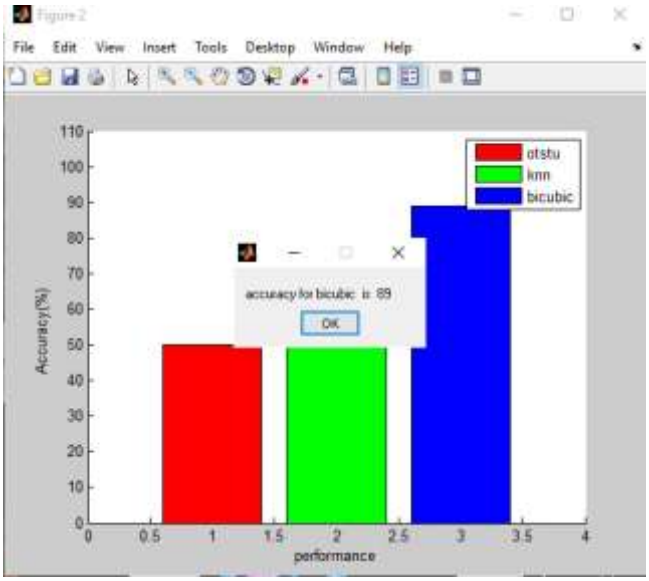


Fig. performance analysis

VI Conclusion

We have presented a technique for the segmentation, detection and yield measurement of citrus fruit. The proposed approach gives very good results under varying lighting conditions, occlusion of leaves and overlapping of fruits on the images taken from varying distances from the orange trees. Our experiments on three different datasets showed an accuracy of 95%.

References

1. N.-S. Chang & K. - S. Fu (1980), "Query by Pictorial", IEEE Transactions on Software Engineering, Pp. 519–524. [2] W. Niblack (1993), "
2. The QBIC Project: Querying Images by Content using Color, Texture and Shape", Proceedings of SPIE, Vol. 1908, Pp. 173–187.
3. A. Pentland (1994), "Photobook: Content-based Manipulation of Image Databases", Proceedings of SPIE Storage and Retrieval for Image and Video Databases II, Pp. 34–47.
4. Markus Stricker & Markus Orengo (1995), "Similarity of Color Images", Proceedings / Conference Na Proceedings of SPIE Storage and Retrieval for Image and Video Databases, Pp. 381–392.
5. R. Harrell, D. Slaughter, and P. Adsit, "A fruit-tracking system for robotic harvesting," Machine Vision and Applications, vol. 2, no. 2,pp. 69–80, 1989.
6. R. Harrell, P. Adsit, T. Pool, R. Hoffman et al., "The florida robotic grove-lab." Transactions of the ASAE, vol. 33, no. 2, pp. 391–399, 1990.
7. A. Jimenez, R. Ceres, J. Pons et al., "A survey of computer vision methods for locating fruit on trees," Transactions of the ASAE-American Society of Agricultural Engineers, vol. 43, no. 6, pp. 1911–1920, 2000.
8. F. Pla, F. Juste, and F. Ferri, "Feature extraction of spherical objects in image analysis: an application to robotic citrus harvesting," Computers and Electronics in Agriculture, vol. 8, no. 1, pp. 57–72, 1993.
9. E. Tutle, "Image controlled robotics in agricultural environments," 1984.
10. D. Stajnko, M. Lakota, and M. Hoˇcevar, "Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging," Computers and Electronics in Agriculture, vol. 42,no. 1, pp. 31–42, 2004.
11. A. Saranya and Dr. N. Sujatha," Performance analysis of Various Segmentation Techniques", in IJIR-Vol-3, Issue-2.2017.
12. L. Dongju and Y. Jian, "Otsu method and k-means," in Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on, vol. 1, 2009, pp. 344–349.